DEPARTAMENTO DE ENGENHARIAS E DE CIÊNCIAS COMPUTAÇÃO

MESTRADO EM ENGENHARIA INFORMÁTICA E DE TELECOMUNICAÇÕES

UNIVERSIDADE AUTÓNOMA DE LISBOA

"LUÍS DE CAMÕES"

PREVENÇÃO DE LESÕES EM JOGADORES DE FUTEBOL USANDO MACHINE LEARNING

Dissertação para a obtenção do grau de Mestre em Engenharia Informática e Telecomunicações

Autor: David Alexandre Cruz Monteiro

Orientador/a: Professor Doutor Laercio Cruvinel Júnior

Número do/a candidato/a: 30003043

Julho de 2023

Lisboa

Dedicatória

Este trabalho é dedicado especialmente à minha família, pois sem a força deles eu não estaria a escrever esta dedicatória e a alcançar esta conquista. Dedico particularmente aos meus pais, por tudo aquilo que fizeram por mim e me apoiarem tanto no meu percurso do ensino primário até ao secundário, mas especialmente no ensino superior por ter sido uma das etapas mais difíceis da minha vida. Agradecer imenso à minha avó Idalina que sem o suporte dela também não seria possível cumprir esta missão. Às minhas tias Carla e Marísia também queria agradecer por todo o carinho e apoio prestado, assim como ao meu irmão Igor e primos/as. Mas de frisar o apoio da minha namorada que mesmo quando estive quase a desistir puxou-me para cima e foi insistente para eu estar a terminar este trabalho, sem ela também não seria possível a conclusão deste trabalho. A todos os citados este trabalho é para vocês muito obrigado de coração!

Agradecimentos

Primeiramente agradecer à instituição a que faço parte que é a UAL (Universidade Autónoma de Lisboa) ao Departamento de Engenharias e Ciências da Computação, por esta grande jornada que foi o mestrado.

Gostaria de bastante agradecer ao Professor Laercio Cruvinel Júnior, orientador da minha dissertação por toda a ajuda dada durante este período, pelos ensinamentos durante toda a minha caminhada no Ensino Superior e nesta dissertação. Obviamente que também gostaria de agradecer aos restantes professores que fizeram parte desta minha caminhada não só do mestrado como nos meus 5 anos de Ensino Superior, Professor Mário Marques da Silva, Professor Gonçalo Valadão e Professora Adriana Fernandes, pois sem a vossa ajuda e transmissão de conhecimentos eu não estaria a terminar esta dissertação, muito obrigado por tudo!

Epígrafe

“If A is success in life, then A = X + Y + Z.” Work is X; Y is play; and Z is keeping your mouth shut.” – Albert Einstein

Resumo

O objetivo geral desta dissertação é analisar de que maneira os clubes podem prever lesões musculares nos seus atletas, mediante o estudo das métricas que os *staff*s técnicos dos clubes utilizam ou poderiam utilizar, de maneira a construir-se um modelo preditivo de ML (Machine Learning), para então prevenir-se este tipo de lesões. Para alcançar este objetivo, temos primeiramente de perceber como os clubes de futebol atualmente utilizam a tecnologia para analisarem os seus jogadores.

Para podermos conseguir prever estas possíveis lesões, também precisámos de saber porque os futebolistas se lesionam tanto atualmente.

Antes da construção do modelo, foi necessário primeiramente pesquisar um conjunto de dados que iriam conter as métricas idênticas às que os *staff*s técnicos dos clubes de futebol avaliavam para prevenir lesões nos seus jogadores de futebol. Como não foi possível encontrar dados reais e completos dos clubes e jogadores de futebol, acabou-se por encontrar um conjunto de dados fictícios com métricas idênticas às que pretendíamos. Embora este conjunto de dados não possuísse todas as métricas necessárias para a construção de um modelo idêntico ao mundo real, foi utilizado por possuir métricas de dados físicos, apesar de não possuir métricas de dados biológicos.

Para a construção do modelo de ML foi utilizado o serviço da Google, o Google Colab para melhor entendimento dos dados, já que este trabalho se foca bastante na área de mineração de dados, análise de dados e ciência de dados. Durante a construção do modelo, é requisito identificar o tipo de algoritmo a implementar. Para este tipo de problema, os algoritmos de classificação podem satisfazer os objetivos do trabalho. Após a implementação do modelo de ML, a identificação do melhor algoritmo para resolver este problema permite analisar os dados minerados para assim responder ao objetivo geral do trabalho.

Palavras-chave: Prevenção de Lesões; Machine Learning; Jogadores de Futebol; Modelo Preditivo.

Abstract

The main objective of this dissertation is to analyze how clubs can prevent their athletes from contracting muscle injuries, through metrics that the technical *staff* of the club's use, to build a predictive model of ML (Machine Learning), in order to prevent this type of injury. But before that, we would have to understand how football clubs currently use technology to analyze their players, that is, what is behind everything.

To be able to predict these possible injuries, we also needed to know why footballers get injured so much these days.

Before building the model, it was first necessary to research a *dataset* that would contain metrics identical to those that the technical *staff* of football clubs evaluated to prevent injuries in their football players. As it was not possible to find real and complete data on football clubs and players, we ended up finding a fictitious dataset with metrics identical to the ones we wanted. Although this dataset did not have all the necessary metrics to build a model identical to the real world, we ended up using this dataset because it has physical data metrics despite not having biological data metrics.

For the construction of the ML model, the Google service, Google Colab, was used for a better understanding of the data, since this work focuses a lot on the area of data mining, data analysis and data science. During the construction of the model, the type of algorithm to implement should be identified. For this type of problems, the classification algorithms would satisfy the objectives of the work. After implementing the ML model, it is necessary to identify which would be the best algorithm to solve this problem and analyze the mined data in order to respond to the main objective of the work.

Keywords: Injury Prevention; Machine Learning; Football Players; Predictive Model.

Índice

Lista de Quadros/Gráficos

Lista de Fotografias/Ilustrações

Lista de Siglas e Acrónimos

ACWR	Acute Chronic Workload Ratio
AU	Arbitrary Units
CIO	Chief Information Officer
COVID-19	Corona virus disease 2019
CRISP-DM	Cross Industry Standard Process for Data Mining
EWMA	Exponentially Weighted Moving Average
FIFA	Fédération Internationale de Football Association
GPS	Global Positioning System
IoT	Internet of Things
K-NN	K-Nearest Neighbors
ML	Machine Learning
NFC	Near Field Communication
RA	Rolling Averag
RPE	Rating of Perceived Exertion
sRPE	sessions Rating of Perceived Exertio
SVM	Support Vector Machine
UAL	Universidade Autónoma de Lisboa
UEFA	Union of European Football Associations
VAR	Video Assistant Referee

1 Introdução

O Futebol é um desporto que exige muito esforço físico e mental dos jogadores para se obter o sucesso pretendido. Com o avanço das novas tecnologias e surgimento de novas ferramentas, os clubes e o seu *staff* tentam compreender como podem tirar proveito destas para extrair informações sobre o desempenho dos jogadores e em como diminuir o risco dos seus atletas contraírem lesões. As lesões são a principal razão pela ausência dos atletas nos jogos, o que pode afetar o desempenho de uma equipa durante uma temporada [1]. Verifica-se também que a maioria das lesões são provocadas sem nenhum contato, o que pode demonstrar que existem certos fatores decisivos que possam estar por de trás destas lesões, tal como a idade ou o número de jogos disputados, que podem pôr um jogador em risco de lesão [2].

Nos dias de hoje, com o avanço tecnológico, os clubes de futebol têm como desafio perceber vários aspetos dentro do jogo e saber como podem tirar partido dos dados obtidos dos seus atletas para conseguirem explorar como melhorar vários aspetos para o coletivo, como prevenir aspetos que as possam prejudicar [3].

1.1 Formação do Problema

O problema que se pretende resolver nesta dissertação é: Como os clubes de futebol conseguem antecipar possíveis lesões dos seus jogadores?

Para resolver este problema, precisam de ser identificadas as métricas para construir-se um modelo de ML (*Machine Learning*) com dados históricos para no futuro os clubes de futebol saberem com precisão quais dos seus jogadores estão em risco de lesão muscular.

1.2 Objetivos

O objetivo geral é analisar de que maneira os clubes de futebol podem prevenir que os seus jogadores sofram lesões, verificando algumas variáveis, de maneira a evitar o comprometimento do sucesso desportivo devido a lesões.

1.2.1 Objetivos específicos

Os objetivos específicos desta dissertação são:

- Perceber como os clubes tiram partido das tecnologias para analisarem os seus jogadores.
- Compreender as análises feitas do *staff* do clube aos seus jogadores.
- Compreender por que os jogadores atualmente lesionam-se tanto.
- Identificar as métricas avaliadas para antecipar possíveis lesões dos jogadores.

- Perceber que tipo de algoritmo seria mais apropriado para resolver este tipo de problema.

1.3 Justificação

Tenciona-se realizar a investigação deste tema, já que além de ser uma área bastante motivadora para o autor e não só, sabe-se que no futebol existe uma grande área de análise de dados, para que o *staff* tenha a melhor informação disponível sobre eles, como avaliação física e desempenho individual, assim como coletivo, relatórios sobre os adversários, informação sobre futuros talentos que podem vir a ser "estrelas" e muito mais. Pretende-se também realizar esta investigação para adquirir mais conhecimento nesta área e que estes conhecimentos que forem colhidos neste trabalho sejam disponibilizados para futuras investigações.

Com a "sofisticação do desporto" com estas novas tecnologias que aparecem nos nossos dias, os clubes de futebol dispõem de ferramentas poderosas que podem ajudá-los a corrigir detalhes dentro da sua equipa, como conhecer mais ainda o seu adversário, ao utilizarem a análise de dados. Uma das principais adversidades dos clubes de futebol são as lesões dos seus principais atletas que podem os ausentar por diversas semanas ou meses, desfalcando assim a equipa. Para evitar isto, os clubes de futebol tentam perceber como antecipar quando um jogador está no seu limite físico com ferramentas analíticas.

Além disso, pretende-se adquirir mais conhecimento acerca da análise de dados dentro do futebol e perceber mais sobre a "sofisticação" do desporto.

1.4 Estrutura do documento

Este documento está estruturado da seguinte forma: o Capítulo 1 é esta Introdução; o Capítulo 2 apresenta o resultado das pesquisas de artigos e relatórios sobre lesões no futebol; no Capítulo 3 o documento aborda a utilização de *Machine Learning* para análise de dados no futebol; O Capítulo 4 apresenta a metodologia utilizada neste estudo; o Capítulo 5 mostra o cronograma seguido neste trabalho; no Capítulo 6 abordamos a integração da tecnologia para melhoria de rendimento no desporto rei; o Capítulo 7 introduz a teoria subjacente às métricas escolhidas para este trabalho; o Capítulo 8 trata do modelo ML escolhido para usar as métricas; no Capítulo 9 delineamos as nossas conclusões; e finalmente no Capítulo 10 apresentamos algumas propostas de evoluções futuras e desenvolvimento deste trabalho.

2 Lesões no futebol

As lesões no futebol fazem com que os atletas se ausentem por um ou vários dias de trabalho, falhando assim sessões de treino e até jogos. A causa da ausência de um jogador pode ser por uma lesão traumática ou por uma lesão sem contato, ou seja, lesões musculares [1].

2.1 UEFA Elite Club Injury Study Report

Num um relatório da UEFA (*Union of European Football Associations*) os principais clubes de futebol da Europa que participam na Liga dos Campeões fazem contribuições com os seus dados para oferecerem informação importante para análise e estudo para o tratamento e prevenção de lesões dos jogadores de futebol [4]. Este relatório começou a ser feito a partir de 2001 e vem sendo feito ao longo destes anos todos.

Neste capítulo, vai-se apresentar alguns dados referentes às épocas 2016/17 até à época 2019/20. Mais adiante, irão ser analisados alguns gráficos desde o início do relatório de estudo das lesões da UEFA, ou seja, desde 2001 até 2020.

Os relatórios são referentes a uma época inteira do futebol europeu, mas é de frisar que a época 2019/20 foi uma época atípica, pois houve uma pausa que foi inesperada devido ao COVID-19 (*Corona Virus Disease 2019*), portanto esta época só contém dados de Julho até Março.

Na Figura 1, podemos observar que durante as 3 primeiras épocas havia uma tendência de crescimento de lesões, podemos também analisar que na época 2019/20, com 2 a 3 meses para o fim da época, caminhava para os mesmos números ou até iria prolongar a sequência de crescimento das lesões. Podemos observar também que na época de 2016/17 as lesões sofridas nos jogos representaram 57 % de toda a época, enquanto o restante foram as lesões sofridas nos treinos. Na época de 2017/18 essa tendência aumentou para os 60% o que confirmava que os jogadores se lesionavam em quase 2/3 da época em jogos e o restante nos treinos. Na época seguinte, 47% das lesões surgiram durante os treinos, o que contrariou o que se verificava anteriormente. Porém, na época de 2019/20, as lesões durante os jogos voltaram a aumentar para 58% até o mês de Março.

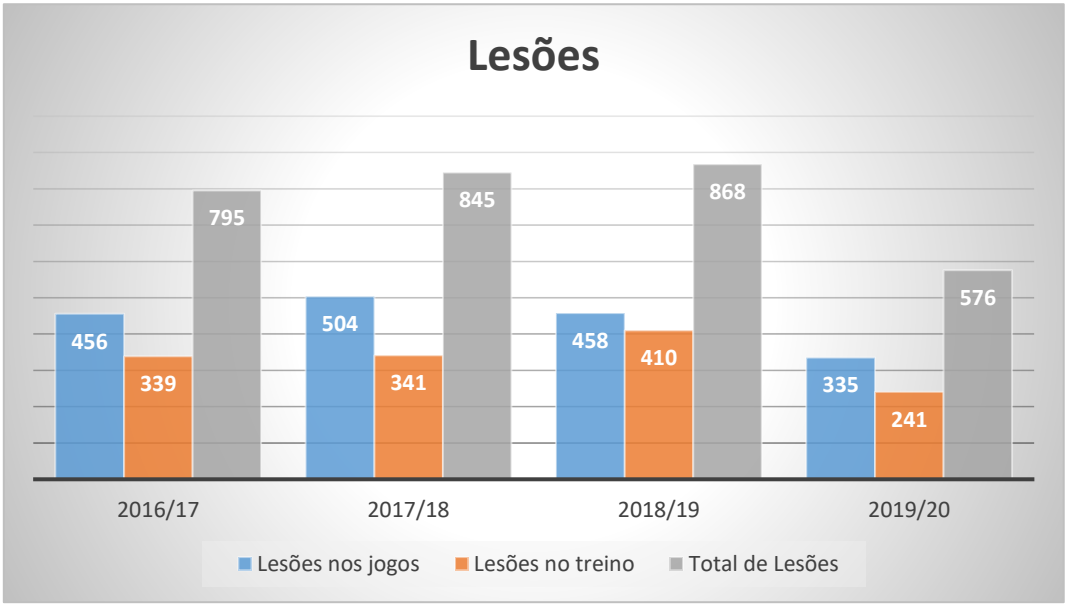

Figura 1- Lesões sofridas nos jogos e treinos em cada época (Fonte: Elaboração Própria).

2.1.1 Tipos de lesões de acordo ao relatório da UEFA

De acordo com o relatório da UEFA, são analisados 3 tipos de lesões [4]:

- **Lesões Graves**: Segundo a UEFA, são aquelas lesões que resultam em mais de 4 semanas de ausência. As lesões graves mais comuns nos últimos anos têm sido lesões nos joelhos, coxas e virilhas. E os tipos de lesões graves podem ser tanto muscular ou nos ligamentos. Na Figura 2 podemos notar que havia uma tendência crescente desde a época 16/17 que foi interrompida devido à época atípica de 19/20, mas que a tendência era de continuar a aumentar.

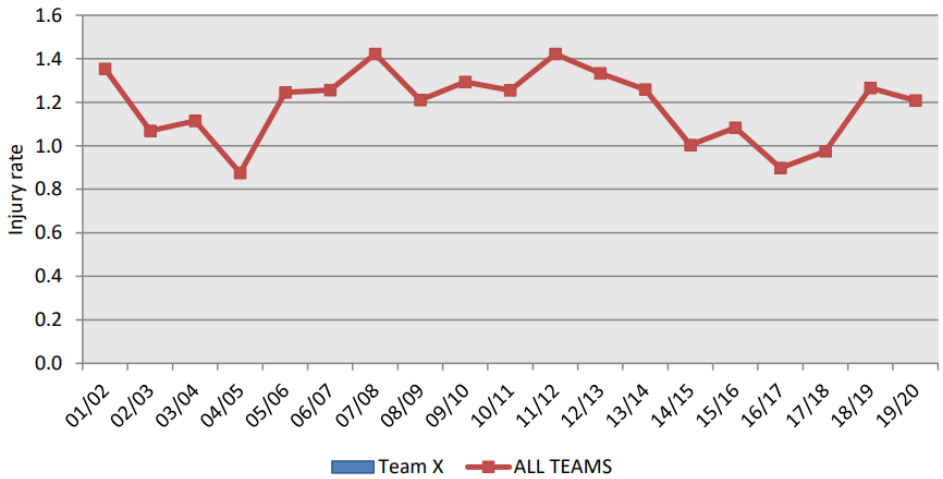

Figura 2- Taxa de Lesões Graves ao longo dos anos a cada 1000 horas de trabalho de todas as equipas juntas (Fonte: [5]).

- **Lesões Musculares**: São todas as lesões sem contato, ou seja, em que não é preciso um contato para o jogador se lesionar. As lesões musculares também estão divididas em níveis de gravidade [4].

    - Mínima (1-3 dias)
    - Leve (4 -7 dias)
    - Moderada (8 – 28 dias)
    - Grave (> 28 dias)

Na Figura 3 podemos notar que a mesma tendência que apresentava as lesões graves também se confirma aqui, entretanto se formos a comparar ao longo de aproximadamente 20 anos, podemos conferir que a taxa de lesões musculares diminuiu ligeiramente.

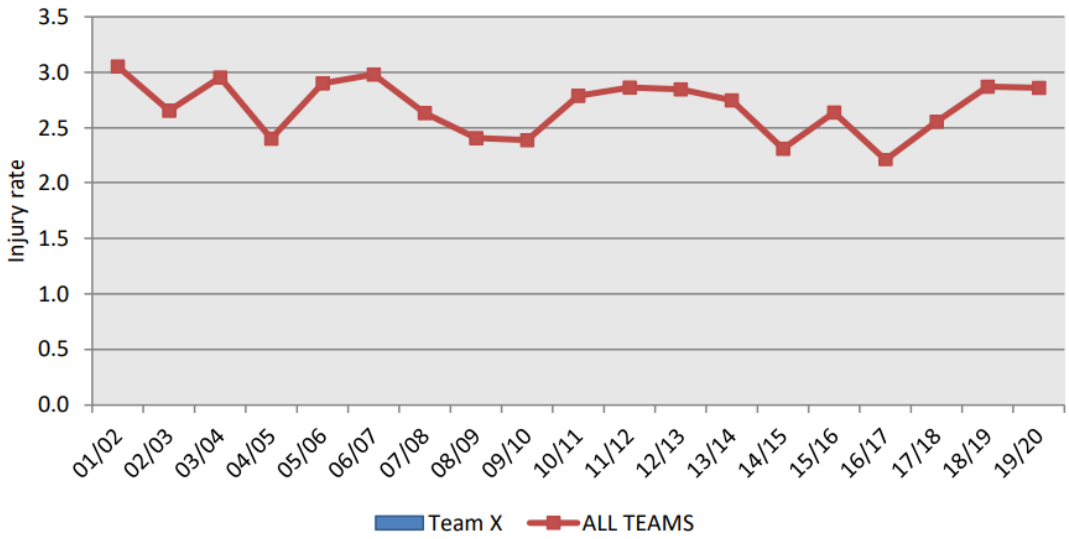

Figura 3- Taxa de lesões musculares de todas as equipas a cada 1000 horas. (Fonte: [5]).

- **Lesões nos ligamentos**: Todas aquelas lesões que afetam os ligamentos. As lesões de ligamentos mais comuns são no joelho e no tornozelo. E assim como as lesões musculares também têm níveis de gravidade [4]:

    - Mínima (1-3 dias)
    - Leve (4 -7 dias)
    - Moderada (8 – 28 dias)
    - Grave (> 28 dias)

Na Figura 4 podemos observar que a taxa de lesões nos ligamentos a cada 1000 horas de trabalho diminuíram de 2.0 para 1.0 em quase 20 anos, o que representa uma queda de 50% deste tipo de lesões a cada 1000 horas.

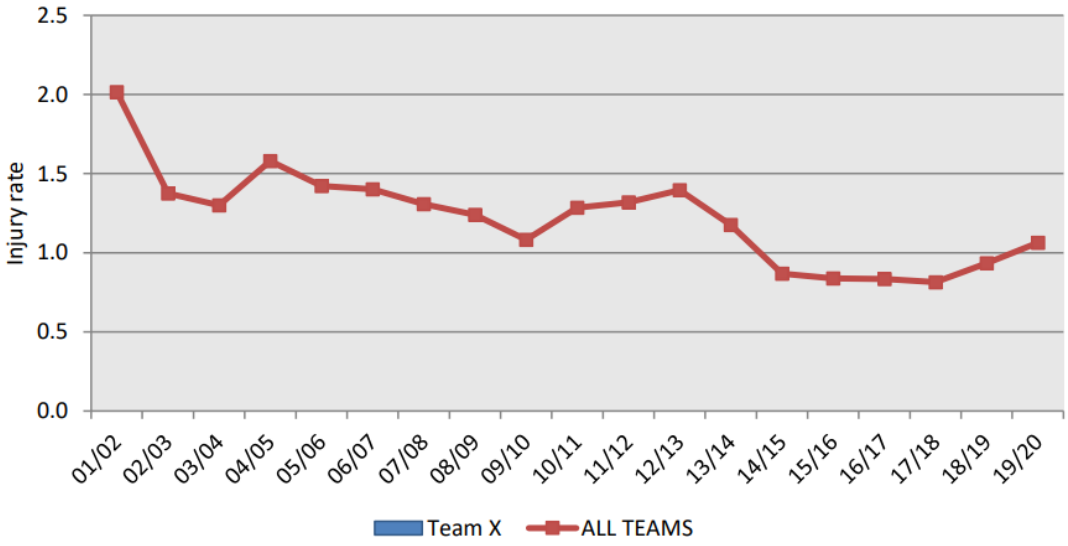

Figura 4- Taxa de lesões nos ligamentos de todas as equipas a cada 1000 horas. (Fonte: [5]).

Ao analisar estas 4 temporadas, podemos verificar que nas primeiras 3 épocas, existia uma grande tendência para as lesões musculares continuarem a aumentar, assim como as lesões graves que aumentavam ligeiramente. Já as lesões nos ligamentos, estagnaram nas primeiras duas temporadas e na terceira caíram ligeiramente.

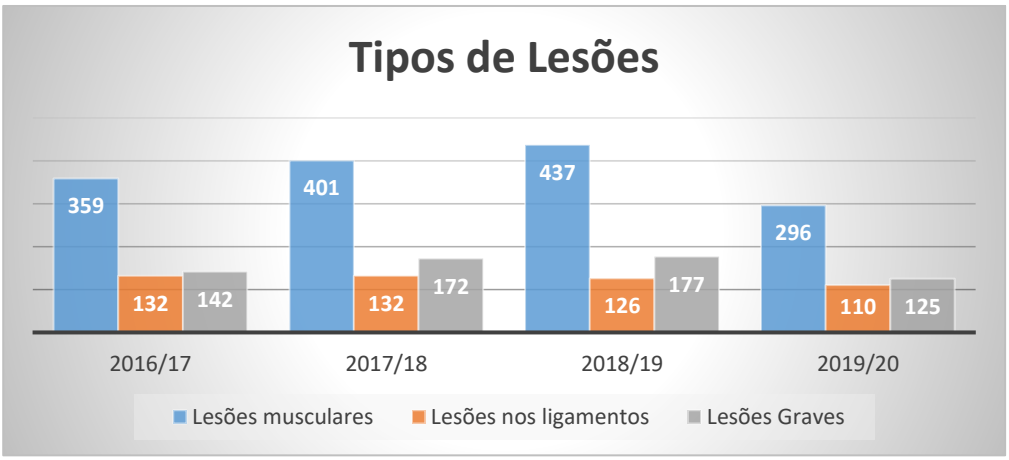

Figura 5- Tipos de lesões sofridas (Fonte: Elaboração Própria).

Consequentemente, os jogadores que se lesionavam estariam indisponíveis para os próximos treinos ou jogos, mediante a gravidade da sua lesão. Na Figuras 6 podemos analisar a taxa de com que o plantel treinasse todo junto, onde podemos observar que a média de todas as equipas fica acima dos 80%. Já na Figura 7 podemos analisar essa taxa ao longo dos anos e comparando a época 01/02 até à 19/20 podemos observar que houve um aumento de disponibilidade do plantel nos treinos, enquanto em 01/02 tínhamos menos de 75% a 79% de disponibilidade, em 19/20 temos entre 80% a 95%.

