UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS DE CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO

NICHOLAS SANGOI BRUTTI

APRENDIZADO DE MÁQUINA APLICADO À PREVISÃO DA EFETIVIDADE DE SUBSTITUIÇÕES DE JOGADORES NO CAMPEONATO BRASILEIRO DE FUTEBOL SÉRIE A

CHAPECÓ
2019

NICHOLAS SANGOI BRUTTI

APRENDIZADO DE MÁQUINA APLICADO À PREVISÃO DA EFETIVIDADE DE SUBSTITUIÇÕES DE JOGADORES NO CAMPEONATO BRASILEIRO DE FUTEBOL SÉRIE A

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.
Orientador: Prof. Dr. Denio Duarte

CHAPECÓ
2019

Brutti, Nicholas Sangoi

Aprendizado de máquina aplicado à previsão da efetividade de substituições de jogadores no Campeonato Brasileiro de Futebol Série A / Nicholas Sangoi Brutti. – 2019.
69 f.: il.

Orientador: Prof. Dr. Denio Duarte.
Trabalho de conclusão de curso (graduação) – Universidade Federal da Fronteira Sul, curso de Ciência da Computação, Chapecó, SC, 2019.

1. Aprendizado de máquina. 2. Efetividade de substituições. 3. Análise de