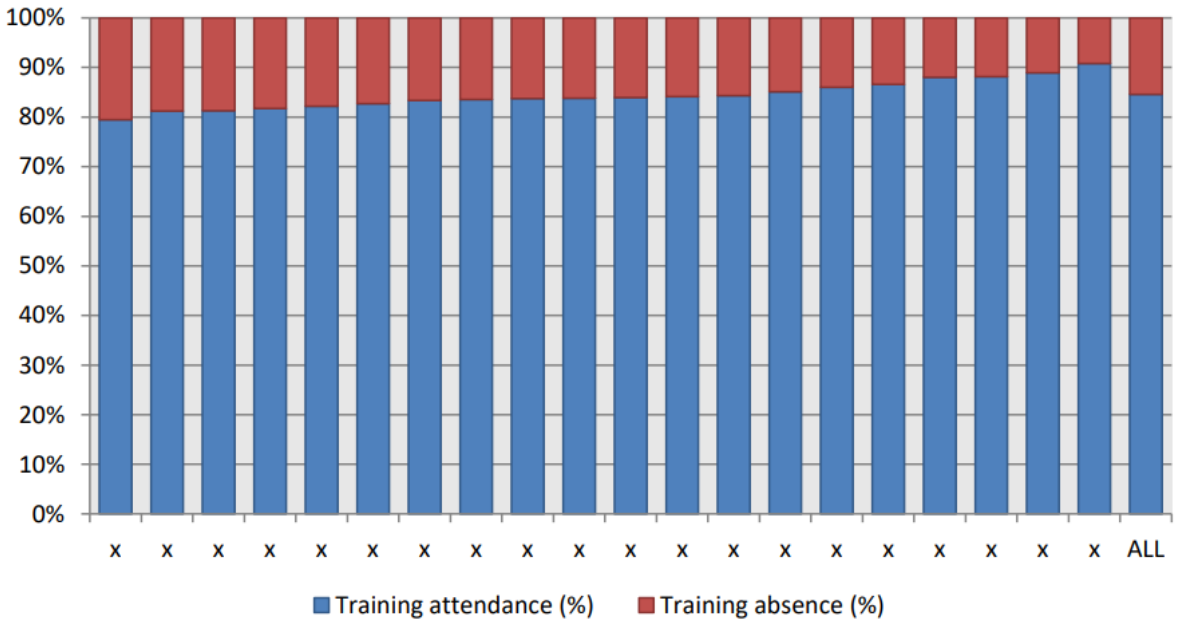

Figura 6- Taxa de disponibilidade do plantel nos treinos (Fonte: [5]).

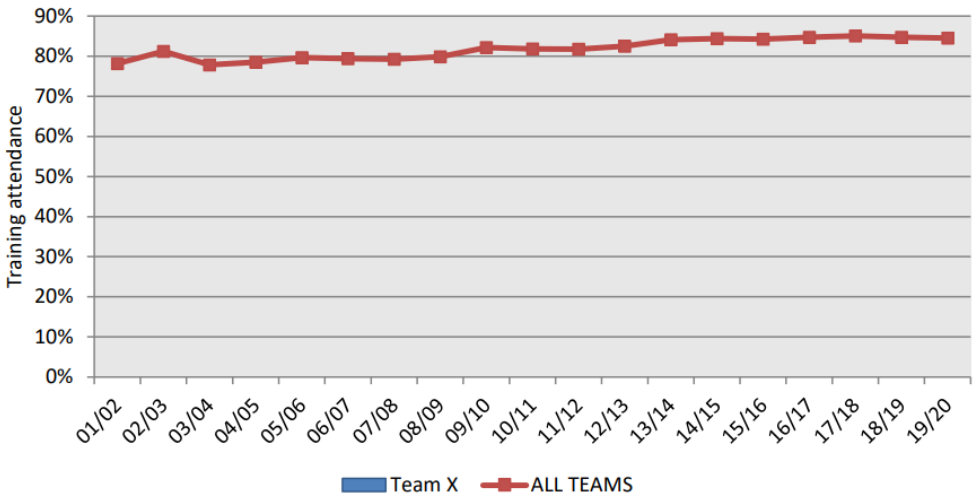

Figura 7- Taxa de disponibilidade do plantel nos treinos ao longo dos anos (Fonte: [5]).

Falando agora da taxa de disponibilidade do plantel nos dias de jogos, podemos analisar as Figuras 8 e 9 e verificar a mesma tendência que a taxa de disponibilidade dos plantéis nos treinos.

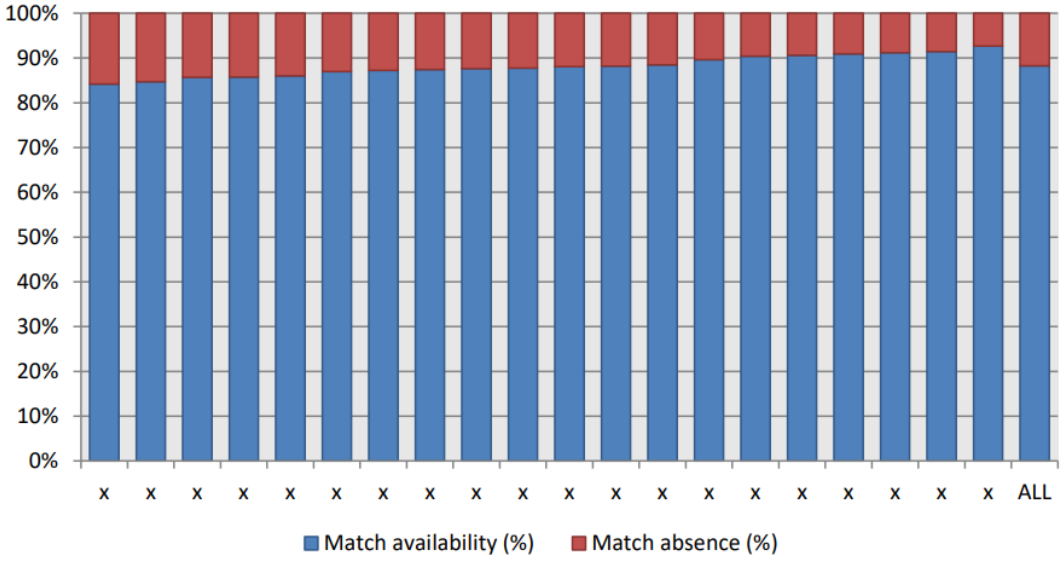

Figura 8- Taxa de disponibilidade dos plantéis nos dias de jogos (Fonte: [5]).

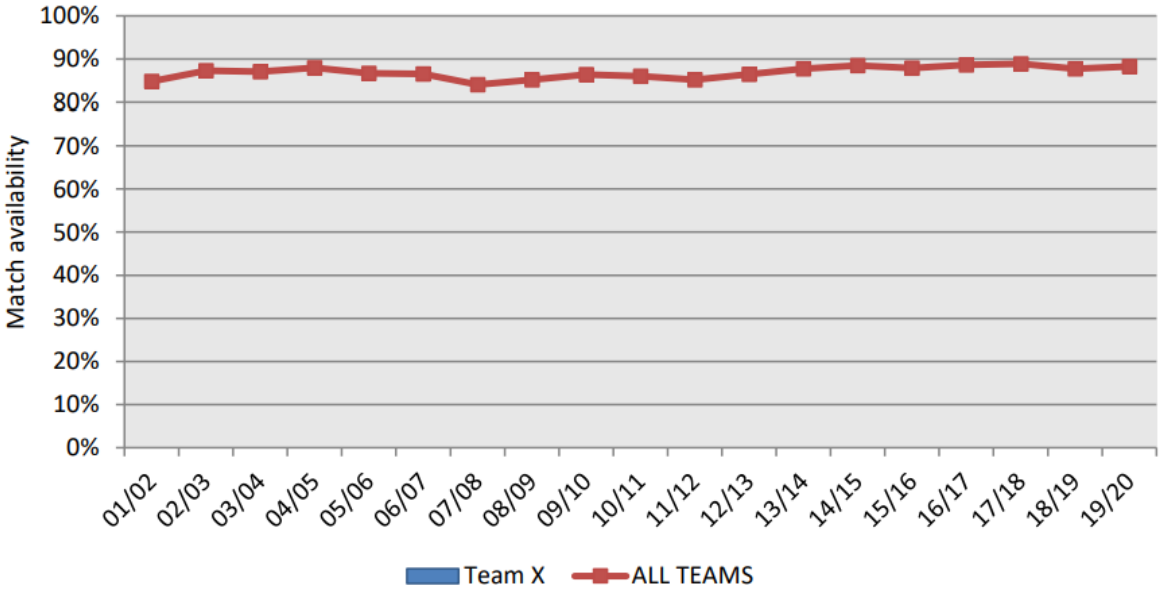

Figura 9- Taxa de disponibilidade dos plantéis nos dias de jogos ao longo dos anos. (Fonte: [5]).

A UEFA também tenta perceber quais são as razões do porquê os jogadores estarem ausentes nos dias de jogos e treinos, e nas Figuras 10 e 11 podemos perceber que a maioria das razões de estarem ausentes em média de todas as equipas abrangidas por este estudo é devido

às lesões, a segunda razão é de estarem a representar os seus países, nas seleções, ou seja, a jogarem pelo seu país, em seguida por motivos de doença.

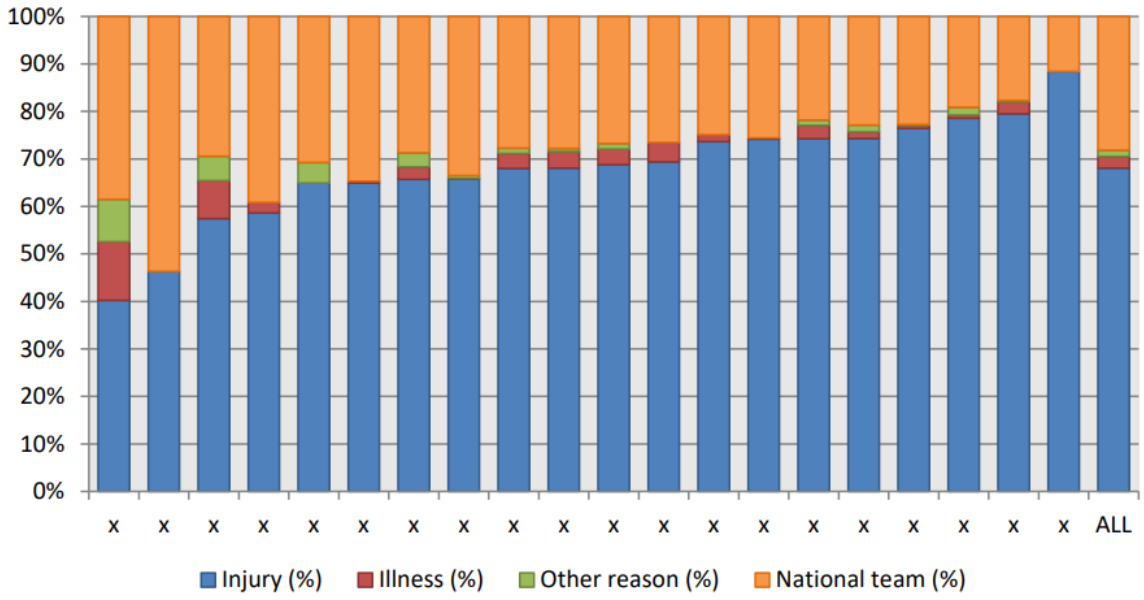

Figura 10- Razões das ausências nos treinos. (Fonte: [5]).

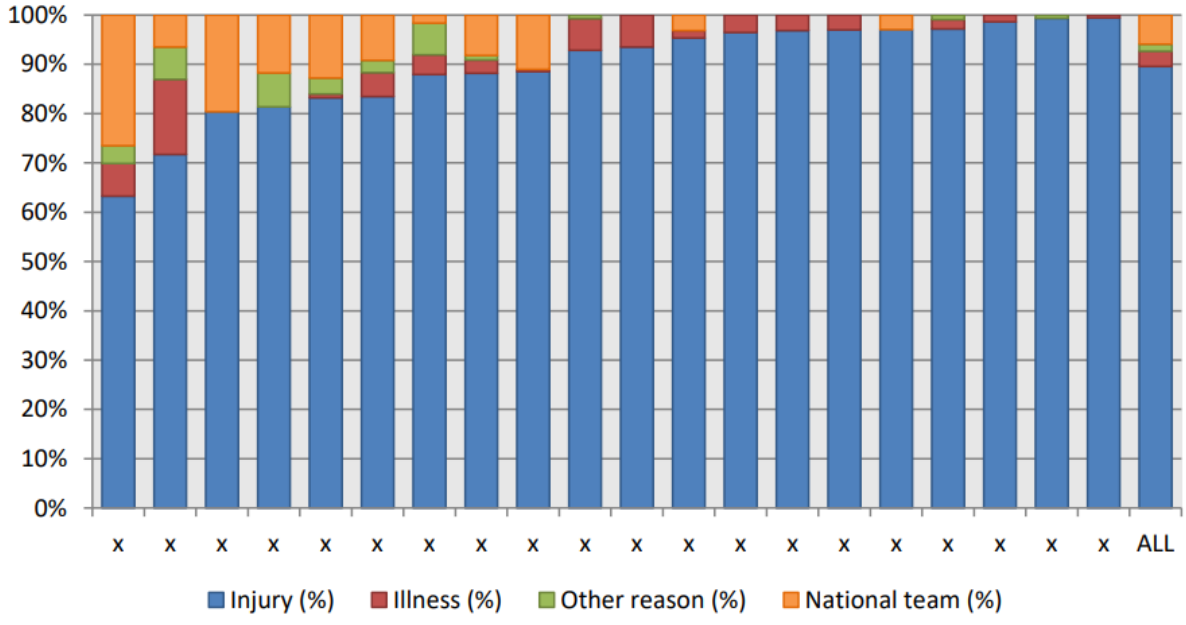

Figura 11- Razões das ausências nos jogos (Fonte: [5]).

Por último, referimos ainda a proporção que um jogador tem de se lesionar na mesma época ao longo dos anos. Como podemos ver na Figura 12, em quase 20 anos, a proporção de um jogador lesionar-se novamente na mesma época, caiu para quase metade, já que na época

01/02 tínhamos quase uma taxa de 16% de um jogador em todas as equipas do estudo, em voltar a lesionar-se e em 19/20 tínhamos uma taxa de aproximadamente 9%. Mas como foi uma época atípica a de 19/20, podemos comparar com a anterior que foi a 18/19, onde continuamos a observar um decréscimo de aproximadamente 4%.

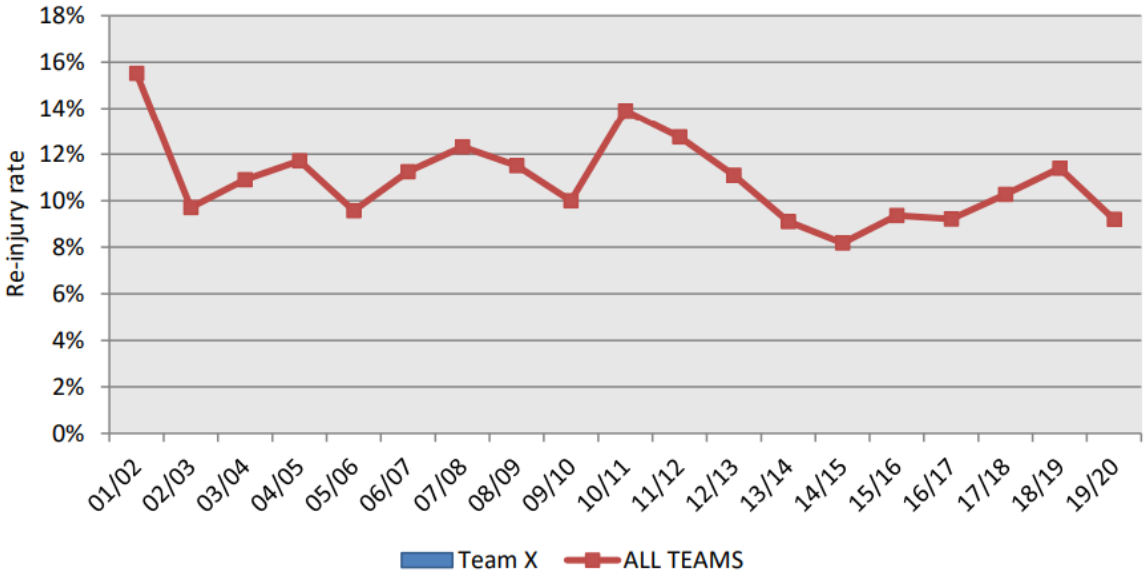

Figura 12- Taxa de um jogador voltar a lesionar-se ao longo dos anos (Fonte: [5]).

2.2 Tipos de lesões

- **As lesões traumáticas** são aquelas que não conseguimos prever, em que um jogador sofre de um contato provocado pelo jogador da equipa adversária durante o jogo [6].
- **As lesões sem contato** podem ocorrer devido à sobrecarga física ou à pouca preparação física, provocadas corridas de alta intensidade durante os jogos ou durante os treinos [7].

Algumas destas lesões fazem com que os jogadores fiquem longos períodos fora dos relvados e em alguns pontos chegam mesmo a terminar com a carreira de alguns jogadores. Com isto as equipas tentam hoje antecipar e acompanhar mais os seus jogadores com os diferentes departamentos, de maneira a rastrearem todos os seus passos para estudarem os seus corpos, evitando assim perderem os seus melhores jogadores com a utilização de dispositivos e software de IoT (*Internet of Things*).

2.3 Lesões mais comuns

Dentro das lesões mais frequentes de acontecerem aos atletas podemos incluir os dois tipos de lesões acima citados. Do lado das lesões traumáticas temos a contusão, embora seja uma lesão que só em alguns casos deixa os atletas no boletim clínico por alguns dias. No entanto, as lesões mais comuns e que normalmente fazem com que os jogadores estejam fora dos treinos e dos jogos são as lesões sem contato, dependentemente do nível de competição em que se apresentam, género e idade e são [8]:

- **Entorse do tornozelo**: Neste caso, a entorse acontece sem contato. Ocorre quando o atleta apoia mal o pé no solo, durante uma corrida, um salto ou mesmo em um corte para tentar desarmar o adversário, fazendo com que torça os ligamentos do tornozelo, causando na maioria das vezes um grande inchaço na zona.
- **Entorse do joelho**: Desportos como o futebol que envolvem este tipo de acelerações, desacelerações, grandes movimentos de rotação do atleta, fazer um corte ou mesmo em um mau apoio, seja o suficiente para causar a tensão suficiente de maneira a romper os ligamentos que estão à volta do joelho.
- **Estiramento dos isquiotibiais**: Devido à alta intensidade de carga de treino e nos dias de jogo que alguns atletas têm, resultam em distensão do músculo isquiotibial (músculo da coxa). Esta lesão é a mais comum entre todos os jogadores, desde os mais jovens até aos profissionais e é responsável por cerca de 15% a 20% de todas as lesões no futebol [9]. Sendo o bíceps femoral o mais afetado dos 3 músculos dos isquiotibiais.
- **Estiramento da virilha**: É idêntico ao estiramento dos isquiotibiais, mas desta vez na virilha (parte interna da coxa), causado pela carga de treinos de alta intensidade somada aos jogos em que o nível de exigência deve ser máximo. O atleta pode sofrer um estiramento quando estica demais o músculo e pode sentir o músculo a rasgar, algumas fibras ou mesmo com uma rutura completa.

2.4 Fatores principais das lesões musculares

Os principais fatores das lesões musculares vão depender muito de jogador para jogador e de parâmetros como a idade, o género do atleta, a fadiga, número de partidas jogadas, distância percorrida por cada jogo e treino, velocidade média percorrida por jogo e treino, número de

sprints, histórico de lesões, a carga nas sessões de treino. Um atleta com uma carga de treino de alta intensidade tem mais probabilidade de se lesionar, devido à exposição a acelerações e desacelerações de alta intensidade [10]. O nível das provas disputadas pelos jogadores também influencia, em competições como o Campeonato do Mundo de seleções os jogadores têm mais probabilidade de se lesionarem do que nas suas ligas nacionais [2].

2.5 A tecnologia e a prevenção de lesões

Podemos citar um caso específico, o do Sport Lisboa e Benfica que investiu milhões em tecnologia em ciência de dados, quando começou a coletar dados a equipa principal tinha sido alvo de 8 lesões graves e uma temporada depois de coletar dados, as lesões graves diminuíram apenas para 3 [6].

Nos casos em que as lesões são traumáticas, a situação é mais complexa, porque estes tipos de lesões são causadas por contato e é incerto quando um jogador irá sofrer um toque ou uma carga durante um jogo ou durante um treino.

2.6 Carga de treino

Existe uma vasta quantidade de estudos que apontam que existe uma grande relação da carga de trabalho dentro do treino com as lesões. Como já foi citado no capítulo 2.3, as lesões musculares podem ser provocadas por uma série de fatores, e esta série de fatores normalmente é inerente à carga de trabalho dentro do treino [11].

Um estudo analisou vários atletas de várias modalidades como ciclismo, corredores de atletismo e skaters, que aumentaram a carga de treino em 10x e tiveram um aumento de 10% na performance. Diferentemente dos desportos com contato como o futebol e outros desportos coletivos que são caracterizados por várias acelerações, sprints, desacelerações e mudanças de direção repentinas, o que faz com que se puxe mais pelos músculos dos atletas, este estudo demonstrou que quanto mais pesado seja a carga de treino dos jogadores maiores são as chances dos jogadores lesionarem-se, e confirmou-se que menos carga de treino diminui-se a chance de sofrerem lesões. Porém, também se apresentou que os atletas de desportos de colisão que não concluem um determinado período de pré-época aumentam as chances de sofrerem lesões [12].

A carga de treino pode ser medida de duas formas [13]:

- **Cargas de treino internas**: São as medidas biológicas como a frequência cardíaca, lactato sanguíneo e níveis de oxigénio. Estes dados são coletados por sensores *wearables* [10].

- **Cargas de treino externas**: São aqueles medidos quando o atleta está em atividade e medem a potência, velocidade, aceleração, análise do movimento. Estas cargas são medidas através de GPS (*Global Positioning System*), enquanto os atletas estão com os seus coletes durante os treinos e jogos.

Existe uma pesquisa que sugere que a taxa de carga de trabalho aguda/crónica, conhecida em inglês como ACWR (*Acute:Chronic Workload Ratio*), proteja os atletas contra lesões [13]. É o mais popular e mais pesquisado modelo do processo de lesões [10], este método auxilia a monotorização do desempenho, saúde e prevenção de lesão dos atletas. Há uma relação dentro da carga de trabalho aguda com a carga de trabalho crónica. Segundo Bruno Mendes (chefe do desempenho humano do Benfica em 2016), autor de uma pesquisa publicada no *Journal Of Science and Medicine in Sport in 2016*, a taxa de carga de trabalho aguda é uma rajada de atividades que se cumpre em uma semana que é muito mais alta que as próximas 4 semanas em média, que é a taxa de carga de trabalho crónica. Nesta pesquisa chegou-se à conclusão que, se os jogadores aumentarem a carga de trabalho crónica entre os jogos, ficam menos suscetíveis a lesões, ou em caso de algum fator que faça com que a carga de trabalho crónica diminua, o treinador pode limitar a carga de trabalho aguda [6]. Quando o ACWR está entre os valores de 0.8 a 1.3, ou seja, por exemplo, se a carga de trabalho aguda for semelhante à carga de trabalho crónica, o risco de lesão ocorrer é baixo, mas se estiver igual ou acima de 1.5, ou seja, se a carga de trabalho aguda for muito maior que a carga de trabalho crónica, esse risco aumenta [13], como mostra a Figura 13.

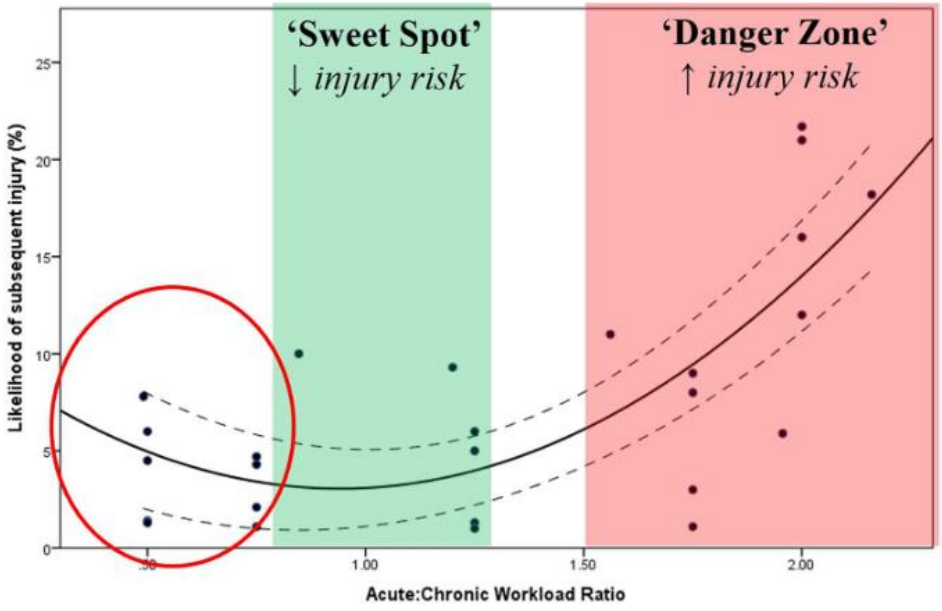

Figura 13- Relação entre a taxa de carga de trabalho aguda/crónica e o risco de lesões (Fonte: [13]).

3 Futebol e ML

A modernização do futebol fez com que as equipas utilizassem ferramentas que coletam dados físicos, técnicos e psicológicos mais detalhados dos seus jogadores, através de sistemas de rastreamento e sensores, com a finalidade de serem usados para *scouting* (processo de observação para recolher informações individuais ou coletivas), análise de desempenho, táticas e também para entenderem melhor as lesões sofridas pelos seus jogadores [10]. Estas técnicas são utilizadas para analisar, tentar perceber o risco de lesão de um jogador com base em certas métricas e ajudar a prever utilizando modelos estatísticos de prevenção de lesões. Neste campo existem poucos estudos, ou seja, estudos limitados onde são exploradas poucas variáveis para analisar o risco de lesões, sem explorar um grande padrão dos dados que estão disponíveis, pois os dados relacionados à atividade física dos jogadores têm sua disponibilidade limitada pelos clubes [14].

Além da potencial má performance da equipa, as lesões trazem consigo consequências financeiras negativas aos clubes de futebol [1]. Usando certos modelos, podemos estimar a probabilidade destas lesões, incluídas aqui todas aquelas lesões sem contato, ou seja, lesões musculares. E é aqui onde o ML pode entrar em ação para permitir a análise de certas métricas e ajudar a prevenir este tipo de lesões [15].

Voltando ao caso do Sport Lisboa e Benfica, o ML é utilizado juntamente com a análise preditiva dentro do Microsoft Azure ML, pelo *staff* da equipa. Os jogadores são monitorizados com diversos sensores e aparelhos, permitindo encontrar os seus pontos fortes e pontos fracos, o que permite uma adaptação para reduzir bastante as chances de sofrerem lesões [6]. Apesar de serem utilizados muitos sensores que trabalham juntamente com o ML e a Análise Preditiva e que recolhem bastantes dados, alguns dados têm de ser recolhidos manualmente, como a dieta dos jogadores e alguns atributos psicológicos, que não se consegue recolher automaticamente durante os treinos ou jogos [6].

Neste caso do Sport Lisboa e Benfica, um dos principais objetivos na altura era de desenvolver um modelo preciso para prever quando é que um jogador estava em risco de sofrer uma lesão e assim antecipar essa ocorrência, iniciando um tipo de trabalho específico e possivelmente figurar como suplente em alguns jogos no começo da partida [6]. Na Figura 14 exemplificamos alguns tipos de dados recolhidos.

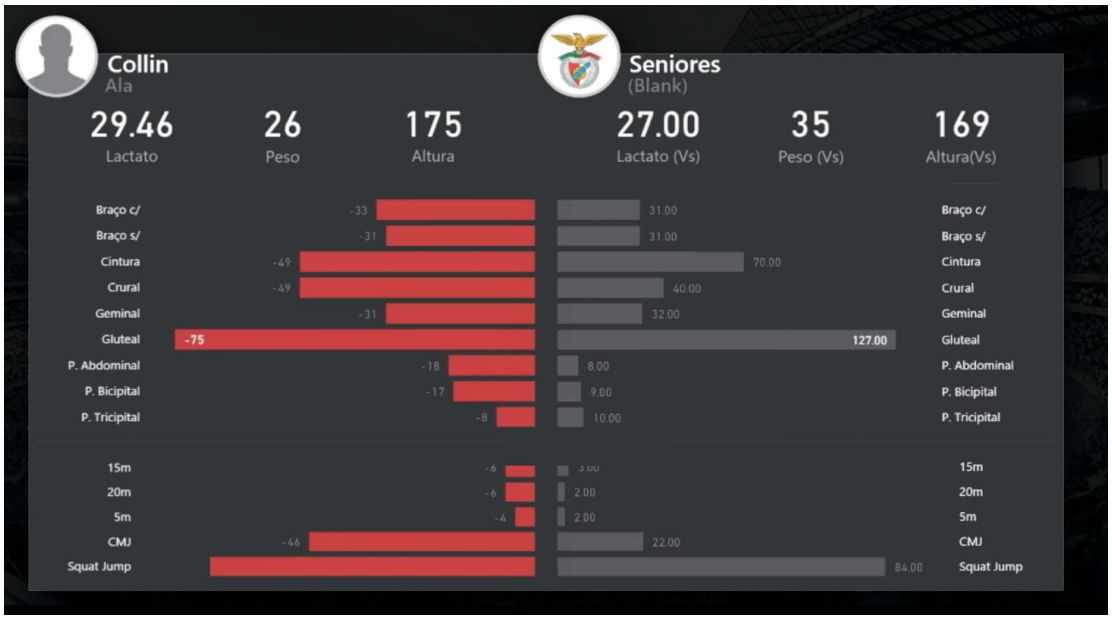

Figura 14-Tipo de dados recolhidos pelo Benfica (não são dados reais) (Fonte: [6])..

3.1 Machine Learning

É um ramo da inteligência artificial que estuda cientificamente a matemática e modelos estatísticos, de modo a habilitar os computadores a melhor utilizarem os dados de forma automática, tal como no reconhecimento de padrões, e para melhorar as suas decisões. É aplicada em diversos setores como o da saúde, a indústria financeira, na bolsa de valores, no desporto e muito mais [10]. Podemos dizer que também tem a capacidade de aprender e melhorar com a experiência, sem ser propriamente programado.

O ML pode ser muito útil para desempenhar tarefas muito complexas e difíceis para os humanos [15]. O principal objetivo é permitir que as máquinas aprendam automaticamente sem intervenção dos humanos e sem assistência.

O ML é categorizado em métodos supervisionados (onde está classificado em regressão e classificação) e não supervisionados (onde está classificado em *clustering* e associação).

3.1.1 Aprendizado Não Supervisionado

Este aprendizado também trabalha com o reconhecimento de padrões, mas desta vez os dados de entrada são não rotulados e não têm alguma variável de saída correspondente [10]. Neste caso a variável seria se está lesionado ou não.

3.1.2 Aprendizado Supervisionado

Este aprendizado utiliza exemplos em dados rotulados de entrada e saída e que foram utilizados para a máquina "aprender" e assim prever acontecimentos futuros. No caso da previsão de lesões, como variáveis de entrada teríamos as cargas de treino e como variáveis de saída teríamos a ocorrência de lesões [10].

Para esta dissertação, será necessário basearmo-nos neste tipo de aprendizado, pois utiliza um *dataset* com uma variável de saída que vamos saber identificar. Neste aprendizado, um grande volume de dados de treino irá permitir que o algoritmo de ML aprenda com os dados e identifique padrões complexos e não lineares (se houver algum padrão detetável) [16].

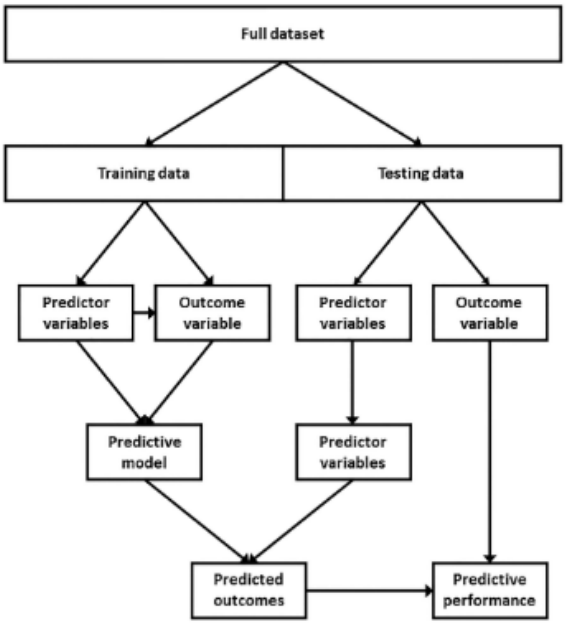

Figura 15-Abordagem da aprendizagem supervisionada (Fonte: [16]).

3.1.3 Tipos de algoritmos para construir modelos preditivos

Existem diferentes tipos de algoritmos que se podem usar para construir um modelo preditivo. Alguns algoritmos têm as suas funções matemáticas assim como os seus parâmetros de forma a serem utilizados com dados adequados, pois diferentes algoritmos são para serem utilizados para casos específicos de dados, apesar de que é sempre bom comparar o seu desempenho. Esses algoritmos são:

- **Naive Bayes**: É um classificador probabilístico muito utilizado para classificação textual em ML, baseado no "Teorema de Bayes", criado por Thomas Bayes [16].

- **Regressão Logística**: É usado em tarefas de classificação e é semelhante à regressão linear, para prever uma variável dependente categórica e estimamos a probabilidade da variável dependente. Há 3 tipos de regressões logísticas [16]:
    - Regressão Logística Binária.
    - Regressão Logística Multinominal.
    - Regressão Logística Ordinária.
- ***Random Forest***: É um classificador que é baseado em várias árvores de decisões, onde cada árvore é construída adotando um algoritmo A, onde o conjunto de dados de treinos S e um vetor aleatório adicional, θ, onde, θ é amostrado e a previsão do *random forest* é adquirida pela maioria de juízos sobre as árvores de decisão individuais [17].
- **K-NN (*K-Nearest Neighbors*)**: Um dos algoritmos mais simples dos algoritmos de ML. Basicamente este algoritmo tenta memorizar o conjunto de dados de treino e tentar prever com base nos rótulos dos vizinhos mais próximos dentro do conjunto de dados de treino, o rótulo da próxima instância [17].
- **Rede Neural**: É uma classe dos algoritmos de ML que são utilizados para mapear dados complexos [16]. É inspirado nas redes neurais do cérebro [17].
- **SVM (*Support Vector Machine*)**: A ideia do SVM é determinar a superfície de decisão com a maior margem possível, ou seja, a superfície de decisão cujo exemplo mais próximo está o mais longe possível [17].
- **Árvore de decisão**: É uma abordagem de modelagem de previsão usada em ML bastante simples, pode ser bem poderosa e é utilizada como classificador. São construídos através de uma raiz da árvore e vai criando vários nós abaixo, como se fossem as folhas dessa árvore [18], como exemplificado na Figura 16.

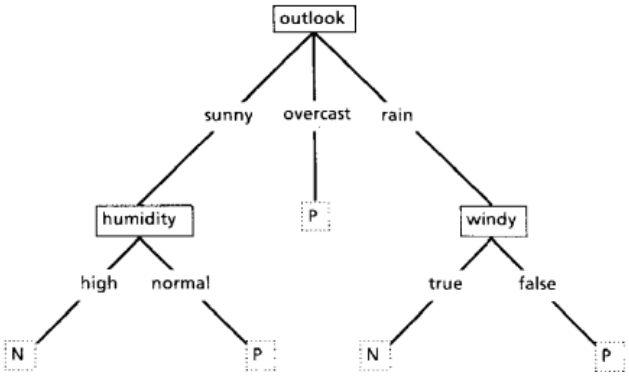

Figura 16- Árvore de decisão simples (Fonte: [18]).

4 Metodologia

Neste capítulo explicamos as abordagens tomadas para atingir os objetivos desta dissertação. Podemos dizer que houveram dois tipos de pesquisa: a pesquisa quantitativa exploratória e a pesquisa qualitativa descritiva.

Fez-se uma pesquisa quantitativa exploratória para melhor compreender sobre como os clubes de futebol conseguem tirar partido do ML para benefício das suas equipas, as lesões dos jogadores de futebol, quais são os tipos de lesões que mais afetam os jogadores e também tentar saber como fazem os profissionais para prevenir estas lesões.

Fez-se uma pesquisa de vários artigos na base de dados "b-on", também foram pesquisados artigos no Google Scholar, Research Gate com estes temas, onde os principais títulos pesquisados para obter resultados preteridos foram palavras-chaves como:

- *"Soccer players injuries"*
- *"Predicting Soccer Players Injuries"*
- *"Machine Learning to predict soccer players injuries"*
- *"Injury prevention in soccer"*.

Por não se encontrar *datasets* para compreender de forma prática os modelos em ML para prevenir as lesões, iniciou-se com este tipo de pesquisa para tentar perceber como tudo funciona, desde os conceitos básicos de lesões e tipos de lesões. As equipas de futebol só muito recentemente começaram a utilizar certas técnicas de ML para ajudar a detetar o risco de lesão de um jogador. Esta pesquisa também foi utilizada para saber mais a fundo os fatores que levam os jogadores a terem este tipo de lesão e compreender os tipos de trabalhos feitos nos treinos que podem ajudar a prevenir certos tipos de lesões.

Já na pesquisa quantitativa exploratória, teve-se em conta a análise de dados secundários de outros artigos e que utilizavam diferentes técnicas para fazerem os seus modelos de ML. Não foi feita uma pesquisa exaustiva sobre esses modelos, apenas tentamos perceber que variáveis foram utilizadas em cada caso e se havia algum tipo de padrão para se iniciar este modelo. E foi verificado que os modelos observados tinham poucas variáveis comuns e muitas diferentes, com vários tipos de algoritmos a serem executados.

Para obter respostas aos objetivos traçados no princípio desta dissertação, houve necessidade elaborar algumas considerações:

- **Objetivo específico 1:** Perceber como os clubes tiram partido das tecnologias para analisarem os seus jogadores.
    - Não adianta investir em tecnologia sem perceber o que ela faz e não ter recursos humanos qualificados o suficiente para extrair o melhor desta tecnologia.
    - Para responder a este objetivo, houve necessidade de perceber quais as áreas em que os clubes consideravam conseguir vantagem competitiva para beneficiarem com a tecnologia.

- **Objetivo específico 2**: Compreender as análises feitas pelo *staff* do clube aos seus jogadores.
    - Este objetivo é a continuação da resposta do objetivo anterior.
    - Deve-se compreender como os jogadores de futebol poderiam adaptar-se dentro de campo.
    - Assim, melhorar a sua performance e explorar na totalidade as suas capacidades físicas e psicológicas.

- **Objetivo específico 3**: Compreender por que os jogadores atualmente lesionam-se tanto.
    - Para este objetivo, foi investigada a razão do porquê que os jogadores se lesionam bastante atualmente.
    - Também foi investigado que tipos de lesões existem e quais são as mais frequentes e os fatores da ocorrência destas lesões.

- **Objetivo específico 4**: Analisar as métricas avaliadas para antecipar possíveis lesões dos jogadores.
    - Para responder a este objetivo, teve primeiro de se responder e compreender o objetivo anterior a este.
    - Só assim poderíamos saber quais são os fatores da ocorrência destas lesões.
    - E depois analisar as métricas principais para antecipar as lesões dos jogadores.

5 Tecnologia no futebol

5.1 Tecnologia no futebol em geral

Com espaço para melhorarem o seu rendimento dentro do campo, os clubes de futebol juntamente com as suas equipas técnicas queriam entender como a sua equipa se portava no relvado, aperfeiçoando os melhores movimentos dos seus jogadores e corrigindo os seus erros. Foi então na década de 1980 que surgiu a hipótese de a equipa técnica dos clubes de futebol conseguir visualizar seus jogos várias vezes e de ângulos diferentes, o que lhes dava novas perspetivas táticas para resolver alguns problemas e arranjar de imediato soluções [19].

5.1.1 Monitorização da frequência cardíaca

Foi assim que começou a análise das equipas técnicas aos seus jogadores, o que fez com que o futebol fosse revolucionado tanto tática como fisicamente como vamos abordar aqui.

Na década seguinte, os clubes de futebol conseguiram dar mais um passo em frente na análise física dos seus jogadores, pois surgiu a monitorização de frequência cardíaca que até os dias de hoje, juntamente com os aparelhos de GPS, servem para dar *insights* de cargas internas possibilitando que os clubes observem as medidas biológicas dos jogadores. Neste caso, a monitorização de frequência cardíaca serviu para os clubes recolherem dados do esforço dos seus jogadores e saberem a frequência cardíaca após os jogos e treinos, numa altura em que alguns jogadores de futebol morriam por paragens cardíacas sem se saber o porquê. O que foi não só para análise e evolução dos jogadores como também uma questão de saúde.

5.1.2 Sistemas de Posicionamento Global (GPS)

Já no final dos anos 1990 e início dos anos 2000, surgem coletes com os GPS para recolher os dados e analisar detalhadamente o desempenho dos jogadores durante os treinos e durante os jogos. Estes dispositivos são fabricados com uma taxa de amostragem de 1,5 e 10 Hz, que condicionam a unidade de tempo em que os dados são coletados [20]. Servem também para coletar os dados de carga externa, que por outras palavras são os dados físicos dos jogadores como distância percorrida, velocidades máximas, acelerações e desacelerações durante um jogo ou treino. Estes aparelhos foram introduzidos no futebol para tentar extrair dados com o propósito de os analisar e obter o maior rendimento dos jogadores ao longo da época. Estes dispositivos são muito importantes na medida de prevenir lesões nos jogadores de futebol, pois as equipas técnicas conseguem observar quando um jogador está em risco ou não de se lesionar ao analisar estes dados. Como já foi dito anteriormente, atualmente no futebol, a

maioria das lesões têm sido provocadas por eventos sem contato, o que pode ser controlado durante os treinos e jogos pelas equipas técnicas. São utilizados também para se observar e identificar em que espaços pisa cada jogador em média durante os jogos de maneira a saber-se quais são os seus pontos fortes e quais são os seus pontos fracos, tentando assim melhorar o seu desempenho.

"De acordo com o SimpliFaster (https://simplifaster.com/, 2017), há 4 tipos de sensores usados em dispositivos que rastreiam jogadores atualmente: um acelerómetro, um giroscópio, um magnetómetro e um módulo GPS" [21].

Mesmo assim, não há dentro dos clubes de futebol um dispositivo que se prove 100% assertivo e confiável, há sempre margens de erro, mas mostra-nos que há ferramentas bastante poderosas para auxiliar bastante os clubes de futebol e as equipas técnicas na tomada de decisões importantes com os seus atletas.

5.1.3 Técnicas de amostras de saliva e crioterapia

No começo dos anos 2010, por uma questão de saúde, os clubes de futebol começaram a recolher amostras de saliva e alguns clubes de futebol optaram por utilizar câmaras de crioterapia, que são câmaras com temperaturas negativas onde os jogadores entram por alguns segundos ou poucos minutos para ajudar mais rapidamente na recuperação dos músculos dos atletas.

5.1.4 VAR

E nos últimos anos, para a melhoria do jogo em si, de forma a trazer mais verdade desportiva, auxiliando os árbitros a melhorar a tomada de decisão em lances duvidosos ou em casos em que por uma questão de visibilidade não ia conseguir tomar a melhor decisão, que sem a tecnologia seria muito difícil de evitar, foram introduzidas algumas novas tecnologias como a tecnologia VAR (*Video Assistant Referee*). O VAR foi introduzido na maioria das ligas de futebol mundial na época 2018/19. Esta tecnologia é composta por duas configurações diferentes. Há os sistemas de VAR que se dizem completos e que utilizam no mínimo 4 câmeras até um número ilimitado de câmeras; e o VAR light que utiliza de 4 a 8 câmeras [22]. O que o VAR faz é auxiliar o árbitro principal na tomada de decisão de um lance duvidoso de uma outra perspetiva da qual o árbitro principal julga. O VAR pode interferir nas seguintes situações:

- Golos: Um golo pode ser anulado pelo VAR se no decorrer dessa jogada ocorrer uma irregularidade, como uma falta, fora-de-jogo ou a saída da bola pelas 4

linhas. Pode ser validado um golo também caso aconteça o contrário, no caso de o árbitro anular um golo onde não houve qualquer irregularidade o VAR também pode interferir nesta decisão, fazendo com que o árbitro principal reverta a sua decisão.

- Grande Penalidade: Caso o árbitro principal não assinale ou assinale uma grande penalidade onde o VAR entende que seja ou não, há uma comunicação entre o árbitro principal e o VAR de maneira que o árbitro principal vá rever as imagens.
- Expulsões: Caso o árbitro principal decida expulsar um jogador com o cartão vermelho direto ou no caso de houver uma situação que justifique o cartão vermelho e o árbitro não tenha dado, é logo avisado pelo VAR para rever as imagens, mantendo ou revertendo a sua decisão.

É de frisar aqui que nas situações onde o VAR intervém, chamando o árbitro para rever as imagens, quem tem o poder da última decisão é o árbitro principal e não o VAR, este apenas ajuda que o árbitro entenda melhor e veja de outro ângulo a ocorrência de uma dada jogada. Já os foras-de-jogo são analisados exclusivamente pelo VAR já que possui outra tecnologia dentro do VAR que são as linhas de fora-de-jogo. Estas são linhas traçadas por diversas câmeras que estão ao redor do estádio e que permitem ao VAR indicar se um jogador está em fora-de-jogo ou não durante um lance em que o VAR pode intervir, como exemplificado na Figura 17.

Figura 17- Análise do VAR a um golo apontado pelo Manchester City FC vs. West Ham United FC (10-ago-2019) (Fonte: [23]).

5.1.4.1 Tecnologia da linha de golo

Outra tecnologia muito importante no futebol é a tecnologia da linha de golo, que serve para verificar se a bola passou totalmente a linha da baliza ou não, decidindo-se assim instantaneamente se é golo ou não é. Esta tecnologia é composta por diversas câmeras à volta do estádio e a informação sobre se a bola entrou ou não é apresentada no relógio do árbitro principal, de maneira a decidir rapidamente. Esta tecnologia tem ainda a particularidade das várias câmeras capturarem imagens a 500 frames por segundo e enviá-las para um sistema de processamento [24]. Este sistema tem uma precisão elevadíssima, quase 100%, dando um sinal instantaneamente ao árbitro de que a bola entrou.

5.1.4.2 Bolas Inteligentes

A própria bola que é utilizada nos jogos de futebol atualmente, contém um sistema inteligente. As principais ligas europeias e americanas possuem este sistema. Esta bola que foi desenvolvida na Alemanha pela marca Adidas e pela Cairos Technologies, é embebida com um sensor NFC (*Near Field Communication*) chip [24].

5.1.5 Realidade Virtual e Aumentada

Ainda há uma década, os adeptos só conseguiam visualizar as estatísticas dadas pela emissora que transmitia o jogo, quando eram apresentadas. Quem estava no estádio só conseguia ter acesso às estatísticas individuais e coletivas pela informação que era passada pelos ecrãs gigantes. Atualmente, com a realidade virtual e a realidade aumentada, é possível os adeptos visualizarem em tempo real os dados e estatísticas sobre o jogo, tanto individuais como coletivas. Esta tecnologia ainda tem muito mais para ser desenvolvida e nos próximos anos promete envolver mais os adeptos com o jogo em si, de maneira interativa e mais emocionante.

No passado Mundial do Catar de 2022, a aplicação oficial da FIFA (*Fédération Internationale de Football Association*) ofereceu uma experiência que pode ser o indicativo de uma grande revolução tecnológica para os fãs mergulharem ainda mais numa melhor experiência do evento. O que aconteceu foi que com a aplicação, se estivéssemos no estádio, poderíamos apontar a câmera do telemóvel para o relvado e ver as informações do jogo em que estávamos em tempo real, identificando assim por exemplo um jogador e logo de seguida visualizar as suas estatísticas ou também

poderíamos visualizar as estatísticas de uma equipa. A Figura 18 exemplifica o uso desta aplicação.

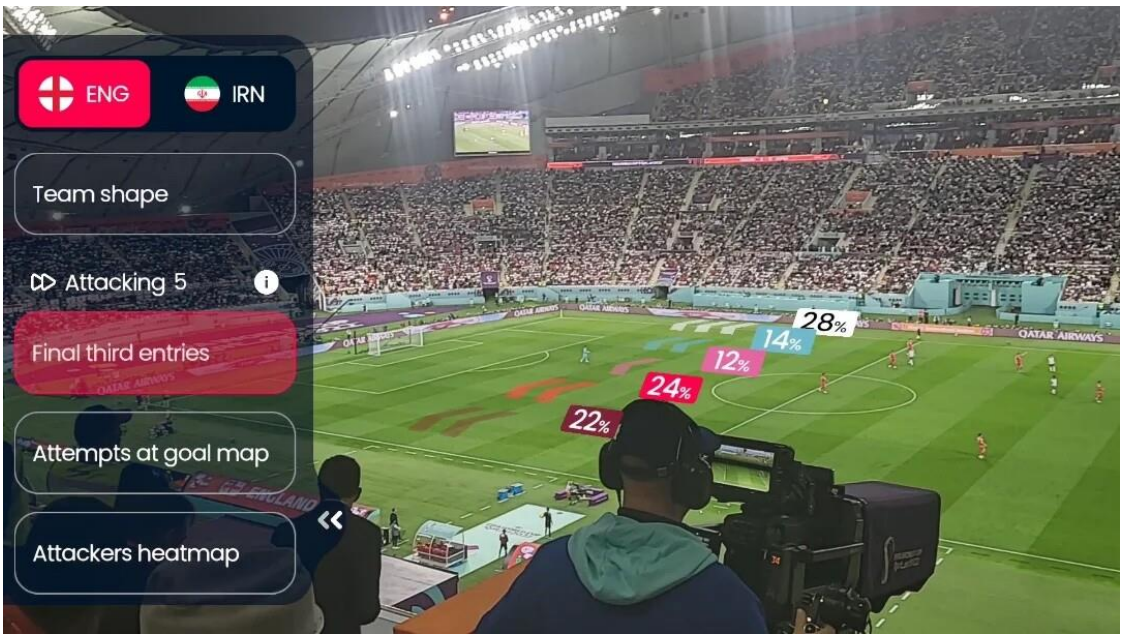

Figura 18- Realidade Virtual da app FIFA+ (Fonte: [36]).

5.2 Vantagem da tecnologia para os clubes

Este tópico tem o objetivo de nos fazer perceber como os clubes conseguem obter vantagens para a sua equipa usando a tecnologia.

Primeiro, temos de perceber que um clube que quer atingir o mais alto patamar e desempenho, tem de investir bastante, não só em infraestruturas, *staff*, jogadores, mas sobretudo em tecnologias, para assim potencializar as suas chances de obter sucesso desportivo e consequentemente financeiro.

Basta compararmos "os três grandes" de Portugal com uma equipa das divisões inferiores, e logo vemos a vantagem que os grandes têm, não só em infraestruturas, mas também pelo seu poderio financeiro que lhes possibilita adquirir tecnologias de ponta, o que no caso de um clube das divisões mais inferiores não acontece porque requer um grande investimento.

Atualmente para os clubes, a tecnologia tem um papel crucial em vários aspetos (vamos desenvolver cada um dos aspetos mais à frente), que são:

- Desempenho dos jogadores em campo.
- Descoberta de talento.
- Análise tática de adversários.
- Melhorar a tomada de decisão.
- Prevenção e recuperação de lesões.

5.2.1 Desempenho dos jogadores em campo

O desempenho dos jogadores em campo é um dos principais aspetos em que os clubes devem se preocupar para terem chances de sucesso no final de cada época, pois sem um bom desempenho dos seus jogadores dentro do campo, as chances do clube ter sucesso são mais baixas. Assim, os clubes de futebol utilizam os seus recursos tecnológicos principalmente para analisar e monitorizar os seus ativos. Há várias análises que são feitas pelos analistas de dados do clube para promover *insights* à equipa técnica de maneira a provar quais são os melhores jogadores para serem os principais protagonistas e quais irão efetuar o melhor papel de acordo com a posição ou consoante o adversário, pois cada jogo é diferente e surgem circunstâncias diferentes, e os clubes têm de estar prevenidos para isso.

Para a análise de desempenho dos jogadores pode-se analisar tanto dados físicos, como psicológicos e técnicos.

- Dados Físicos: neste tipo de dados incluem-se: distância percorrida, velocidade máxima, número de acelerações e desacelerações assim como as suas médias, força, equilíbrio e impulsão.
- Dados Psicológicos: concentração, decisões, agressividade, liderança, índice de trabalho e estresse.
- Dados Técnicos: são aquelas estatísticas dos jogadores durante um jogo de futebol ou um treino como: número de passes efetuados, assim como a sua eficácia, número de remates efetuados, assim como a sua eficácia, número de golos e assistências por 90 minutos, número de cruzamentos, livres, cabeceamentos, etc.

A análise que pode propiciar melhoria de desempenho dos jogadores também pode ser utilizada para criar planos de jogos para contextos específicos dentro dos jogos. Por exemplo, vamos criar um cenário que a equipa A que joga em um estilo de contra ataque (que é jogar à espera do erro do adversário) vai jogar contra a equipa B que é teoricamente mais fraca e quer entrar de maneira diferente a pressionar a outra equipa. Isto vai exigir mais esforço físico dos seus jogadores e isto consequentemente vai implicar um certo trabalho específico durante aquela semana de treino para preparar o jogo desta maneira. Assim, exige que se melhore o desempenho dos jogadores em certos aspetos para o molde da ideia que o treinador quer propor para aquele jogo, já que o futebol atual está muito dinâmico nas maneiras de abordar os vários jogos.

Também existem muitos clubes de futebol que analisam os seus jogos de futebol de uma perspetiva diferente. Analisam os jogadores que estão a tomar tanto decisões acertadas ou decisões erradas, dando assim *feedback* à equipa técnica para manter ou efetuar uma substituição de um certo jogador.

5.2.1.1 Descoberta de talento

Entrando agora em um caso real de um clube do futebol, o Sport Lisboa e Benfica tem um projeto chamado "Benfica LAB", onde o clube tem alocada uma equipa de fisiologistas mais conhecidos como s*port scientists*, que fazem o acompanhamento dos escalões sub-14 até à equipa B, preparando os jogadores jovens para subirem à equipa A. Esta equipa intervém em áreas de desempenho como força e condicionamento, onde vai melhorar a condição física dos jogadores de maneira a melhorar os seus desempenhos e reduzindo as chances de lesão e avalia aspetos técnicos dando mais *insights* à estrutura sobre cada jogador [25]. Outra grande área onde tem intervenção é de monitorização, onde consegue-se controlar as cargas treino internas e externas, monitorização do sono e avaliação da disposição do atleta para treinar [25].

Dentro do Benfica LAB existe uma tecnologia chamada "360S simulador " que é um simulador baseado em situações reais de um jogo de futebol, onde o analista pode criar o maior número possíveis de exercícios para simular situações de jogo, fazendo assim que o jogador experiencie este tipo de situações para melhorar o seu desempenho dentro das 4 linhas em um jogo real. São trabalhos específicos por atleta e por posição, ou seja, dependendo das circunstâncias em que se encontra um atleta são atribuídos a ele vários tipos de exercícios para uma determinada posição. Neste simulador, o jogador fica no centro rodeado por 4 paredes, onde nestas paredes vão passar bonecos a simular jogadores da sua equipa ou da equipa adversária, o jogador consegue identificar ambas as equipas pelas cores dos bonecos. Ainda nesta sala existem vários canhões que vão disparar a bola diretamente ou com efeito, de forma a simular uma situação real, com velocidades entre os 20km/h e os 80km/h, para o atleta receber e a seguir completar o exercício, seja ele de finalização ou um passe a um colega de equipa. Resumindo, trata-se de uma realidade virtual que trata situações reais de treino ou de jogo. Os dados registados durante esta simulação vão diretamente para a base de dados do clube que depois vai ser partilhado para o Benfica Lab e por todos os departamentos técnicos, de forma a se ter um historial do desempenho do atleta e da sua evolução técnica ao longo do tempo. A Figura 19 mostra um exemplo de utilização deste sistema.

Figura 19-360S Simulator do Benfica Lab (Fonte: [34]).

Para descobrir talento, a análise de dados tem um papel fundamental para possibilitar que os grandes clubes recrutem os melhores talentos do mundo. Os melhores clubes do mundo têm olheiros espalhados por todos os continentes e analisam detalhadamente todos estes talentos, identificando rapidamente aqueles com maior potencial segundo os dados colhidos. Isto faz com que as equipas melhorem a sua tomada de decisão e minimizem o risco de maus investimentos para os seus plantéis.

5.2.2 Análise de adversários

Os clubes também tiram partido da tecnologia quando o assunto é analisar os seus próximos adversários. Os clubes possuem tecnologia suficiente para coletar dados sobre a forma recente do seu adversário e sua maneira de jogar, de forma a avaliar o seu adversário e estudar a melhor forma de contrariar o seu jogo. Estes softwares específicos geram tabelas e gráficos que são muito importantes para a análise.

Além destes softwares, as equipas técnicas utilizam técnicas simples como a visualização de imagens e vídeos para perceber a filosofia de jogo dos seus próximos adversários. Com este conhecimento podem explorar as vulnerabilidades dos seus adversários, conhecerem também os pontos fortes do adversário como bolas paradas e cobranças de cantos, para tentar minimizar os "danos" que a equipa adversária possa fazer, neutralizando-os assim, de maneira a ganhar vantagem competitiva, se a informação for aproveitada corretamente.

A Figura 20 mostra uma análise a um adversário. Ela foi retirada de um jogo de computador, mas que é um simulador realístico e que se assemelha ao mundo real. Observamos

que foi traçado um gráfico em rede muito utilizado para este tipo de análise dentro do mundo de futebol, permitindo realizar que o adversário dentro deste exemplo possui estatísticas acima da média dos clubes da sua liga, estando acima da média em todas as variáveis deste gráfico, deste o "rácio de desarmes ganhos (%)", até aos" Remates por jogo". Neste tipo de análise conseguimos perceber visualmente o que as equipas conseguem obter de *insights* para o estudo do seu adversário e conhecer os seus pontos fortes e os seus pontos fracos.

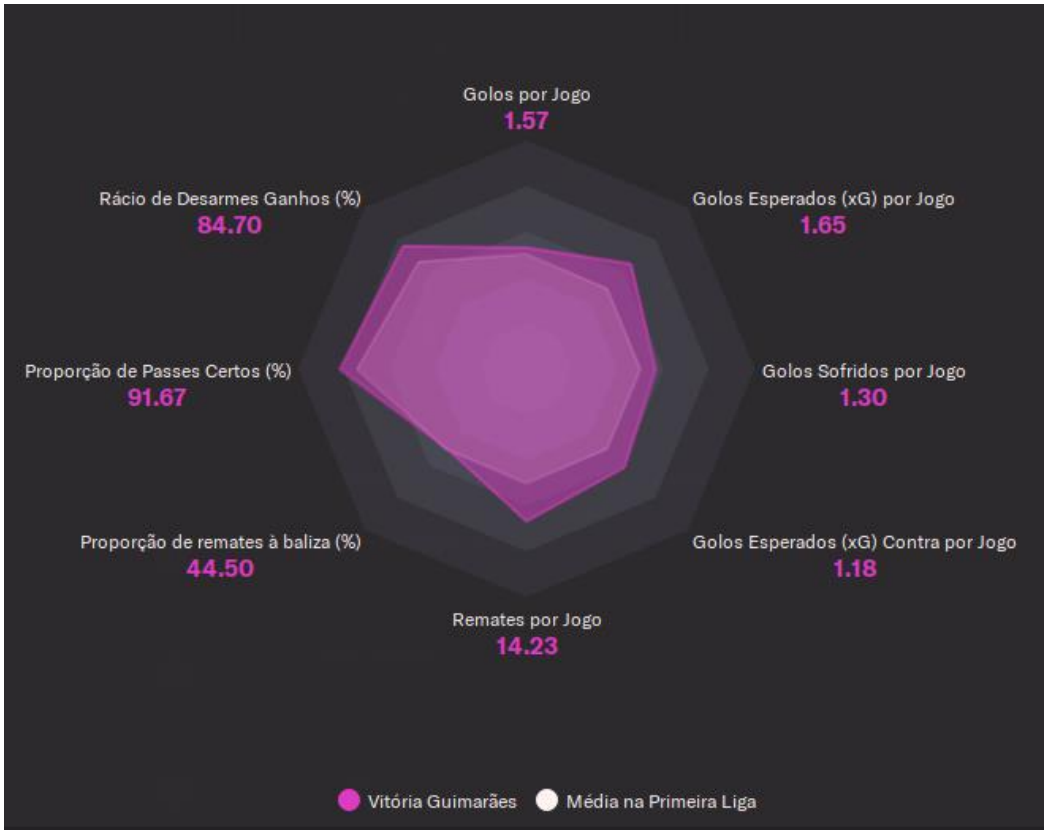

Figura 20-Exemplo de análise de um adversário (Fonte: Football Manager 22)

5.2.3 Melhor tomada de decisão

Os clubes de futebol também podem tirar partido das suas tecnologias para tomarem a melhor decisão em benefício próprio. Por exemplo, na Figura 21, vemos uma comparação entre dois jogadores na qual a equipa técnica pode tomar a decisão de escolher um dos jogadores pelas características que mais se assemelha com o que querem ou apenas optar pelo melhor jogador em geral segundo o gráfico em rede ou ainda o melhor jogador pelas estatísticas. Esta é uma das formas onde que a tecnologia pode beneficiar a tomada de decisão dos clubes de futebol.

Figura 21-Exemplo de comparação de jogadores (Fonte: Football Manager 22)

Além de perceber os pontos fortes e fracos dos elencos, a tecnologia que é utilizada atualmente no futebol pode também ajudar os clubes a preparar melhor os seus plantéis no início de cada época ao melhorar as políticas de contratação e transferências de jogadores ou *staff* técnico. No caso dos jogadores pode se observar se o histórico de lesões é favorável ou não, pois se for contratado um jogador com um histórico complicado de lesões, será uma má decisão. A análise de adversários, como vimos anteriormente, também é uma forma de melhorar a tomada de decisão, já que a equipa melhora as suas táticas e decisões de forma a tentar aumentar as suas chances de vencer os jogos.

Outra forma de melhorar as tomadas de decisão é quanto às finanças do clube e a forma como são geridas, onde pode-se explorar formas de crescer as receitas. Uma abordagem muito em voga é que os clubes de futebol também devem ser geridos como empresas normais, maximizando os lucros e minimizando os gastos, explorando abordagens de *merchandising*, bilhetes anuais e subscrições de sócios, para assim o clube obter lucro rapidamente, não esquecendo dos patrocínios que são uma das principais fontes de receitas e pilares de todos os clubes de futebol atualmente.

5.2.4 Prevenção e recuperação de lesões

Um dos papéis cruciais da tecnologia no futebol atualmente é na prevenção de lesões assim como também a recuperação dos jogadores.

Os analistas de dados e os cientistas de dados dos clubes de futebol estão constantemente a receber dados durante os treinos e jogos de futebol com informações privilegiadas do estado físico dos seus jogadores. De acordo com estes dados, conseguem perceber se os seus atletas estão em risco de lesão ou não. Se estiverem em risco de lesão, o departamento médico juntamente com o departamento de análise informa a equipa técnica para não colocar o jogador em um próximo jogo, ou diminuir o tempo de jogo e também adaptar o plano de treino e carga de trabalho, para assim evitar que o jogador desfalque a equipa por incapacidade física.

Voltando ao projeto do Sport Lisboa e Benfica, o Benfica LAB trabalha com os seus atletas de maneira a reduzir a incidência de lesões, fazendo uma abordagem adaptada ao perfil dos seus atletas com os respetivos históricos de lesões. Para os jogadores que se lesionaram, existe uma área chamada "*Return to play*", onde é otimizada a condição física do atleta antes de se reintegrar, efetuada uma avaliação específica da patologia e elaborado um plano de prevenção com avaliações do atleta periodicamente [25].

Dando ainda mais ênfase ao mundo real, durante a quarentena, os campeonatos de futebol foram obrigados a fazerem uma longa pausa, devido à situação pandémica a que o mundo esteve sujeito. Os clubes da *Premier League* (clubes de futebol da primeira divisão inglesa) enviaram *wearables* para monitorizarem os níveis de condicionamento físico dos seus atletas e o seu padrão de sono, que como vimos é fundamental para a recuperação muscular de um jogador [26]. Esta tecnologia, assim como os coletes que os jogadores usam, surgiu pelo desenvolvimento no campo militar, para obtenção de informação e monitorização do bem-estar das forças militares pela estação central [27]. Estes *wearables* são uma tecnologia que funcionam como sensores que servem para coletar dados de informação biológica como: frequência cardíaca, saturação de oxigénio no sangue, tensão arterial, lactatos do sangue, temperatura corporal [28]. Depois destes dados serem recolhidos a informação vai para uma única *cloud* onde os dados vão ser analisados. Ao falarmos de *wearables* podemos incluir tanto os coletes utilizados em treinos e jogos que medem os dados físicos do atleta, como dos relógios inteligentes para medir os dados biológicos.

Na Figura 22, podemos observar através de uma simulação como pode ser observada a análise do risco dos jogadores contraírem lesões. Pode se verificar a carga de jogo na primeira coluna, onde refere que o jogador efetuou um número de jogos dentro de um número determinado de dias que pode ser considerada ligeira, média e forte. A seguir, na segunda coluna podemos observar a capacidade de treino que pode indicar que o jogador efetuou treino com a equipa e/ou treino individual, pode ser considerada ligeira, média ou forte dependentemente da sua capacidade. Na terceira coluna, vemos a suscetibilidade a lesões que

os jogadores têm. Podemos observar que há jogadores com suscetibilidade baixa, que quase nunca se lesionam, depois temos abaixo da média que também não têm lesões frequentes, mas depois podemos ter jogadores com uma suscetibilidade média que pode variar muito dependendo do momento que esse jogador está a passar, tanto pode contrair lesões como não. E depois temos os jogadores com mais suscetibilidade a contrair lesões, os que são acima da média e os que têm suscetibilidade a lesões muito alta. Na coluna a seguir temos os níveis de fadiga consoante o jogo anterior e no final de tudo, o algoritmo determina se o jogador está em risco de lesão ou não. Como se pode ver, há jogadores com risco elevado de lesão e outros com risco muito elevado. Estes últimos demandam maior atenção da equipa técnica para reduzir as suas cargas de treino ou diminuir o tempo de jogo. Caso a situação seja de risco muito elevado, o mais seguro é fazer o jogador descansar e não atuar no próximo jogo. Por fim, temos jogadores com o risco considerável, o que podemos considerar normal.

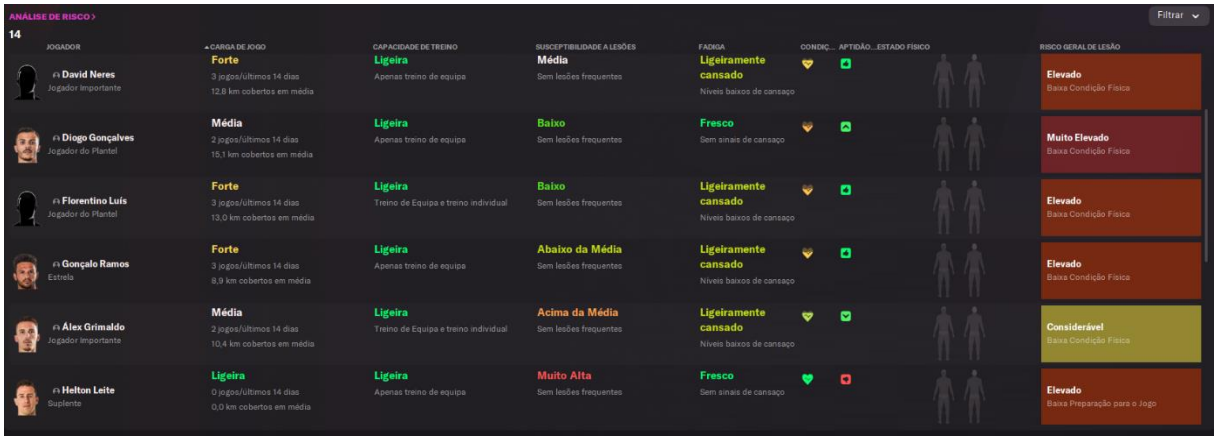

Figura 22-Análise Risco de lesão (Fonte: Football Manager 22)

6 Compreender as lesões atualmente

Tem sido frequente no futebol moderno o surgimento de muitas lesões, tanto sem contato como com contato, mas a maioria das lesões atualmente que têm surgido são lesões musculares, aquelas que são causadas sem contato algum com outro jogador.

O futebol moderno é caracterizado por ser um futebol físico, ou seja, de alta intensidade. O atleta tem de estar bem preparado fisicamente para fazer várias acelerações, desacelerações, mudanças de direções repentinas, ter capacidade de percorrer vários quilómetros durante o jogo todo, ser veloz, ter capacidade de choque com o adversário, pois o futebol é um desporto com muito contato físico.

6.1 Fatores que influenciam as lesões

6.1.1 Número de jogos

Existem inúmeros fatores que podem explicar estas lesões. Um dos fatores associados está relacionado com o calendário de jogos apertados nas competições dos clubes. Com um maior número de jogos e mais competições, incluindo os treinos exigentes que preparam os jogadores para esses jogos, os atletas são levados muitas vezes ao extremo. Muitas vezes, além de estarem a realizar vários jogos pelo clube de futebol, alguns jogadores ainda efetuam jogos pela seleção do seu respetivo país quando são frequentemente convocados. Alguns atletas, se participarem em todas as competições de um respetivo ano e forem até às fases finais entre seleção nacional e clube, chegam a efetuar mais de 60 jogos por ano, o que requer bastante esforço físico para um atleta e obviamente este logo fica em risco de lesão por ter uma carga de trabalho muito alta. Citamos um exemplo da vida real: na temporada 2020/21 o jogador do Futbol Club Barcelona chamado Pedri González realizou uns inacreditáveis 71 jogos entre Setembro de 2020 e Agosto de 2021, que é uma carga consideravelmente exagerada. Entre estes 71 jogos o atleta realizou 52 pelo seu clube, 6 jogos pela seleção nacional espanhola no Euro 2020, sendo 3 destes jogos com prolongamento, ou seja, houve 3 jogos em que jogou durante 120 minuto (falhando apenas 1 minuto de toda a competição), 6 jogos nos jogos olímpicos de Tóquio de 2020 pela seleção olímpica espanhola e os restantes foram jogos de qualificação pela seleção espanhola. Após este período, o jogador foi obrigado a tirar duas semanas de férias pelo Futbol Club Barcelona e a federação espanhola de futebol depois de já ter efetuado mais 2 jogos da época seguinte, o que totalizou 73 jogos sem o jogador tirar férias. Porém, logo após o primeiro jogo depois das férias que o jogador usufruiu contra o Fußball-Club Bayern de Munique no dia 14 de Setembro de 2021, em uma partida a contar para a fase de grupos da liga dos campeões, o jogador sofreu uma recaída muscular. Após supostamente ter passado essa recaída muscular o jogador realizou um jogo frente ao Sport Lisboa e Benfica, no dia 29 de Setembro de 2021, um jogo a contar para a fase de grupos da liga dos campões da época 2021/22, e sofreu uma lesão muscular no quadríceps da coxa direita. No início, o departamento médico do F.C. Barcelona avaliou a paragem do atleta por mais de um mês, mas após várias avaliações, a lesão era mais grave do que se pensava. E o atleta teve uma longa paragem e só retornou em Janeiro de 2022.

Isto é um bom exemplo para contextualizar o cenário em que os jogadores estão envolvidos, pois aqui conseguimos perceber a carga de trabalho pesada a que este jogador esteve sujeito, sendo ele um jogador que desempenha o papel de médio centro, ou seja, é um dos jogadores que mais corre em campo, esteve com um risco de lesão muito alto durante muito tempo. Podemos observar na Figura 23 os dados da temporada 20/21.

Jogadores	Idade	Nac.	Clube 20/21	Jogos no clube	Internacionalizações	Minutos	Total
Pedri Médio Centro	17		Barcelona LaLiga	52	19	4793'	71

Figura 23- Nº de jogos efetuados por Pedri em 21/22 (Fonte: [35])

O número de jogos e as demasiadas competições como vimos neste exemplo, é um dos fatores que contribuem para as inúmeras lesões dos atletas de futebol. O exemplo do Pedri é de um jovem de 17 anos que conseguiu fazer 71 jogos, mas os jogadores mais velhos e com históricos de lesões musculares têm de ser geridos, porque muitos deles não iriam conseguir fazer esta quantidade de jogos, por conta da questão da idade e do histórico de lesões. Ou iriam se lesionar em um determinado tempo no decorrer da época ou a sua forma iria baixar bastante e não estariam a render tão bem quanto poderiam. É por isso que as equipas técnicas juntamente com o departamento médico e o de análise têm de traçar estratégias e planos específicos para cada jogador, consoante o perfil de cada atleta, para evitar riscos de lesões por fadiga.

6.1.2 Alta-intensidade

Um fator que também contribui para o aumento das lesões musculares dos jogadores é a alta intensidade, referida anteriormente quando falamos sobre o futebol moderno. As mudanças de direção repentinas, as constantes acelerações, desacelerações, o choque entre jogadores, um *sprint* muito forte que os atletas têm de fazer numa transição rápida, tudo isto faz com que o jogo em si seja muito rápido. Isto também vai depender de fatores como a maneira como uma determinada equipa joga, qual seria a sua tática. A forma como uma equipa joga influencia muito na intensidade que aplica no jogo. Por exemplo, uma equipa que está constantemente a pressionar o adversário, precisa de estar sempre em movimento e ter capacidade de choque com o adversário para tentar roubar a bola. Já uma equipa que joga no contra-ataque opta por esperar o adversário no seu campo de defesa e quando recupera a bola efetua *sprints* em uma jogada rápida e objetiva para tentar o golo. O que além de requerer um bom preparo físico, também implica que o jogador tem de estar concentrado para conciliar todos

esses fatores à intensidade e os utilizar de maneira inteligente. A Figura 24 sintetiza o conjunto de fatores associados à intensidade dos confrontos.

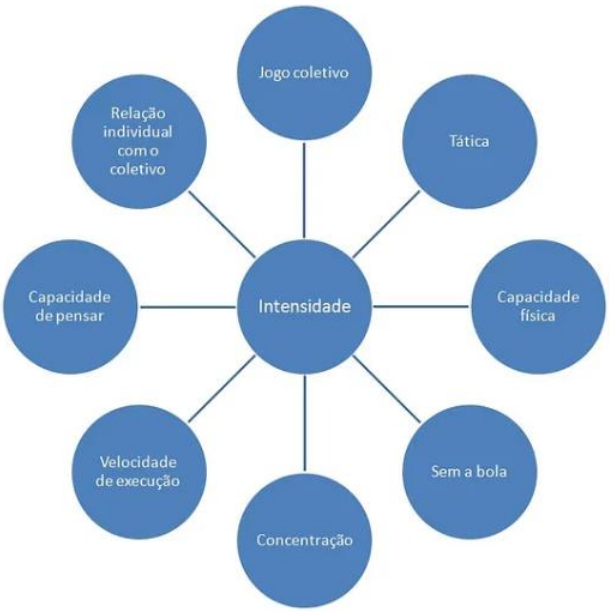

Figura 24- Fatores associados à intensidade(Fonte: [37]).

6.1.3 Relvados dos estádios

Outro fator que pode influenciar no aumento das lesões sem contato dos jogadores, são os relvados de alguns estádios, pois nem todos os clubes têm as condições dos clubes de primeira liga e por vezes o seu relvado acaba por ter impacto na questão física dos jogadores e também o seu desempenho dentro dele. Muita vezes, os jogadores nestas situações acabam por ter lesões de entorse, tanto no joelho, quanto no tornozelo. Normalmente estas lesões são causadas por mau apoio do atleta na superfície de jogo o que causa estas lesões, pois existem muitos relvados em más condições e alguns atletas também têm vindo a lesionar se bastante em relvados artificiais. Para prevenir isto, os atletas devem ter um plano específico de treino de equilíbrio e propriocepção para fortalecer as articulações e diminuir o risco de lesão nestas zonas, assim como utilizar pitons das chuteiras próprias para um determinado relvado onde se vai jogar.

6.1.4 Falta de preparo físico

Um fator que se pode apontar é a falta de preparo por parte de um atleta durante os treinos ou mesmo ausência de ritmo de jogo, pois por não estar preparado muitas vezes ao ritmo a que está sujeito durante o jogo, o atleta lesiona-se. O que também pode ser indicado por má

preparação física durante os treinos ou mesmo excesso de carga durante os treinos. Nestas situações há que seguir um determinado plano sugerido pelo *staff* técnico de maneira a gerir as cargas de trabalho e descanso do atleta. O que acontece também muita das vezes é a falta de preparo físico apropriado, principalmente para jovens jogadores, durante os jogos ou treinos com choques físicos e esses jogadores neste aspeto são mais suscetíveis a sofrerem lesões.

6.2 Prevenir este tipo de lesões

Para prevenir este tipo de lesões, o departamento médico e de análise juntamente com o *staff* técnico têm de trabalhar em conjunto e obter um plano específico para os jogadores todos, ou seja, um treino coletivo que esteja dentro dos intervalos aceitáveis do ACWR, nem muito baixo para não estarem sem ritmo, o que aumenta o risco de lesão, nem muito alto para não estar com demasiada carga de trabalho e também com risco de lesão.

6.2.1 ACWR (*Acute:Chronic Workload Ratio*) Carga de Trabalho Crónica Aguda

O ACWR é um indicador que nos permite monitorizar a carga de trabalho de um atleta no passado e indica quanto pode utilizar em um futuro próximo, de maneira a gerir as suas cargas de trabalho. O atleta pode ser analisado por este ACWR que está dividido em duas partes:

- Taxa de Carga de Trabalho Aguda
- Taxa de Carga de Trabalho Crónica

6.2.1.1 Taxa de Carga de Trabalho Aguda

A taxa de carga de trabalho aguda é o trabalho realizado por um atleta durante uma semana, o que vai incluir informações tanto de treino quanto de jogo e vai ser representado como o aspeto de "fadiga" para o ACWR [29]. Por exemplo, "se quisermos calcular a carga de trabalho basta multiplicar sRPE (*sessions Rating of Perceived Exertion*) pelo tempo de treinamento. Vamos imaginar que o sRPE de um atleta foi de 6 e treinou por 100 minutos, a carga de trabalho do atleta será de 600 AU (*Arbitrary Units*)" [29]. Se o atleta efetua um treino bi-diário, será calculado o valor de cada sessão e faz-se o somatório.

6.2.1.2 Taxa de Carga de Trabalho Crónica

A taxa de carga de trabalho crónica vai ser a média das 4 semanas seguintes ao trabalho realizado na taxa de carga de trabalho aguda. Representa a parte 'fitness'.

As análises destes dados vão depender muito dos modelos a serem utilizados para a interpretação dos dados. Existem dois modelos principais para o cálculo do ACWR:

- O modelo de média móvel (RA – *Rolling Average*).
- O modelo de média móvel exponencialmente ponderada (EWMA – *Exponentially Weighted Moving Average*).

6.2.1.3 Modelo de média móvel (RA)

Este modelo sugere que se trabalhe de maneira linear, ou seja, sugere que as duas cargas de trabalho sejam iguais durante um determinado período de tempo. Entretanto, este modelo não é tão preciso. Para melhorar a precisão existe o modelo de média móvel exponencialmente ponderada.

6.2.1.4 Modelo de média móvel exponencialmente ponderada (EWMA)

Considera com mais relevância a carga de trabalho que o atleta realizou mais recentemente e foi feito para levar em consideração a condição física do atleta, a imprevisibilidade de lesões e da carga de trabalho. Também representa melhor a forma como as cargas de trabalho são acumuladas e variadas.

A Figura 25 exemplifica uma coleta de dados relacionando RA e EWMA.

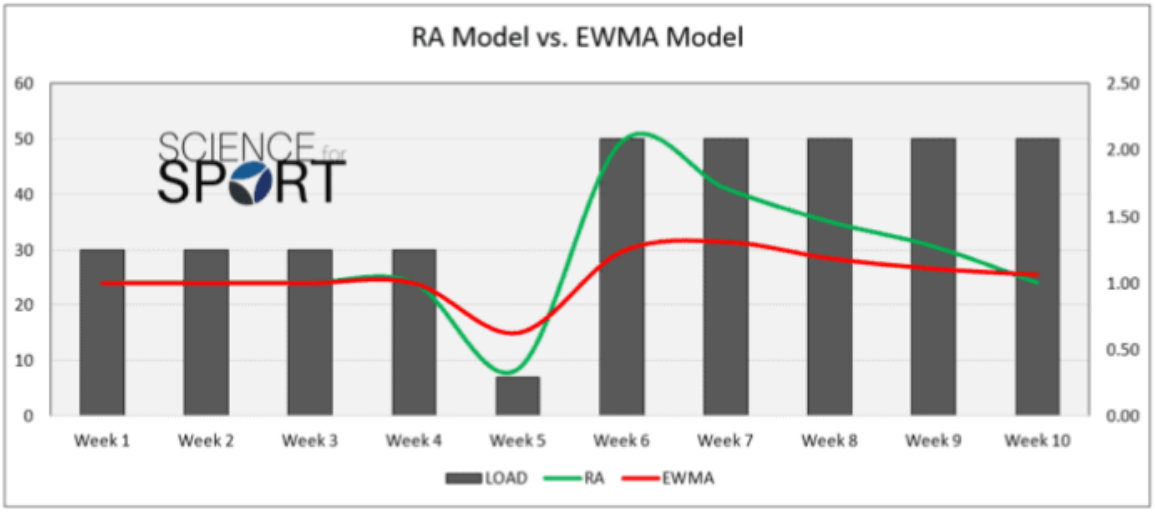

Figura 25- Relação entre os modelos RA e EWMA (Fonte: [29]).

6.2.1.5 Prevenção de Lesões com o ACWR

Para conseguirmos aprofundar ainda mais o conhecimento dentro do ACWR, precisamos de saber quais são as métricas utilizadas dentro dos dois tipos de carga que foi visto

na revisão da literatura: a carga externa que é observada através do GPS e extrai dados físicos como velocidade máxima do jogador, acelerações, desacelerações, etc., e a carga interna que extrai dados biológicos como a taxa de esforço percebido, a frequência cardíaca, o lactato sanguíneo e níveis de oxigénio.

Para calcularmos o ACWR precisamos dividir a taxa de carga de trabalho aguda pela taxa de carga de trabalho crónica, por exemplo, se a taxa de carga de trabalho aguda for 1550 AU e a taxa de carga de trabalho crónica for de 1603 AU o ACWR será de 0.97. O que isto significa? Semelhante ao que já tínhamos visto na revisão da literatura, se a carga aguda for baixa (o que significa que a fadiga do atleta está baixa) e a carga crónica for alta (o que influencia diretamente o estado "fitness" do atleta), o ACWR será <1.00 e se for o inverso será >1.00 [29]. Bem, isto significa que com 0.97 o atleta está dentro do intervalo esperado para um jogador não estar em risco de contrair uma lesão por não ter demasiada carga de trabalho, mas também não está com pouca carga de trabalho que iria colocar o atleta em sub-rendimento o que também estaria em risco de contrair uma lesão. Vamos agora observar as análises aos resultados do ACWR e o que significa cada intervalo:

- < 0.80 – O atleta está fora de ritmo e está em sub-rendimento, o que aumenta as suas chances de contrair uma lesão muscular durante um treino ou durante um jogo.
- 0.80 – 1.30 – Carga de trabalho boa e risco de lesão muscular baixa. Melhor intervalo para um atleta estar durante uma época.
- > 1.50 – Zona mais perigosa para um atleta se lesionar, é aqui que se tem a maior hipótese de contrair uma lesão muscular.

Esta análise do AWCR é fundamental para os departamentos médicos e de análise dos clubes do futebol, pois não só monitoriza os atletas diariamente como também permite fazer um planeamento durante certos períodos para os atletas, para protegê-los de possíveis lesões musculares.

6.2.2 Aquecimento

Outra das medidas a seguir para evitar o risco de lesão muscular é efetuar um grupo de exercícios de aquecimento de maneira bem realizada antes de qualquer treino e jogo, exercícios de alongamento e mobilidade isto faz com que os músculos não fiquem rígidos, reduzindo assim a resistência dos tecidos musculares e melhorando também a flexibilidade das articulações

todas do corpo. Isto é um sinal de alerta para o corpo preparar-se para um período de atividade física esforçada. O aquecimento serve também para o exercício físico não apanhar os músculos desprevenidos de maneira a evitar as lesões.

6.2.3 Fortalecimento muscular

Mais uma técnica para evitar lesões musculares em atletas de futebol é o fortalecimento muscular. Há casos em que as lesões musculares acontecem devido a desequilíbrio muscular, falta de força e falta de flexibilidade, e isto acontece com maior frequência em jogadores mais jovens. Para resolver esse problema, o departamento médico irá programar com o *staff* técnico um programa específico de treino de força e treino de resistência muscular, para mais tarde evitar lesões mais graves, lesões de cartilagens que podem afetar o menisco e os ligamentos do tornozelo e do joelho.

6.2.4 Abordagens com ML

Nos últimos anos esta tem sido a grande revolução nos grandes clubes de futebol, estamos a referir os grandes porque são os que possuem maior poderio financeiro e este tipo de tecnologias complexas exige um grande investimento principalmente para a coleta de dados. Mas de facto traz um grande retorno para os clubes de futebol. Os clubes mais pequenas na maioria das vezes não possuem orçamento disponível para investir neste tipo de tecnologias.

O objetivo dos clubes com a tecnologia de ML é de buscar um modelo que consegue prever se um jogador pode estar em risco de lesão ou não, mediante as variáveis que são analisadas, já que os clubes estão praticamente todos os dias a recolher dados dos seus atletas, tanto nos treinos e jogos (dados como distâncias percorridas, velocidades, quanto tempo o jogador esteve a caminhar, quando tempo esteve a correr a um intervalo de velocidades, quanto tempo esteve a “sprintar”), até aos dados que coletam quando os atletas estão fora das 4 linhas (qualidade do sono, duração do sono, stress do jogador). Estes últimos tipos de dados ajudam a perceber quais são os jogadores que melhor recuperam depois de uma sessão de jogo ou uma sessão de treino. Isto está diretamente relacionado também com a melhoria na tomada de decisões por parte dos clubes de futebol e também pode trazer vantagem competitiva aos demais desportos, dependendo da tecnologia de clube para clube. Pode haver clubes com tecnologias mais refinadas e que processem mais e melhor informação para as suas bases de dados e assim os diversos departamentos podem se beneficiar dessa informação.

Voltando a dar exemplo de um caso real, o laboratório de ciência de dados do Sport Lisboa e Benfica, com o suporte da tecnologia da Microsoft, iniciou o primeiro grande projeto de Transformação Digital no desporto, para a Microsoft Portugal [30].

Na altura da implementação do projeto, Bruno Mendes, o responsável pelo laboratório do Sport Lisboa e Benfica, em declarações à revista Wired, disse e passo a citar "*Através do uso de tecnologias de ML e análise preditiva podemos aprender que informação nos vai levar ao sucesso. Os jogadores podem usar esta aprendizagem para melhorar a sua performance e evoluir de forma consistente*" [30].

O na altura CIO (*Chief Information Officer*) do Sport Lisboa e Benfica, João Copeto, afirmou que antes da implementação deste projeto o clube utilizava *datacenters* e servidores próprios, mas o clube optou pelo Microsoft Azure, por ser poderoso [30].

Esta tecnologia permite aos cientistas de dados recolher os dados para um único local, e após isso as equipas utilizam ferramentas como Microsoft Azure e o Power BI para uma análise mais detalhada e para tomar decisões sobre os seus atletas.

7 Prevenção de lesões com ML

A presente dissertação tem o objetivo de perceber como é que os clubes de futebol conseguem antecipar possíveis lesões musculares nos seus jogadores e para resolver este problema, com as métricas identificadas iremos construir neste ponto um modelo de ML com dados históricos para no futuro os clubes de futebol possam ter uma noção precisa sobre quais dos seus jogadores estão em risco de lesão.

Para a construção deste modelo, vamos utilizar uma metodologia muito conhecida no mundo da análise de dados que é o CRISP-DM (*Cross Industry Standard Process for Data Mining*).

7.1 CRISP-DM

Cross Industry Standard Process for Data Mining é um modelo de processo de ciência de dados utilizada para mineração de dados que foi desenvolvido e apresentado no final dos anos 1990.

Esta metodologia tem o objetivo de transformar um grande volume de dados em conhecimento para as empresas, de maneira a facilitar de forma mais eficaz e produtiva o trabalho da análise dos dados. Esta metodologia possui 6 fases que são:

1. **Compreender o negócio**

   Nesta fase deve se começar a compreender os objetivos do projeto, quais são os requisitos, quais são as necessidades do negócio. Nesta fase existem as seguintes tarefas:

   - **Definição dos objetivos de negócio**
   - **Avaliação detalhada da situação**
   - **Definição dos objetivos técnicos**
   - **Construção do plano de projeto**

2. **Compreender os dados**

   É a fase em que se deve recolher e identificar os dados mais importantes para o projeto, de maneira a conseguir descrever os dados quando estiverem no processo de mineração. Aqui temos as seguintes tarefas:

   - **Recolha de dados inicial**
   - **Análise descritiva**
   - **Análise Exploratória**
   - **Validação da qualidade dos dados**

3. **Preparar os dados**

   Na fase da preparação dos dados é esperado que o analista prepare os dados para depois começar-se a modelá-los para a construção do modelo. Esta fase é também conhecida como também pré-processamento dos dados. Basicamente recebe-se os dados identificados como foi feito na fase da compreensão dos dados e prepara-se para futuras análises através do processo de mineração de dados. As tarefas nesta fase são as seguintes:

   - **Seleção de variáveis**
   - **Limpeza de dados**
   - **Cálculo de variáveis derivadas**
   - **Integração de dados**
   - **Formatação de dados**

4. **Construção do modelo**

   Esta fase é a mais interessante dos projetos de ciência de dados e também a mais curta. É a fase em que construímos o modelo que vai no final avaliar e vamos fazer uma análise comparativa. As tarefas desta fase são:

   - **Seleção das técnicas de modelagem**
   - **Definição do plano de testes**
   - **Construção do modelo**
   - **Avaliação do modelo**

5. **Teste e avaliação**

   Verifica se o modelo que foi construído vai de acordo às necessidades e objetivo de negócio. Avalia também a precisão do modelo construído. Para esta fase temos as seguintes tarefas:

   - **Avaliação dos resultados**
   - **Revisão do processo**
   - **Determinação dos próximos passos**

6. **Implementação**

   "*A fase de implementação pode ser tão simples como gerar um relatório ou tão complexa como implementar um processo de mineração de dados repetível em toda a empresa*" [31]. Nesta fase as tarefas são as seguintes:

   - **Definição do plano de entrega**
   - **Definição do plano de monitorização e manutenção**
   - **Construção do relatório final**
   - **Revisão do projeto**

Para esta dissertação não vai ser necessário desenvolver a implementação pois a própria dissertação pode ser considerada como o relatório do projeto.

7.1.1 Compreender o negócio

7.1.1.1 Definição dos objetivos de negócio

Para esta dissertação será analisado um conjunto de dados de jogadores fictícios, desde já porque as métricas necessárias para a avaliação deste modelo para a finalidade requerida não

foram encontradas, pois tratando-se de dados sensíveis é provável que os clubes de futebol não disponibilizem ao público este tipo de dados.

Dito isto, nesta dissertação temos o objetivo de ter uma visão de como os clubes de futebol conseguem prever lesões musculares nos seus atletas. E para isto temos um *dataset* com 10 jogadores a serem analisados por um período de 40 dias e com estes dados históricos iremos criar um modelo e ver a sua precisão para avaliar os jogadores no futuro e prevenir lesões aos atletas. Os resultados da análise serão baseadas nos dados colhidos dos jogadores, tanto nas sessões de treino quanto nas sessões de jogo.

7.1.1.2 Avaliação detalhada da situação

Apesar de os dados serem fictícios, estamos a partir do pressuposto que estes dados foram capturados por dispositivos *wearables* durante as sessões de treino e jogo e enquanto os atletas estavam a dormir, que é como no mundo real é feita esta avaliação.

Outro detalhe mais específico, pois não foi conseguido um conjunto de dados que tivesse as métricas do ACWR, nem muitos dados biológicos como lactato do sangue, frequência cardíaca, níveis de oxigénio, é que existem métricas para medir o esforço dos atletas, como RPE (*Rating of Perceived Exertion*), que vão ajudar imenso a construir este modelo, e assim, vai ser possível fazer a previsão se um jogador pode se lesionar ou não.

7.1.1.3 Definição dos objetivos técnicos

O objetivo principal desta mineração de dados é encontrar/construir um modelo que consiga prever se um jogador está em risco de lesão, ou seja, vamos tentar perceber como os clubes conseguem fazer isso com técnicas de ML.

7.1.1.4 Construção do plano de projeto

Para esta mineração de dados vamos utilizar vários algoritmos de classificação para prever se um jogador pode estar lesionado ou não, como: Árvores de Decisão, *Random Forest*, K-NN e Regressão Logística. Vamos também utilizar o algoritmo SVM já que pode lidar com variáveis binárias em um hiperplano para distinguir dois tipos de classe.

7.1.2 Compreender os dados

7.1.2.1 Recolha de dados inicial

Durante esta fase devemos analisar as principais características destes dados, estamos a dar o foco aos dados que estamos a receber dos atletas de futebol.

O conjunto de dados que está a ser utilizado nesta dissertação foi recolhido no site da maior comunidade de ciência de dados do mundo que é o Kaggle e foi coletado neste link: https://www.kaggle.com/datasets/michaelhegedusich/soccer-performance-data. Os dados referem-se a 10 hipotéticos jogadores de futebol com as métricas recolhidas por *wearables* com o propósito de simular dados realísticos para aqueles que pretendem trabalhar com ciência de dados.

7.1.2.2 Análise descritiva

Para esta análise, os dados que existem são os seguintes:

1 Data (*Date*)
2 Nome (*Name*)
3 Posição (*Position*)
4 Tipo de Sessão (*Session_Type*)
5 Duração do sono (*Sleep_Duration*)
6 Pontuação do Sono (*Sleep_Duration*)
7 Qualidade do Sono (*Sleep_Quality*)
8 Dor (*Soreness*)
9 Stress
10 RPE
11 Distância (*Distance*)
12 Nº de acelerações (*Acceleration_Counts*)
13 Máxima Aceleração (*Max_Acceleration*)
14 Nº de desacelerações (*Deceleration_Counts*)
15 Máxima Desaceleração (*Max_Deceleration*)
16 Velocidade máxima (*Max_Speed*)
17 Lesionado? (*Injury_Illness*)
18 Tipo de Lesão (*Injury_Type*)

Para este conjunto de dados existem 3 tipos de variáveis que são:

- **Variáveis numéricas discretas:** "*Sleep_Duration*", "*Sleep_Duration*", "Distance", "*Acceleration_Counts*", "*Deceleration_Counts*".
- **Variáveis numéricas contínuas:** "*Sleep_Quality*, "*Soreness*", "*Stress*", "RPE", "*Max_Acceleration*", "*Max_Deceleration*", "*Max_Speed*".
- **Variáveis categóricas nominais:** "*Date*", "*Name*", "*Position*", "*Session_Type*", "*Injury_Illness*", "*Injury_Type*".

Variável de saída: A variável de saída que responde à pergunta se o jogador está lesionado ou não é a variável "*Injury_Illness*" na qual as saídas serão "*Yes*" ou "*No*".

7.1.2.3 Análise exploratória

Na Tabela 1 podemos verificar a análise exploratória de cada variável.

Tabela 1- Informação das variáveis (Fonte: Elaboração Própria)

Data	Os dias em que cada jogador esteve presente em uma sessão de trabalho
Nome	Nome dos jogadores
Posição	Posição que os jogadores atuam
Tipo de Sessão	Tipo de sessão em que os jogadores atuaram, que podem ser dois tipos ("Practice" ou "Game")
Duração do sono	Número de horas que os jogadores
Pontuação do Sono	Pontuação do sono dos jogadores (em uma escala de 0 a 100)
Qualidade do Sono	Qualidade do sono dos jogadores (em uma escala de 0 a 10)
Dor	Dor sentida pelos atletas (em uma escala de 0 a 10)
Stress	Stress sentido pelos atletas (em uma escala de 0 a 10)
RPE	Esforço exercido pelos jogadores (em uma escala de 0 a 10)
Distância	Distância total percorrida pelos jogadores (em metros)
Nº de acelerações	Contagem de acelerações que os jogadores efetuaram
Máxima Aceleração	Aceleração máxima efetuada por um jogador (em m/s^2)
Nº de desacelerações	Contagem de desacelerações que os jogadores efetuaram
Máxima Desaceleração	Desaceleração máxima efetuada por um jogador (em m/s^2)
Velocidade máxima	Velocidade máxima efetuada por um jogador (em km/h)
Lesionado?	Variável de saída que vai indicar se o jogador se lesionou ou não
Tipo de Lesão	Tipo de lesão sofrida pelo jogador

Os dados desta análise têm 406 observações e 19 variáveis e temos como tipos de dados *strings* (*object*), números inteiros (int64) e números reais (float64), como mostrado na Figura 26.

```
wk.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 406 entries, 0 to 405
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Data                  406 non-null    object 
 1   Nome                  406 non-null    object 
 2   Posição               406 non-null    object 
 3   Tipo_Sessao           406 non-null    object 
 4   Idade                 406 non-null    int64  
 5   Duracao_Sono          406 non-null    int64  
 6   Pontuacao_Sono        406 non-null    int64  
 7   Qualidade_Sono        406 non-null    float64
 8   Dor                   406 non-null    float64
 9   Stress                406 non-null    float64
 10  RPE                   406 non-null    float64
 11  Distancia             406 non-null    int64  
 12  Total_Aceleracoes     406 non-null    int64  
 13  Max_Aceleracao        406 non-null    float64
 14  Total_Desaceleracoes  406 non-null    int64  
 15  Max_Desaceleracao     406 non-null    float64
 16  Max_Velocidade        406 non-null    float64
 17  Lesionado?            406 non-null    int64  
 18  Tipo_de_Lesao         7 non-null      object 
dtypes: float64(7), int64(7), object(5)
memory usage: 60.4+ KB
```

Figura 26- Informação dos dados (Fonte: Elaboração Própria)

A seguir vamos fazer uma análise dos dados obtidos pelos gráficos abaixo:

O primeiro gráfico mostrado na Figura 27 é o de histograma que nos apresenta uma visão de cada variável numérica.

Figura 27-Gráfico de Histograma

O gráfico de barras (Figura 28) apenas nos dá uma visão das variáveis categóricas.

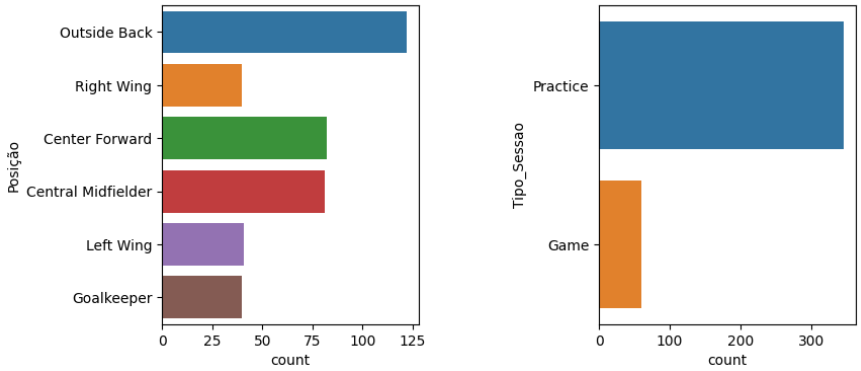

Figura 28-Gráfico de barras das variáveis categóricas (Fonte: Elaboração Própria)

Os gráficos de *boxplot* como na Figura 29 permitem ter uma melhor visão estatística dos dados, podemos observar ter uma noção clara da média, dos quartis, outliers, mínimos, máximos e desvio padrão.

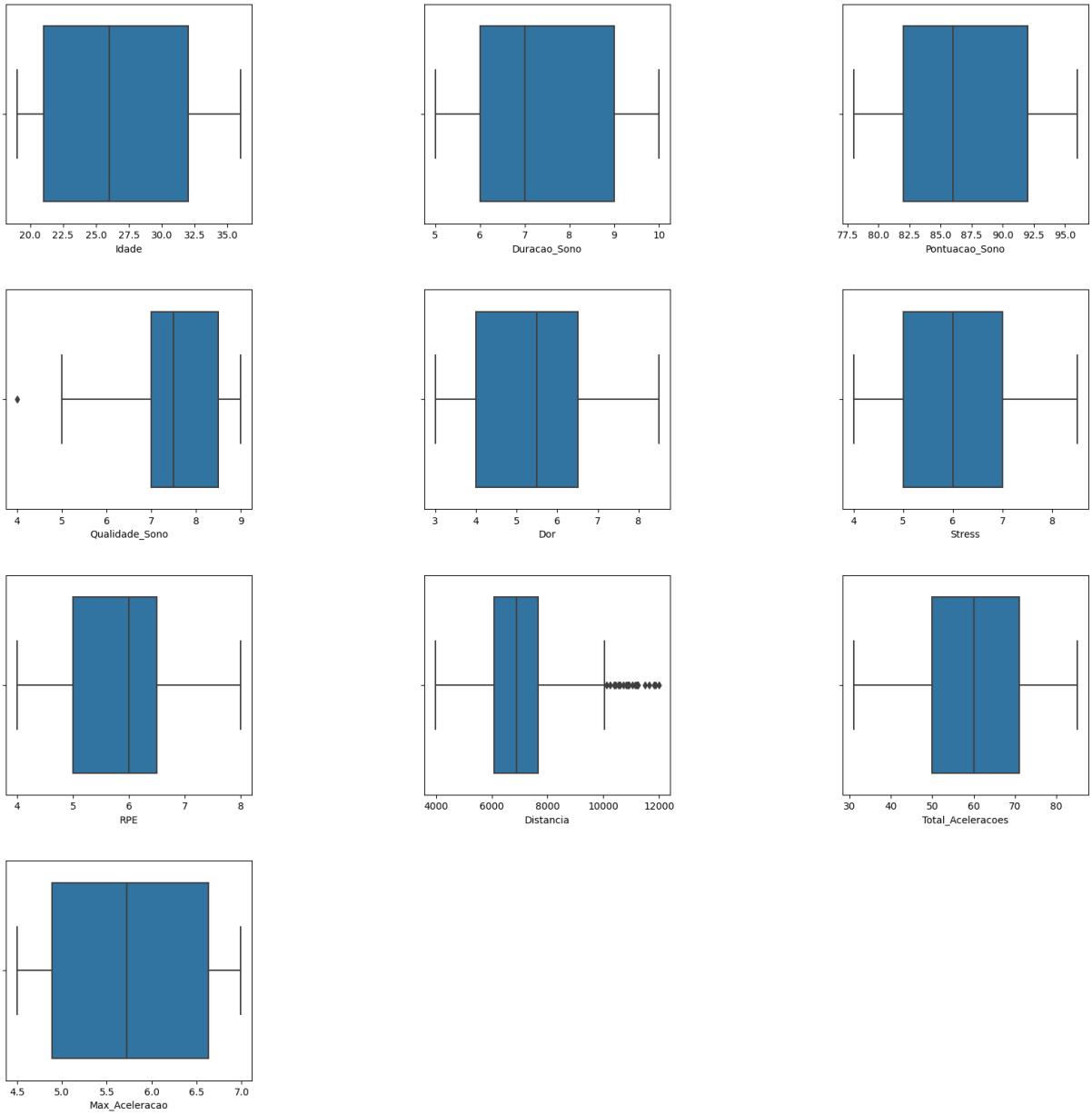

Figura 29-Gráfico de BoxPlot (Fonte: Elaboração Própria)

Já o gráfico de correlação dos dados mostrado na Figura 30 serve para observarmos visualmente e identificarmos a relação entre duas variáveis, verificando se existe alguma relação entre elas as duas. Varia de -1 a 1, quanto mais próxima de 1 podemos afirmar que existe uma correlação positiva forte e quanto se aproxima mais de -1 podemos afirmar que existe uma correlação negativa forte.

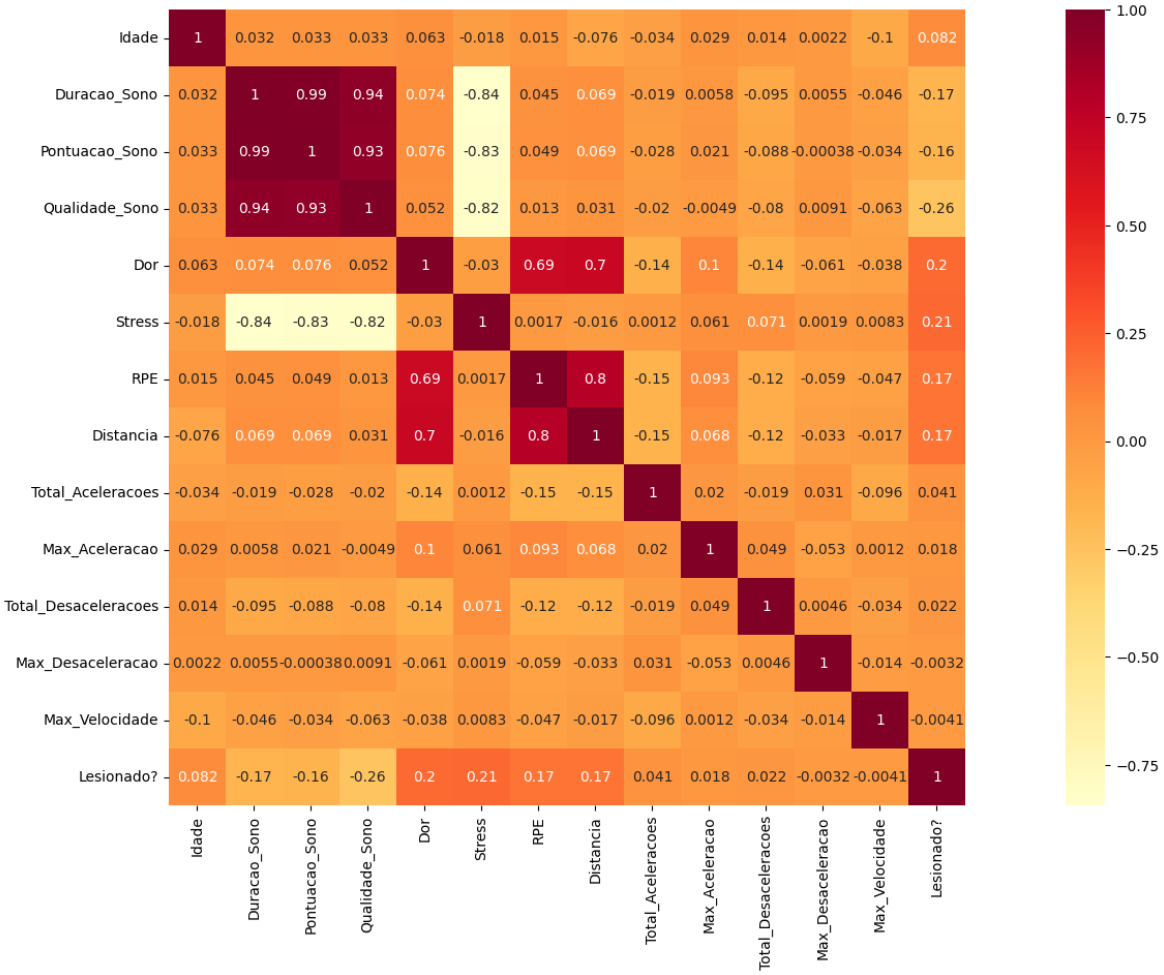

Figura 30-Gráfico de Correlação dos dados (Fonte: Elaboração Própria)

Numa análise dos dados com estes gráficos podemos dizer que:

- Existe uma forte correlação positiva entre as variáveis que correspondem ao sono.
- Também observamos que as variáveis que têm melhor correlação com a variável "Lesionado?" são as variáveis: "Distância", "RPE", "Stress", "Dor" e depois vem a "Idade".
- Verifica-se também uma forte correlação positiva entre as variáveis "Distância" e "RPE"
- Conseguimos perceber que os jogadores dormem em média 7h.
- Os jogadores têm uma média de 25 anos.
- Mais de 2/3 das sessões praticadas são de treino e o resto de jogo.
- Neste conjunto de dados apenas 1% dos jogadores lesionou-se, o que pode condicionar os resultados.

7.1.2.4 Validação da qualidade dos dados

Não foi verificada nenhuma invalidação da qualidade dos dados e acredita-se que não haja nenhum problema em relação à qualidade dos dados.

7.1.3 Preparação dos dados

7.1.3.1 Seleção de variáveis

Ao prepararmos os dados vamos selecionar todas as variáveis numéricas e vamos descartar as variáveis categóricas, pois como referido, o objetivo é analisar as métricas físicas e biológicas dos jogadores e desta maneira não precisamos do nome ou tipo de sessão para a construção do modelo. A única exceção será a variável de saída que será a variável "Lesionado?".

7.1.3.2 Limpeza de dados

Neste conjunto de dados existiam 7 campos nulos em 6 jogadores e essas variáveis duração do sono, qualidade do sono, pontuação do sono, *stress*, *soreness* e RPE, totalizando assim 42 campos nulos. A solução encontrada foi a de preencher manualmente esses campos com valores médios, até porque não há valores muito altos que irão interferir com a média de todas as variáveis.

No conjunto de dados também encontramos 399 campos nulos do tipo de lesão, mas este campo foi descartado, pois não vamos utilizar em momento nenhum.

A Figura 31 e a Figura 32 ilustram estas ocorrências.

```
wk.isnull().sum()

Data                  0
Nome                  0
Posição               0
Tipo_Sessao           0
Idade                 0
Duracao_Sono          7
Pontuacao_Sono        7
Qualidade_Sono        7
Dor                   7
Stress                7
RPE                   7
Distancia_Total       0
Total_Aceleracoes     0
Max_Aceleracao        0
Total_Desaceleracao   0
Max_Desaceleracao     0
Max_Velocidade        0
Lesionado?            0
Tipo de Lesão       399
dtype: int64
```

Figura 31- Valores nulos do conjunto de dados (Fonte: Elaboração Própria)

Duracao_Sono	7.423645
Pontuacao_Sono	86.241379
Qualidade_Sono	7.603448
Dor	5.225369
Stress	6.190887
RPE	5.910099

Figura 32- Valores médios das variáveis que tinham campos nulos (Fonte: Elaboração Própria)

7.1.3.3 Cálculo de variáveis derivadas

Não foi necessário o cálculo de variáveis derivadas.

7.1.3.4 Integração de dados

Consideramos necessário incluir uma coluna no conjunto de dados que foi a coluna da idade, até porque a idade é um dos fatores influenciáveis no risco de lesões, pois quanto mais velho o jogador, maior é o risco de contrair uma lesão muscular porque a resistência muscular não é a mesma que um atleta mais novo.

7.1.3.5 Formatação de dados

Implementamos a formatação da variável de saída que tinha como resposta "No" ou "Yes", para "0" em substituição do "No" e "1" em detrimento do "Yes" com o método ".replace" de forma a substituir os valores, já que é mais confiável para alguns algoritmos a variável de saída ser codificada numericamente.

Também normalizamos as variáveis numéricas com o método "*StandardScaler()*", para que as variáveis com maior valor não parecessem que valiam mais que as variáveis de menor valor, ficando todas na mesma escala de valores.

7.1.4 Construção do modelo

7.1.4.1 Seleção das técnicas de modelagem

Para resolver o problema que deu origem a esta dissertação e atingir os objetivos propostos, vamos precisar de executar um algoritmo de classificação para que o modelo recolha dados históricos e depois aprenda a reconhecer alguns padrões. Este tipo de algoritmo é o mais utilizado em problemas da vida real e tenta prever o que vai acontecer no futuro segundo as variáveis que vai recolher.

Como já dito anteriormente os algoritmos que vamos utilizar são:

- K-NN
- Random Forest
- Regressão Logística
- Árvores de Decisão
- SVM

Apesar do SVM não ser um algoritmo de classificação, ele também é capaz de avaliar problemas binários ao traçar um hiperplano de maneira a distinguir duas classes, provando que é capaz de distinguir atletas lesionados ou não.

7.1.4.2 Definição do plano de testes

Nesta tarefa aplicamos a metodologia *holdout*, onde vamos separar os nossos dados em dois conjuntos, o conjunto de treino que serve para o algoritmo aprender a reconhecer os padrões e o conjunto de teste que, como o nome já o diz, serve para testar o modelo e medir a sua precisão.

Vamos dividir em 65/35, ou seja, 65% dos dados vão ser treinados e 35% vão ser testados.

7.1.4.3 Construção do modelo

Neste tipo de problemas de classificação é possível que o processo de treino e teste se prolongue, irá depender do número de instâncias colocadas e do tipo de algoritmo que vai estar a ser utilizado. Por exemplo, o K-NN só mede a distância para os seus vizinhos, é por essa razão que não demora muito a ser testado.

Depois do modelo ser contruído e testado, iremos partir para a avaliação do modelo e comparar entre todos os algoritmos qual foi o que indicou melhor acurácia e precisão.

Para construir este modelo foi utilizado o Google Colab, que é um produto do Google Research para pesquisas científicas [32]. Permite escrever e executar código Python arbitrário pelo navegador e é útil para este tipo de dados e análise que estamos a usar [32]

7.1.4.4 Avaliação do modelo

Depois de dividirmos os dados em um conjunto de dados de treino e um conjunto de dados de teste, que são geradas amostragens aleatórias dos dados, onde 65% dos dados foram

atribuídos ao conjunto de dados de treino, este conjunto de dados vai ser utilizado para medir a precisão do modelo e 35% ao conjunto de dados de teste como vemos na Figura 33.

Treino e Test Split

```
[568] from sklearn.model_selection import train_test_split

[569] x_wk_treino, x_wk_test, y_wk_treino, y_wk_test = train_test_split(x_wk, y_wk, test_size =0.35, random_state=0 )

[570] x_wk_treino.shape, y_wk_treino.shape

      ((263, 12), (263,))

[571] x_wk_test.shape, y_wk_test.shape

      ((143, 12), (143,))
```

Figura 33-Dados de teste e de treino (Fonte: Elaboração Própria)

Após a divisão do conjunto de dados de treino e o conjunto de dados de teste, começou-se a criar os classificadores de todos os algoritmos. Primeiro foi criado o classificador das **Árvore de Decisão**. Os parâmetros usados foram: "*criterion='entropy', random_state=1, min_samples_leaf=1, min_samples_split=10, splitter="random"* ", como podemos ver na Figura 34:

Árvore de Decisão

```
[575] from sklearn.tree import DecisionTreeClassifier

[576] arvore_wk = DecisionTreeClassifier(criterion= 'entropy', random_state=1 ,min_samples_leaf= 1, min_samples_split=10, splitter='random')
      arvore_wk.fit(x_wk_treino, y_wk_treino)

      DecisionTreeClassifier
      DecisionTreeClassifier(criterion='entropy', min_samples_split=10,
                             random_state=1, splitter='random')

[577] previsoes = arvore_wk.predict(x_wk_test)
      previsoes

      array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
             0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])

[578] y_wk_test

      array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Figura 34- Classificador da Árvore de Decisão (Fonte: Elaboração Própria)

A seguir podemos verificar o relatório de classificação e a acurácia das previsões na Figura 35:

```
[579] from sklearn.metrics import accuracy_score, classification_report

[580] accuracy_score(y_wk_test, previsoes)

      0.993006993006993

[581] print(classification_report(y_wk_test, previsoes))

                    precision    recall  f1-score   support

                 0       1.00      0.99      1.00       141
                 1       0.67      1.00      0.80         2

          accuracy                           0.99       143
         macro avg       0.83      1.00      0.90       143
      weighted avg       1.00      0.99      0.99       143
```

Figura 35- Relatório do classificador e acurácia Árvore de Decisão (Fonte: Elaboração Própria)

Observamos que a acurácia da árvore de decisão é de 0.99. E a seguir na Figura 36 podemos ver a matriz de confusão do classificador da **Árvore de Decisão**:

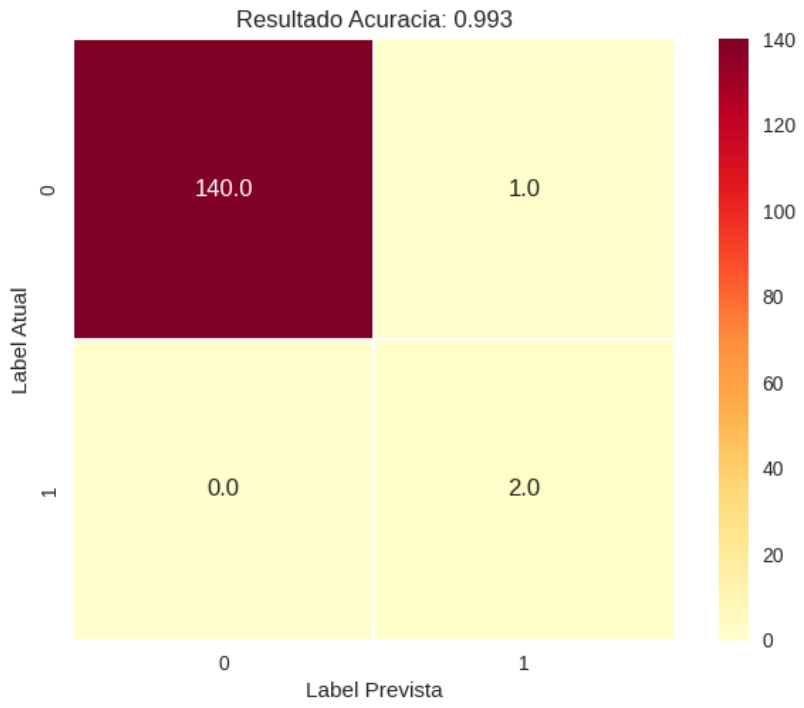

Figura 36- Matriz de confusão da Árvore de Decisões (Fonte: Elaboração Própria)

Nesta matriz pode-se perceber que o classificador da Árvore de Decisão apenas classificou 140 dados corretamente dos 143 originais, que foram os Verdadeiros Negativos. Dos 2 positivos o classificador, classificou corretamente os 2 e assim não obtivemos nenhum falso negativo e obtivemos 1 falso positivo.

Já na Equação 1, 2 e 3 podemos observar os erros deste classificador.

- **Erro Absoluto Médio**: Calcula a diferença média entre o valor real com o que foi previsto e é colocado um módulo pois podem existir valores positivos e negativos.

Equação 1- Erro Médio Absoluto (Fonte: [33]).

$$MAE\,(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

- **Erro Quadrático Médio**: A subtração entre a média do valor que prevê com o real.

Equação 2- Erro Quadrático Médio (Fonte: [33])

$$MSE\,(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- **Raiz de Erro Quadrático Médio**: O mesmo cálculo que o Erro Quadrático Médio, mas é aplicada uma raiz quadrada por toda a equação.

Equação 3- Raiz do Erro Quadrático Médio (Fonte: [33])

$$RMSE\,(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

Calculando, mostramos na Figura 37 que o erro absoluto médio é de 0.01%, o erro quadrático médio é de 0.01% e a raiz de erro quadrático médio é de 0.08%.

```
[652] from sklearn.metrics import mean_squared_error
      from sklearn.metrics import mean_absolute_error
      eam = mean_absolute_error(arvore_wk.predict(x_wk_test), y_wk_test)
      eqm = mean_squared_error(arvore_wk.predict(x_wk_test), y_wk_test)
      reqm =np.sqrt(eqm)
      print('Erro Absoluto Medio (MAE): {:.2f} %' .format(eam))
      print('Erro Quadratico Medio (MSE): {:.2f} %' .format(eqm))
      print('Raiz Erro Quadratico Medio (RMSE): {:.2f} %'.format(reqm))

      Erro Absoluto Medio (MAE): 0.01 %
      Erro Quadratico Medio (MSE): 0.01 %
      Raiz Erro Quadratico Medio (RMSE): 0.08 %
```

Figura 37-Erros da Árvore de Decisão (Fonte: Elaboração Própria)

A seguir foi criado o classificador do ***Random Forest***. Os parâmetros usados foram: "*criterion='entropy', random_state=0, n_estimators=10*", como podemos ver na Figura 38:

Random Forest

```
[584] from sklearn.ensemble import RandomForestClassifier

[585] random_forest_wk = RandomForestClassifier(n_estimators= 10, criterion='entropy', random_state=0)
      random_forest_wk.fit(x_wk_treino, y_wk_treino)

      RandomForestClassifier
      RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=0)

[586] previsoes = random_forest_wk.predict(x_wk_test)
      previsoes

      array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])

[587] y_wk_test

      array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Figura 38-Classificador do Random Forest(Fonte: Elaboração Própria)

A seguir podemos verificar o relatório de classificação e a acurácia das previsões do classificador **Random Forest**, que teve uma acurácia de 0.99, como mostra a Figura 39:

```
[588] accuracy_score(y_wk_test, previsoes)

      0.993006993006993

[589] print(classification_report(y_wk_test, previsoes))

                   precision    recall  f1-score   support

                0       0.99      1.00      1.00       141
                1       1.00      0.50      0.67         2

         accuracy                           0.99       143
        macro avg       1.00      0.75      0.83       143
     weighted avg       0.99      0.99      0.99       143
```

Figura 39-Relatório do classificador e da acurácia do Random Fores (Fonte: Elaboração Própria)t

Após isto podemos ver a matriz de confusão do classificador da **Random Forest** que classificou dos 143 dados 142 corretamente sendo 1 verdadeiro positivo e 141 verdadeiros negativos e apenas 1 falso positivo, como vemos na Figura 40:

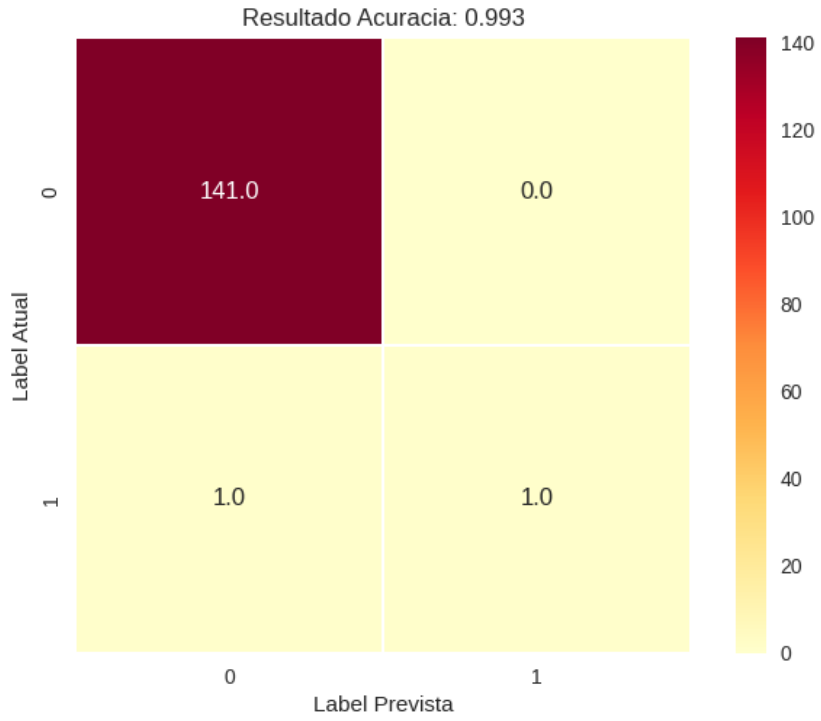

Figura 40- Matriz de confusão Random Forest (Fonte: Elaboração Própria)

Já na Figura 41, podemos observar os erros deste classificador. Vemos que o erro absoluto médio é de 0.01%, o erro quadrático médio é de 0.01% e a raiz de erro quadrático médio é de 0.08%.

```
[653] from sklearn.metrics import mean_squared_error
      from sklearn.metrics import mean_absolute_error
      eam = mean_absolute_error(random_forest_wk.predict(x_wk_test), y_wk_test)
      eqm = mean_squared_error(random_forest_wk.predict(x_wk_test), y_wk_test)
      reqm =np.sqrt(eqm)
      print('Erro Absoluto Medio (MAE): {:.2f} %' .format(eam))
      print('Erro Quadratico Medio (MSE): {:.2f} %' .format(eqm))
      print('Raiz Erro Quadratico Medio (RMSE): {:.2f} %'.format(reqm))

      Erro Absoluto Medio (MAE): 0.01 %
      Erro Quadratico Medio (MSE): 0.01 %
      Raiz Erro Quadratico Medio (RMSE): 0.08 %
```

Figura 41- Erros do Random Forest (Fonte: Elaboração Própria)

A seguir foi criado o classificador do **K-NN**, com os parâmetros "*n_neighbors=1, metric= minkowski, p =2*", como podemos ver abaixo na Figura 42:

KNN

```
[593] from sklearn.neighbors import KNeighborsClassifier

[594] knn_wk = KNeighborsClassifier(n_neighbors=1, metric='minkowski', p = 2)
      knn_wk.fit(x_wk_treino, y_wk_treino)

      KNeighborsClassifier
      KNeighborsClassifier(n_neighbors=1)

[595] previsoes = knn_wk.predict(x_wk_test)
      previsoes

      array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])

[596] y_wk_test

      array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Figura 42- Classificador do K-NN (Fonte: Elaboração Própria)

A seguir podemos verificar o relatório de classificação e a acurácia das previsões do classificador **K-NN,** que tem uma acurácia de 0.97, podemos ver na Figura 43:

```
accuracy_score(y_wk_test, previsoes)

0.972027972027972

print(classification_report(y_wk_test, previsoes))

              precision    recall  f1-score   support

           0       1.00      0.99      1.00       141
           1       0.67      1.00      0.80         2

    accuracy                           0.99       143
   macro avg       0.83      1.00      0.90       143
weighted avg       1.00      0.99      0.99       143
```

Figura 43- Relatório do classificador e acurácia do K-NN (Fonte: Elaboração Própria)

Após isto temos na Figura 44 podemos ver a matriz de confusão do classificador da **K-NN,** que dos 143 dados, classificou corretamente 139, sendo 138 verdadeiros negativos e 1 verdadeiro positivo, sendo os falsos 4, 3 falsos negativos e 1 falso positivo:

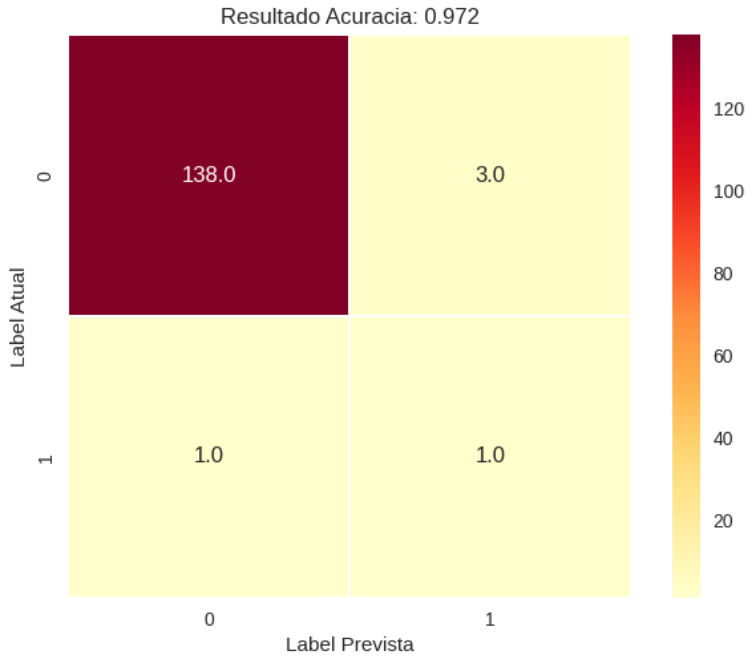

Figura 44- Matriz de confusão do KNN (Fonte: Elaboração Própria)

Já na Figura 45, podemos observar os erros deste classificador. Vemos que o erro absoluto médio é de 0.01%, o erro quadrático médio é de 0.01% e a raiz de erro quadrático médio é de 0.08%.

```
[654] from sklearn.metrics import mean_squared_error
      from sklearn.metrics import mean_absolute_error
      eam = mean_absolute_error(random_forest_wk.predict(x_wk_test), y_wk_test)
      eqm = mean_squared_error(random_forest_wk.predict(x_wk_test), y_wk_test)
      reqm =np.sqrt(eqm)
      print('Erro Absoluto Medio (MAE): {:.2f} %' .format(eam))
      print('Erro Quadratico Medio (MSE): {:.2f} %' .format(eqm))
      print('Raiz Erro Quadratico Medio (RMSE): {:.2f} %'.format(reqm))

      Erro Absoluto Medio (MAE): 0.01 %
      Erro Quadratico Medio (MSE): 0.01 %
      Raiz Erro Quadratico Medio (RMSE): 0.08 %
```

Figura 45- Erros do K-NN (Fonte: Elaboração Própria)

A seguir foi criado o classificador do **Regressão Logística** apenas com o parâmetro "*random_state=0*" como podemos ver abaixo na Figura 46:

Regressão Logística

```
[601] from sklearn.linear_model import LogisticRegression

[602] logistic_wk = LogisticRegression(random_state=0)
      logistic_wk.fit(x_wk_treino, y_wk_treino)

      LogisticRegression
      LogisticRegression(random_state=0)

[603] previsoes = logistic_wk.predict(x_wk_test)
      previsoes

      array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])

[604] y_wk_test

      array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Figura 46- Classificador da Regressão Logística (Fonte: Elaboração Própria)

A seguir podemos verificar o relatório de classificação e a acurácia das previsões do classificador **Regressão Logística**, onde observamos que tem a maior acurácia de todos que é de 1.0, podemos ver na Figura 47:

```
[605] accuracy_score(y_wk_test, previsoes)

      1.0

      print(classification_report(y_wk_test, previsoes))

                    precision    recall  f1-score   support

                 0       1.00      0.99      1.00       141
                 1       0.67      1.00      0.80         2

          accuracy                           0.99       143
         macro avg       0.83      1.00      0.90       143
      weighted avg       1.00      0.99      0.99       143
```

Figura 47- Relatório do classificador e acurácia da Regressão Logística (Fonte: Elaboração Própria)

Após isto temos na Figura 48 podemos ver a matriz de confusão do classificador da **Regressão Logística**, que classificou todos os 143 dados corretamente:

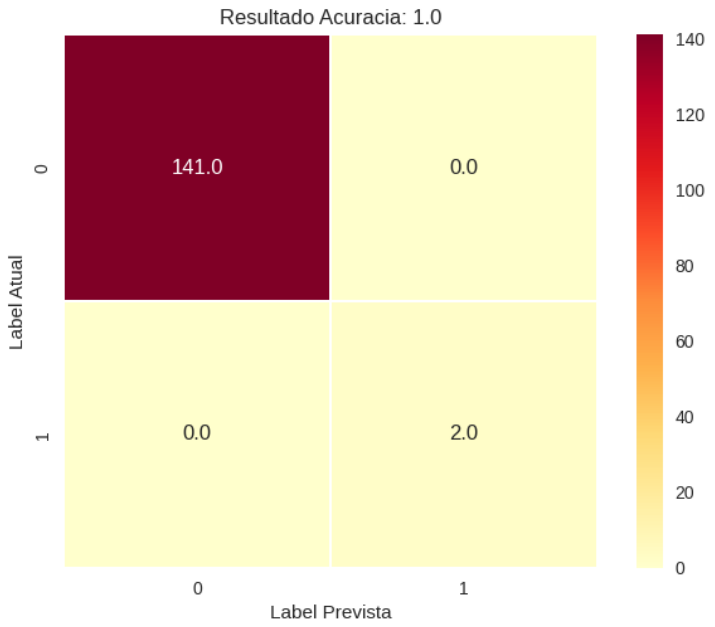

Figura 48- Matriz de confusão da Regressão Logística (Fonte: Elaboração Própria)

Já na Figura 49, podemos observar que este classificador não possui quaisquer erros.

```
[655] from sklearn.metrics import mean_squared_error
      from sklearn.metrics import mean_absolute_error
      eam = mean_absolute_error(logistic_wk.predict(x_wk_test), y_wk_test)
      eqm = mean_squared_error(logistic_wk.predict(x_wk_test), y_wk_test)
      reqm =np.sqrt(eqm)
      print('Erro Absoluto Medio (MAE): {:.2f} %' .format(eam))
      print('Erro Quadratico Medio (MSE): {:.2f} %' .format(eqm))
      print('Raiz Erro Quadratico Medio (RMSE): {:.2f} %'.format(reqm))

      Erro Absoluto Medio (MAE): 0.00 %
      Erro Quadratico Medio (MSE): 0.00 %
      Raiz Erro Quadratico Medio (RMSE): 0.00 %
```

Figura 49- Erros da Regressão Logística (Fonte: Elaboração Própria)

A seguir foi criado o classificador do **SVM**, com os parâmetros "*kernel='poly'*, *random_state=1*, *C=1.0* "como podemos ver na Figura 50:

```
SVM

[609] from sklearn.svm import SVC

[620] svm_wk = SVC(kernel='poly', random_state=1, C=1.0 )
      svm_wk.fit(x_wk_treino, y_wk_treino)

      SVC
      SVC(kernel='poly', random_state=1)

[621] previsoes = svm_wk.predict(x_wk_test)
      previsoes

      array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
             0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])

[622] y_wk_test

      array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Figura 50- Classificador do SVM (Fonte: Elaboração Própria)

Podemos verificar o relatório de classificação e a acurácia das previsões do classificador **SVM**, que possui uma acurácia de 0.99, como mostrado na Figura 51:

```
accuracy_score(y_wk_test, previsoes)
```

```
0.993006993006993
```

```
print(classification_report(y_wk_test, previsoes))
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       141
           1       0.67      1.00      0.80         2

    accuracy                           0.99       143
   macro avg       0.83      1.00      0.90       143
weighted avg       1.00      0.99      0.99       143
```

Figura 51-Relatório de classificação e acurácia SVM (Fonte: Elaboração Própria)

Após isto temos na Figura 52 podemos ver a matriz de confusão do classificador da **SVM**, que dos 143 dados, classificou corretamente 142, sendo 140 verdadeiros negativos e 2 verdadeiros positivos, sendo 1 falso negativo:

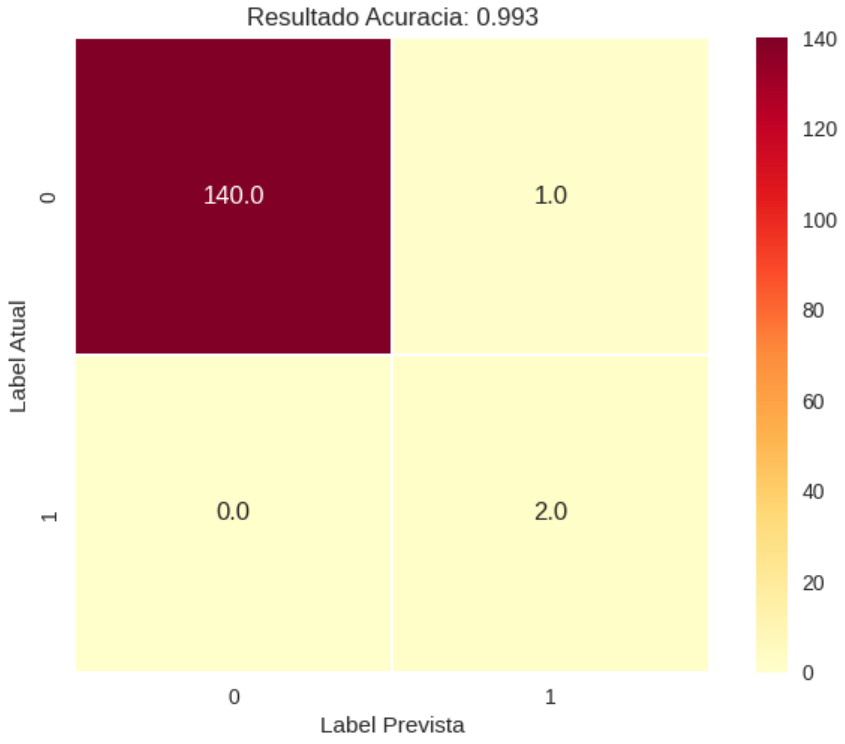

Figura 52- Matriz de confusão SVM (Fonte: Elaboração Própria)

Já na Figura 53, podemos observar os erros deste classificador. Vemos que o erro absoluto médio é de 0.01%, o erro quadrático médio é de 0.01% e a raiz de erro quadrático médio é de 0.08%.

```
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
eam = mean_absolute_error(svm_wk.predict(x_wk_test), y_wk_test)
eqm = mean_squared_error(svm_wk.predict(x_wk_test), y_wk_test)
reqm =np.sqrt(eqm)
print('Erro Absoluto Medio (MAE): {:.2f} %' .format(eam))
print('Erro Quadratico Medio (MSE): {:.2f} %' .format(eqm))
print('Raiz Erro Quadratico Medio (RMSE): {:.2f} %'.format(reqm))
```

```
Erro Absoluto Medio (MAE): 0.01 %
Erro Quadratico Medio (MSE): 0.01 %
Raiz Erro Quadratico Medio (RMSE): 0.08 %
```

Figura 53- Erros do SVM (Fonte: Elaboração Própria)

Após verificar a acurácia, relatórios e os erros dos classificadores, foi feito o *tuning* dos parâmetros dos classificadores, de forma a buscar os melhores parâmetros para obtermos o modelo K-Fold em que os dados tenham menor dispersão, isto é, um desvio padrão baixo.

O modelo K-Folds divide o conjunto de dados em k subconjuntos aleatórios iguais, e é treinado e testado k vezes.

Devido ao conjunto de dados ser pequeno, vamos dividir o nosso conjunto de dados em 5 vezes, como na Figura 54:

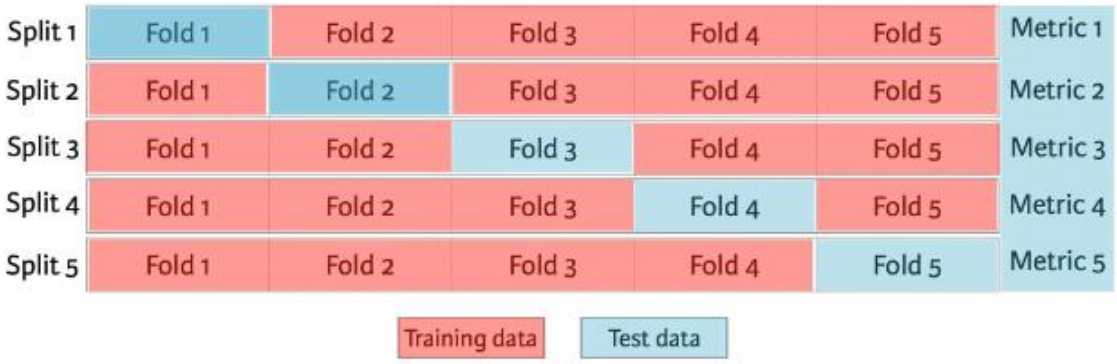

Figura 54- Modelo K-Folds dividido em 5 (Fonte: [38])

Foi criada uma lista para cada classificador, onde foram inseridos os 100 resultados da média de 5 execuções do modelo K-Folds ou também conhecida como validação cruzada, para não termos só 1 e sermos mais específicos e obtermos resultados mais refinados de cada

classificador. Na Figura 55 podemos observar os resultados estatísticos de cada classificador após o modelo K-Folds:

```
resultados.describe()
```

	Árvore	Random Forest	KNN	Logistica	SVM
count	100.000000	100.000000	100.000000	100.000000	100.000000
mean	98.214845	98.249202	98.276302	98.261488	98.276302
std	0.180157	0.076794	0.003563	0.059341	0.003563
min	97.530864	98.027702	98.271605	98.024691	98.271605
25%	98.273863	98.271605	98.274616	98.274616	98.274616
50%	98.274616	98.274616	98.274616	98.274616	98.274616
75%	98.277627	98.277627	98.277627	98.277627	98.277627
max	98.286661	98.286661	98.286661	98.286661	98.286661

Figura 55- Resultados estatísticos da validação cruzada (Fonte: Elaboração Própria)

Podemos verificar logo de imediato que as melhores médias são do classificador K-NN e do SVM, que consequentemente têm o menor desvio padrão, o que corresponde a uma menor dispersão dos dados, isto significa que estes modelos foram mais consistentes com a validação cruzada.

Também podemos observar que o classificador que teve a média mínima foi da Árvore de Decisão, foi o classificador que mais oscilou e por isso tem a maior dispersão de dados como podemos ver no seu desvio padrão. E todos os classificadores tiveram a média máximo, nenhum teve a sua melhor média maior que a dos outros.

7.1.5 Teste e avaliação

7.1.5.1 Avaliação dos resultados

Depois desta extensa análise de dados podemos verificar que:

- Os jogadores que contraíram lesões foram jogadores com <=6h de duração de Sono.
- Também foram os jogadores que tiveram valores de “Dor”, “Stress” e “RPE” acima de 8. E com distâncias percorridas em média de 8900m.
- A maioria dos jogadores lesionados tinham feito acelerações máximas médias de 6.71 m/s^2 e uma média de acelerações máximas de 5.83 m/s^2.
- O melhor classificador como pode-se observar na Tabela 2 foi o de Regressão Logística tendo uma acurácia de 100% para este conjunto de dados, apesar que por se tratar de um pequeno conjunto de dados isto possa ter influenciado nos

resultados. Apesar disso, não deixou de ter mais acurácia, precisão, sensibilidade e F1-score que os outros classificadores.

- O pior classificador foi o K-NN como podemos observar na Tabela 2, mas não podemos deixar de o considerar pois teve uma acurácia de 97.2%, 98% de precisão, 97% de sensibilidade, F1-Score de 98% e um erro quadrático médio de 0.01%.
- No Geral, todos os classificadores foram muito bons e satisfazem o objetivo da principal. Uma visão geral é mostrada na Tabela 2.

Tabela 2- Resultado do desempenho dos classificadores (Fonte: Elaboração Própria)

	Erro Quadrático Médio	Precisão	Sensibiidade	F1-Score	Acurácia
Árvore de Decisão	0.01%	1.00	0.99	0.99	0.993
Random Forest	0.01%	0.99	0.99	0.99	0.993
K-Neighbors	0.01%	0.98	0.97	0.98	0.972
Regressão Logística	0.00%	1.00	1.00	1.00	1.00
SVM	0.01%	1.00	0.99	0.99	0.993

As tarefas seguintes da fase “Teste e Implementação” e fase de “Implementação” não serão desenvolvidas teoricamente, apenas seria já em um contexto real, pois aqui seria praticamente o mesmo que desenvolver a secção das conclusões e do trabalho futuro e para não ser repetitivo.

8 Conclusão

Como conclusão deste trabalho, pode-se afirmar que todos os objetivos tiveram a sua resposta adequada e foram alcançados com sucesso. Concluiu-se que, primeiramente de acordo com os objetivos específicos, os clubes tiram partido das tecnologias para monitorar o desempenho dos jogadores em campo, ou seja, extrair toda a "informação" que o jogador esteja a produzir, tanto as suas movimentações, como os dados estatísticos dentro do jogo, tanto os biológicos e psicológicos também. Também utilizam a tecnologia para jovens talentosos que em um futuro próximo serão as próximas estrelas do futebol. Outra vantagem é a análise tática do adversário, melhorar a tomada de decisão e especialmente para prevenir e ajudar na recuperação de lesões musculares.

Chegou-se a compreender que os clubes analisam os seus jogadores com dispositivos *wearables* para extrair toda a informação física e biológica, através de coletes que os jogadores utilizam durante os treinos e jogos, assim como pulseiras ou relógios inteligentes para os jogadores utilizarem quando estiverem a dormir para medir o seu sono e respetiva qualidade. Os dados são encaminhados para um único repositório de dados, uma *cloud* ou um servidor, onde os cientistas e analistas vão trabalhar os dados e enviar para todos os departamentos, que então vão encaminhar o relatório para a equipa técnica que já vai obter os *insights* de maneira mais compreensiva.

Foi compreendido também nesta dissertação que atualmente os jogadores lesionam-se muito mais que antigamente, devido ao maior número de jogos e competições o que exige bastante esforço físico e mental do jogador, a carga de trabalho durante os treinos também é outro fator que sobrecarrega os jogadores e aumenta o risco de lesão, assim como também os alguns relvados em que os jogadores jogam que não são apropriados.

Determinou-se também que as métricas avaliadas para prevenir lesões nos jogadores incluem: os números de acelerações e desacelerações, distância percorrida, idade, RPE, duração do sono, o stress, os níveis de oxigénio dos atletas, os lactatos sanguíneos, a frequência cardíaca, a taxa de carga de trabalho aguda e crónica.

Concluiu-se ainda que, para os dados selecionados, o melhor modelo a ser proposto para utilização futura para se prever lesões de futebol é o modelo de regressão logística.

E por fim, com esta dissertação para ficou patente que por detrás de muito esforço físico existe um trabalho muito grande, não só os jogadores e treinadores que trazem sucesso para os clubes de futebol, mas sim também de todo o *staff* responsável pela análise (analistas e cientistas de dados), pois eles entregam muitos *insights* fundamentais, tanto para o sucesso desportivo

mas também para a saúde dos próprios atletas, já que não podemos esquecer que é este departamento de análise que entrega os dados ao departamento médico para avaliar o estado dos jogadores. Também ficou compreendido que os desportos no geral têm ainda muito a evoluir porque a análise de dados possui ferramentas poderosas para dar *insights* que levam ao sucesso desportivo e consequentemente financeiro.

9 Trabalho futuro

Com este trabalho ficou percebido que haveria maneira de se dar continuidade em um trabalho futuro, pois durante a sua evolução ficou evidente que estavam abertas as portas para novos projetos relacionados com este.

Com a dificuldade em encontrar dados reais para construir um modelo preditivo para prevenir lesões, sugere-se que para trabalhos futuros faça-se um modelo preditivo com um conjunto de dados reais e que, de preferência, inclua a variável do ACWR que contém dados biológicos, para aproximar-se ainda mais do mundo real. Também durante a pesquisa foi percebido que poderia ser feito um algoritmo que acumulasse a carga dos jogadores nos vários dias de trabalho, avisando que esse jogador corre um certo nível de contrair uma lesão de acordo com a sua carga de trabalho. Isto é possível, de acordo com as pesquisas com o ACWR neste trabalho.

Juntamente com este algoritmo e em uma fase posterior, poder-se-ia armazenar os dados dos jogadores em uma base de dados e enviar os dados para um *frontend* que simule uma intranet do clube de futebol de um certo departamento a consultar os dados dos jogadores em um determinado período de tempo.

Bibliografia

[1] M. Hagglund, M. Waldén, H. Magnusson, K. Kristenson, H. Bengtsson e J. Ekstrand, *Injuries affect team performance negatively in professional football: an 11-year follow-up of the UEFA Champions League injury study,* 3 Maio 2013.

[2] A. Zech e K. Wellmann, "Perceptions of football players regarding injury risk factors and prevention strategies," *PLoS ONE,* 2017.

[3] M. Stein, H. Janetzko, D. Seebacher, A. Jäger, M. Nagel, J. Hölsch, S. Kosub, T. Schreck, D. A.Keim e M. Grossniklaus, *How to Make Sense of Team Sport Data: From Acquisition to Data Modeling and Research Aspects,* 1 Janeiro 2017.

[4] J. Ekstrand, H. Bengtsson, A. Hallén, M. Vouillamoz e N. Papadimitriou, "UEFA Elite Club Injury Study Report 2016/17," Union of European Football Associations, 2017.

[5] J. Ekstrand, H. Bengtsson, A. Hallén, M. Vouillamoz e N. Papadimitriou, "UEFA Elite Club Injury Study Report 2019/20," Union of European Football Associations, 2020.

[6] S. Anthony, "Ars Technica," 24 Maio 2017. [Online]. Available: https://arstechnica.com/science/2017/05/football-data-tech-best-players-in-the-world/. [Acedido em 6 Janeiro 2023].

[7] F. Henriques, "HOW CAN BIG DATA HELP FOOTBALL CLUBS ACHIEVE COMPETITIVE ADVANTAGE," 26 Março 2018.

[8] D. T. Kirkendall e J. Dvorak, "The Physician and Sportsmedicine," *Effective Injury Prevention in Soccer,* pp. 147-157, 13 Março 2015.

[9] A. Teymourlouei, *Prevention of Hamstring Injuries in Male Soccer Athletes,* 20 Junho 2022.

[10] A. Majumdar, R. Bakirov, D. Hodges, S. Scott e T. Rees, "Machine Learning for Understanding and Predicting Injuries in Football," *Sports Medicine - Open,* 2022.

[11] G. Theron, "The use of Data Mining for Predicting Injuries in Professional Football Players," 2020.

[12] T. Gabbett e N. Domrow, “Relationships between training load, injury, and fitness in sub-elite collision sport athletes,” *Journal of Sports Sciences,* 2007.

[13] P. C. Bourdon, M. Cardinale, A. Murray, P. Gastin, M. Kellmann, M. C. Varley, T. J. Gabbett, A. J. Coutts, D. J. Burgess, W. Gregson e N. T. Cable, “Monitoring Athlete Training Loads: Consensus Statement,” *International Journal of Sports Physiology and Performance,* vol. 12, nº 161-170, 2017.

[14] A. Rossi, L. Pappalardo, P. Cintia, F. M. Iaia, J. Fernàndez e D. Medina, “Effective injury forecasting in soccer with GPS training data and machine learning,” *PLos ONE,* 2018.

[15] A. Satvedi e D. R. Pyne, “Injury Predictor For Soccer Players Using Machine Learning,” em *World Academy of Science, Engineering and Technology,* 2020.

[16] J. D. Ruddy, S. J. Cormack, R. Whiteley, M. D. Williams, R. G. Timmins e D. A. Opar, “Modeling the Risk of Team Sport Injuries: A Narrative Review of Different Statistical Approaches,” *Frontiers in Physiology,* vol. 10, 2019.

[17] S. Shalev-Shwartz e S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.

[18] J. Quinlan, “Machine Learning,” *Induction of Decision Trees,* pp. 81-106, Março 1986.

[19] S. Buchan, “Electronic Specifier,” 1 Maio 2018. [Online]. Available: https://www.electronicspecifier.com/news/analysis/a-journey-through-the-history-of-football-technology. [Acedido em 30 Junho 2023].

[20] G. Coimbra, “Futebol Interativo,” 4 Dezembro 2020. [Online]. Available: https://futebolinterativo.com/blog/afinal-para-que-serve-o-gps-nos-atletas. [Acedido em 30 Junho 2023].

[21] G. M. Arastey, “SPORT PERFORMANCE ANALYSIS,” 28 Junho 2018. [Online]. Available: https://www.sportperformanceanalysis.com/article/gps-in-professional-sports. [Acedido em 28 Junho 2023].

[22] FIFA, “Video Assistant Referee Technology,” 8 Dezembro 2022. [Online]. Available: https://www.fifa.com/technical/football-technology/football-technologies-and-innovations-at-the-fifa-world-cup-2022/video-assistant-referee-var. [Acedido em 30 Junho 2023].

[23] C. H. Almeida, “Linha de passe,” 12 Agosto 2019. [Online]. Available: https://linhadepasse.blogspot.com/2019/08/implicacoes-da-introducao-do-var-no.html. [Acedido em 30 Junho 2023].

[24] A. SHAHEEN, “The Use of Technology in Football - From the Ball to VAR,” *theSporting.blog.*

[25] SLBENFICA, “Benfica LAB,” [Online]. Available: https://www.slbenfica.pt/pt-pt/futebol-formacao/benfica-lab. [Acedido em 30 Junho 2023].

[26] A. Kajumba e S. Mokbel, “Daily Mail,” 20 Março 2020. [Online]. Available: https://www.dailymail.co.uk/sport/football/article-8136705/Premier-League-clubs-monitor-players-break-sending-tracking-devices-wear.html. [Acedido em 30 Junho 2023].

[27] H. Minds, “Happiest Minds,” [Online]. Available: https://www.happiestminds.com/insights/wearable-technology/#:~:text=Wearables%20are%20electronic%20technology%20or%20devices%20incorporated%20into,sync%20them%20with%20mobile%20devices%20or%20laptop%20computers.. [Acedido em 30 Junho 2023].

[28] K. Casey, “Codersera,” 2020 Junho 2020. [Online]. Available: https://codersera.com/blog/what-is-wearable-technology-how-it-works/. [Acedido em 30 Junho 2023].

[29] R. White, “SCIENCE FOR SPORT,” 26 Novembro 2017. [Online]. Available: https://www.scienceforsport.com/acutechronic-workload-ratio/. [Acedido em 1 Julho 2023].

[30] M. N. Center, “Microsoft,” 2017 Outubro 2017. [Online]. Available: https://news.microsoft.com/pt-pt/2017/10/05/tecnologia-microsoft-ajuda-sl-benfica-criar-equipa-futuro-2/. [Acedido em 2 Julho 2023].

[31] N. Hotz, “Data Science Process Alliance,” 19 Janeiro 2023. [Online]. Available: https://www.datascience-pm.com/crisp-dm-2/. [Acedido em 5 Julho 2023].

[32] Google, “Google,” [Online]. Available: https://research.google.com/colaboratory/intl/pt-BR/faq.html. [Acedido em 10 Julho 2023].

[33] C. d. O. Júnior, “Medium,” 13 Dezembro 2021. [Online]. Available: https://medium.com/data-hackers/prevendo-n%C3%BAmeros-entendendo-m%C3%A9tricas-de-regress%C3%A3o-35545e011e70. [Acedido em 10 Julho 2023].

[34] S. Austin, “How Benfica & Cricket Australia make sense of data,” 6 Maio 2017.

[35] Transfermarkt, “JOGADORES COM MAIS JOGOS 20/21,” 2021. [Online]. Available: https://www.transfermarkt.pt/meisteeinsaetze/gesamteinsaetze/statistik/2020/plus/1/galerie/0?saison_id=2020&wettbewerb_id=&land_id=0&altersklasse=u23&yt0=Mostrar. [Acedido em 29 Junho 2023].

[36] S. K, “Live the game. Live the moment. FIFA+ augmented reality experience,” 26 Novembro 2022. [Online]. Available: https://www.linkedin.com/pulse/live-game-moment-fifa-augmented-reality-experience-suresh-k. [Acedido em 30 Junho 2023].

[37] C. Gondo, “Medium,” 26 Abril 2016. [Online]. Available: https://medium.com/@CaioGondo/o-que-%C3%A9-essa-tal-intensidade-no-futebol-a158399b3f16. [Acedido em 30 Junho 2023].

[38] E. Allibhai, “Medium,” 3 Outubro 2018. [Online]. Available: https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f. [Acedido em 6 Julho 2023].

Anexo A – Cronograma

Atividades/Meses	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set
Escolha do tema	X											
Realização de leituras e investigação	X	X	X									
Escrita da Introdução, Objetivos e Revisão da Literatura e Metodologia		X	X	X								
Pesquisa prática sobre estudo de ferramentas para componente prática				X	X	X	X					
Desenvolvimento do modelo					X	X	X	X				
Descrição das componentes práticas na Dissertação								X	X			
Reuniões com orientador	X	X	X	X		X	X	X	X	X	X	X
Revisão Final										X		
Entrega preliminar											X	
Preparação para defesa											X	X
Apresentação final												X

Anexo B – Script Python

```
# -*- coding: utf-8 -*-
"""Untitled1.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/13DWZyeQih7BN7Y6cRDx29KnS0sbkgIGh
"""

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
from sklearn.metrics import confusion_matrix as cm
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import cross_val_score, KFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

wk = pd.read_csv('/content/Workout_Routine_Dirty_Original.csv', sep=';')
wk

wk.rename(columns = {'Date':'Data',
                             'Name':'Nome',
                             'Position':'Posição',
                             'Age':'Idade',
                             'Session_Type':'Tipo_Sessao',
                             'Sleep_Duration':'Duracao_Sono',
                             'Sleep_Score':'Pontuacao_Sono',
                             'Sleep_Quality':'Qualidade_Sono',
                             'Soreness':'Dor',
                             'Stress':'Stress',
                             'Distance':'Distancia',
                             'Acceleration_Count':'Total_Aceleracoes',
                             'Max_Acceleration':'Max_Aceleracao',
                             'Deceleration_Count':'Total_Desaceleracoes',
                             'Max_Deceleration':'Max_Desaceleracao',
                             'Max_Speed':'Max_Velocidade',
                             'Injury_Illness':'Lesionado?',
                             'Injury_Type':'Tipo_de_Lesao'}, inplace = True)
wk

wk['Lesionado?'] = wk['Lesionado?'].replace({'Yes': 1, 'No': 0})
```

```
wk

lesionados = wk[wk['Lesionado?'] == 1]
lesionados

lesionados = wk[wk['Lesionado?'] == 1]
lesionados.mean()

wk.isnull().sum()

wk.describe()

wk                                                                    =
pd.read_csv('C:\Users\david\Desktop\dataset_dissertacao\Workout_Routine_Di
rty.csv', sep=';')
wk

wk.describe()

wk.rename(columns = {
                            'Name':'Nome',
                            'Position':'Posição',
                            'Age':'Idade',
                            'Session_Type':'Tipo_Sessao',
                            'Sleep_Duration':'Duracao_Sono',
                            'Sleep_Score':'Pontuacao_Sono',
                            'Sleep_Quality':'Qualidade_Sono',
                            'Soreness':'Dor',
                            'Stress':'Stress',
                            'Distance':'Distancia',
                            'Acceleration_Count':'Total_Aceleracoes',
                            'Max_Acceleration':'Max_Aceleracao',
                            'Deceleration_Count':'Total_Desaceleracoes',
                            'Max_Deceleration':'Max_Desaceleracao',
                            'Max_Speed':'Max_Velocidade',
                            'Injury_Illness':'Lesionado?'}, inplace = True)
wk['Lesionado?'] = wk['Lesionado?'].replace({'Yes': 1, 'No': 0})
wk

correlacao = wk.corr(method = "pearson")
plt.figure(figsize = (25,10))
sns.heatmap(correlacao,vmax = 1, square = True, annot = True, cmap="YlOrRd")
plt.show()

#Definicao de variaveis continuas e impressao do nome das colunas
variaveis_cont = wk.describe().columns
print(variaveis_cont)

#Definicao de variaveis categoricas e impressao do nome das colunas
variaveis_categ = wk.describe(include=[object]).columns
print(variaveis_categ)

wk.hist(column = variaveis_cont, figsize = (20,20))
plt.show()
```

```
#Criacao dos graficos de barras, sobre as variaveis categoricas
fig, axes = plt.subplots(4, 4, figsize = (20, 20))
plt.subplots_adjust(left = None, bottom = None, right = None, top = None,
wspace = 0.7, hspace = 0.3)
for i, ax in enumerate(axes.ravel()):
    if i > 20:
        ax.set_visible(False)
        continue
    sns.countplot(y = variaveis_categ[i], data = wk, ax = ax)
plt.show()

#Criacao das boxplots, sobre as variaveis numericas
fig, axes = plt.subplots(4, 3, figsize = (20, 20))
plt.subplots_adjust(left = None, bottom = None, right = None, top = None,
wspace = 0.7, hspace = 0.3)
for i, ax in enumerate(axes.ravel()):
    if i > 9:
        ax.set_visible(False)
        continue
    sns.boxplot(x = variaveis_cont[i], data = wk, ax = ax)
plt.show()

wk.isnull().sum()

wk.describe()

wk.info()

wk.mean()

colunas_a_manter = [4,5,7,8,9,10,11,12,13,14,15,16]
x_wk = wk.iloc[:, colunas_a_manter].values
x_wk

type(x_wk)

y_wk = wk.iloc[:,17].values
y_wk

type(y_wk)

"""Normalizar"""

scaler_wk = StandardScaler()
x_wk = scaler_wk.fit_transform(x_wk)

x_wk[:,1].max(), x_wk[:, 0].min(), x_wk[:, 0].max()
#Valores vão estar na mesma escala

"""Treino e Test Split"""
```

```
x_wk_treino, x_wk_test, y_wk_treino, y_wk_test = train_test_split(x_wk,
y_wk, test_size =0.35, random_state=0 )

x_wk_treino.shape, y_wk_treino.shape

x_wk_test.shape, y_wk_test.shape

x_wk_treino

"""Árvore de Decisão"""

arvore_wk = DecisionTreeClassifier(criterion= 'entropy', random_state=1
,min_samples_leaf= 1, min_samples_split=10, splitter='random')
arvore_wk.fit(x_wk_treino, y_wk_treino)

previsoes = arvore_wk.predict(x_wk_test)
previsoes

y_wk_test

accuracy_score(y_wk_test, previsoes)

print(classification_report(y_wk_test, previsoes))


resultados_rf = round(accuracy_score(y_wk_test, previsoes),3)
cm1 = cm(y_wk_test, previsoes)
sns.heatmap(cm1, annot=True, fmt=".1f", linewidths =.3,
        square = True, cmap = 'YlOrRd')
plt.ylabel('Label Atual')
plt.xlabel('Label Prevista')
plt.title('Resultado Acuracia: {0}'.format(resultados_rf), size = 12)
plt.show()

eam = mean_absolute_error(arvore_wk.predict(x_wk_test), y_wk_test)
eqm = mean_squared_error(arvore_wk.predict(x_wk_test), y_wk_test)
reqm =np.sqrt(eqm)
print('Erro Absoluto Medio (MAE): {:.2f} %' .format(eam))
print('Erro Quadratico Medio (MSE): {:.2f} %' .format(eqm))
print('Raiz Erro Quadratico Medio (RMSE): {:.2f} %'.format(reqm))

"""K-Folds AD

"""

acuracias_ad = cross_val_score(arvore_wk, X = x_wk_treino, y = y_wk_treino,
cv = 5)
print("Acuracia (media): {:.2f} %" .format(acuracias_ad.mean() * 100))
print("Desvio Padrao: {:.2f} %" .format(acuracias_ad.std() * 100))

res_ad = cross_val_score(arvore_wk, x_wk_treino, y_wk_treino,
scoring='neg_mean_absolute_error', cv=5,)
print ("Erro Absoluto Medio: {:.2f} %\n" .format(res_ad.mean()))
```

```
"""Random Forest"""


random_forest_wk      =      RandomForestClassifier(n_estimators=      10,
criterion='entropy', random_state=0)
random_forest_wk.fit(x_wk_treino, y_wk_treino)

previsoes = random_forest_wk.predict(x_wk_test)
previsoes

y_wk_test

accuracy_score(y_wk_test, previsoes)

print(classification_report(y_wk_test, previsoes))


resultados_rf = round(accuracy_score(y_wk_test, previsoes),3)
cm1 = cm(y_wk_test, previsoes)
sns.heatmap(cm1, annot=True, fmt=".1f", linewidths =.3,
        square = True, cmap = 'YlOrRd')
plt.ylabel('Label Atual')
plt.xlabel('Label Prevista')
plt.title('Resultado Acuracia: {0}'.format(resultados_rf), size = 12)
plt.show()

print(classification_report(y_wk_test, previsoes))

eam = mean_absolute_error(random_forest_wk.predict(x_wk_test), y_wk_test)
eqm = mean_squared_error(random_forest_wk.predict(x_wk_test), y_wk_test)
reqm =np.sqrt(eqm)
print('Erro Absoluto Medio (MAE): {:.2f} %' .format(eam))
print('Erro Quadratico Medio (MSE): {:.2f} %' .format(eqm))
print('Raiz Erro Quadratico Medio (RMSE): {:.2f} %'.format(reqm))

"""K-FOLDS RF"""

acuracias_rf = cross_val_score(arvore_wk, X = x_wk_treino, y = y_wk_treino,
cv = 5)
print("Acuracia (media): {:.2f} %" .format(acuracias_rf.mean() * 100))
print("Desvio Padrao: {:.2f} %" .format(acuracias_rf.std() * 100))

res_rf    =    cross_val_score(arvore_wk,    x_wk_treino,    y_wk_treino,
scoring='neg_mean_absolute_error', cv=5,)
print ("Erro Absoluto Medio: {:.2f} %\n" .format(res_rf.mean()))

"""KNN"""


knn_wk = KNeighborsClassifier(n_neighbors=1, metric='minkowski', p = 2)
knn_wk.fit(x_wk_treino, y_wk_treino)

previsoes = knn_wk.predict(x_wk_test)
previsoes
```

```
y_wk_test

accuracy_score(y_wk_test, previsoes)

print(classification_report(y_wk_test, previsoes))


resultados_knn = round(accuracy_score(y_wk_test, previsoes),3)
cm1 = cm(y_wk_test, previsoes)
sns.heatmap(cm1, annot=True, fmt=".1f", linewidths =.3,
        square = True, cmap = 'YlOrRd')
plt.ylabel('Label Atual')
plt.xlabel('Label Prevista')
plt.title('Resultado Acuracia: {0}'.format(resultados_knn), size = 12)
plt.show()

print(classification_report(y_wk_test,  previsoes))  ###teestar  com  mais
parametros n_neighbors

eam = mean_absolute_error(random_forest_wk.predict(x_wk_test), y_wk_test)
eqm = mean_squared_error(random_forest_wk.predict(x_wk_test), y_wk_test)
reqm =np.sqrt(eqm)
print('Erro Absoluto Medio (MAE): {:.2f} %' .format(eam))
print('Erro Quadratico Medio (MSE): {:.2f} %' .format(eqm))
print('Raiz Erro Quadratico Medio (RMSE): {:.2f} %'.format(reqm))

"""K-FOLDS KNN"""

acuracias_knn = cross_val_score(arvore_wk, X = x_wk_treino, y = y_wk_treino,
cv = 5)
print("Acuracia (media): {:.2f} %" .format(acuracias_knn.mean() * 100))
print("Desvio Padrao: {:.2f} %" .format(acuracias_knn.std() * 100))

res_knn    =    cross_val_score(arvore_wk,    x_wk_treino,    y_wk_treino,
scoring='neg_mean_absolute_error', cv=5,)
print ("Erro Absoluto Medio: {:.2f} %\n" .format(res_knn.mean()))

"""Regressão Logística"""


logistic_wk = LogisticRegression(random_state=0)
logistic_wk.fit(x_wk_treino, y_wk_treino)

previsoes = logistic_wk.predict(x_wk_test)
previsoes

y_wk_test

accuracy_score(y_wk_test, previsoes)

print(classification_report(y_wk_test, previsoes))
```

```
resultado_rl = round(accuracy_score(y_wk_test, previsoes),3)
cm1 = cm(y_wk_test, previsoes)
sns.heatmap(cm1, annot=True, fmt=".1f", linewidths =.3,
        square = True, cmap = 'YlOrRd')
plt.ylabel('Label Atual')
plt.xlabel('Label Prevista')
plt.title('Resultado Acuracia: {0}'.format(resultado_rl), size = 12)
plt.show()

print(classification_report(y_wk_test, previsoes))

eam = mean_absolute_error(logistic_wk.predict(x_wk_test), y_wk_test)
eqm = mean_squared_error(logistic_wk.predict(x_wk_test), y_wk_test)
reqm =np.sqrt(eqm)
print('Erro Absoluto Medio (MAE): {:.2f} %' .format(eam))
print('Erro Quadratico Medio (MSE): {:.2f} %' .format(eqm))
print('Raiz Erro Quadratico Medio (RMSE): {:.2f} %'.format(reqm))

acuracias_rl = cross_val_score(arvore_wk, X = x_wk_treino, y = y_wk_treino,
cv = 5)
print("Acuracia (media): {:.2f} %" .format(acuracias_rl.mean() * 100))
print("Desvio Padrao: {:.2f} %" .format(acuracias_rl.std() * 100))

res_rl    =    cross_val_score(arvore_wk,    x_wk_treino,    y_wk_treino,
scoring='neg_mean_absolute_error', cv=5,)
print ("Erro Absoluto Medio: {:.2f} %\n" .format(res_rl.mean()))

"""SVM"""


svm_wk = SVC(kernel='poly', random_state=1, C=1.0 )
svm_wk.fit(x_wk_treino, y_wk_treino)

previsoes = svm_wk.predict(x_wk_test)
previsoes

y_wk_test

accuracy_score(y_wk_test, previsoes)

print(classification_report(y_wk_test, previsoes))


resultado_svm = round(accuracy_score(y_wk_test, previsoes),3)
cm1 = cm(y_wk_test, previsoes)
sns.heatmap(cm1, annot=True, fmt=".1f", linewidths =.3,
        square = True, cmap = 'YlOrRd')
plt.ylabel('Label Atual')
plt.xlabel('Label Prevista')
plt.title('Resultado Acuracia: {0}'.format(resultado_svm), size = 12)
plt.show()

print(classification_report(y_wk_test, previsoes))
```

```
eam = mean_absolute_error(svm_wk.predict(x_wk_test), y_wk_test)
eqm = mean_squared_error(svm_wk.predict(x_wk_test), y_wk_test)
reqm =np.sqrt(eqm)
print('Erro Absoluto Medio (MAE): {:.2f} %' .format(eam))
print('Erro Quadratico Medio (MSE): {:.2f} %' .format(eqm))
print('Raiz Erro Quadratico Medio (RMSE): {:.2f} %'.format(reqm))

acuracias_svm = cross_val_score(arvore_wk, X = x_wk_treino, y = y_wk_treino,
cv = 5)
print("Acuracia (media): {:.2f} %" .format(acuracias_svm.mean() * 100))
print("Desvio Padrao: {:.2f} %" .format(acuracias_svm.std() * 100))

res_svm    =    cross_val_score(arvore_wk,    x_wk_treino,    y_wk_treino,
scoring='neg_mean_absolute_error', cv=5,)
print ("Erro Absoluto Medio: {:.2f} %\n" .format(res_svm.mean()))

"""Tuning dos Parâmetros com GridSearch"""

x_wk = np.concatenate((x_wk_treino, x_wk_test), axis=0)
x_wk.shape

x_wk

y_wk

"""Tuning dos Parâmetros Árvores de Decisão"""

parametros = {'criterion': ['gini', 'entropy'],
              'splitter': ['best', 'random'],
              'min_samples_split': [2, 5, 10],
              'min_samples_leaf': [1, 5, 10]}

grid_search       =       GridSearchCV(estimator=DecisionTreeClassifier(),
param_grid=parametros)
grid_search.fit(x_wk, y_wk)
melhores_parametros = grid_search.best_params_
melhor_resultado = grid_search.best_score_
print(melhores_parametros)
print(melhor_resultado)

"""Tuning dos Parâmetros RF"""

parametros = {'criterion': ['gini', 'entropy'],
              'n_estimators': [10, 40, 100, 150],
              'min_samples_split': [2, 5, 10],
              'min_samples_leaf': [1, 5, 10]}

grid_search       =       GridSearchCV(estimator=RandomForestClassifier(),
param_grid=parametros)
grid_search.fit(x_wk, y_wk)
melhores_parametros = grid_search.best_params_
melhor_resultado = grid_search.best_score_
print(melhores_parametros)
print(melhor_resultado)
```

```
"""Tuning dos Parâmetros KNN"""

parametros = {'n_neighbors':[3,5,10,20],
              'p':[1,2]}

grid_search         =         GridSearchCV(estimator=KNeighborsClassifier(),
param_grid=parametros)
grid_search.fit(x_wk, y_wk)
melhores_parametros = grid_search.best_params_
melhor_resultado = grid_search.best_score_
print(melhores_parametros)
print(melhor_resultado)

"""Tuning dos Parâmetros Regr Logistica"""

parametros = { 'random_state':[1,2,5,10],
              'tol': [0.001, 0.00001, 0.000001],
              'C':[1.0,1.5,2.0],
              'solver':['lbfgs','sag','saga']}

grid_search          =          GridSearchCV(estimator=LogisticRegression(),
param_grid=parametros)
grid_search.fit(x_wk, y_wk)
melhores_parametros = grid_search.best_params_
melhor_resultado = grid_search.best_score_
print(melhores_parametros)
print(melhor_resultado)

"""Tuning dos Parâmetros SVM"""

parametros = {'tol':[0.001,0.0001,0.00001],
              'C':[1.0,1.5,2.0],
              'kernel': ['rbf', 'linear','sigmoid']}

grid_search = GridSearchCV(estimator=SVC(), param_grid=parametros)
grid_search.fit(x_wk, y_wk)
melhores_parametros = grid_search.best_params_
melhor_resultado = grid_search.best_score_
print(melhores_parametros)
print(melhor_resultado)

"""Validação Cruzada"""


resultados_arvore = []
resultados_rf = []
resultados_knn = []
resultados_rl = []
resultados_svm = []
for i in range(100):
  kfold = KFold(n_splits=5, shuffle=True, random_state=i)
```

```
  arvore = DecisionTreeClassifier(criterion='gini', min_samples_leaf=5,
min_samples_split=2, splitter='best')
  scores =cross_val_score(arvore, x_wk, y_wk, cv=kfold)
  resultados_arvore.append(scores.mean()*100)
  ###############################################parametros baseados nos
melhores
#random forest
  rf = RandomForestClassifier(criterion='gini', min_samples_leaf= 1,
min_samples_split= 2, n_estimators= 10)
  scores =cross_val_score(rf, x_wk, y_wk, cv=kfold)
  resultados_rf.append(scores.mean() *100)

  ########################################################
#knn
  knn = KNeighborsClassifier()
  scores =cross_val_score(knn, x_wk, y_wk, cv=kfold)
  resultados_knn.append(scores.mean()*100)
  ########################################################
#rl

  rl= LogisticRegression(C=1.0, random_state=1, solver='lbfgs', tol= 0.001)
  scores =cross_val_score(rl, x_wk, y_wk, cv=kfold)
  resultados_rl.append(scores.mean()*100)

  ########################################################
  #svm
  svm= SVC(C= 1.0, kernel='rbf', tol= 0.001)
  scores =cross_val_score(svm, x_wk, y_wk, cv=kfold)
  resultados_svm.append(scores.mean() * 100)

print('Resultados das árvores:', resultados_arvore)
print('Resultados do knn:', resultados_knn)
print('Resultados do rf:', resultados_rf)
print('Resultados da rl:', resultados_rl)
print('Resultados do svm:', resultados_svm)

resultados = pd.DataFrame({'Árvore': resultados_arvore, 'Random Forest':
resultados_rf,       'KNN':resultados_knn,       'Logistica':resultados_rl,
'SVM':resultados_svm})
resultados

resultados.describe()

resultados.mean()

resultados.var() *100

"""variância em %"""

(resultados.std() / resultados.mean() * 100)
```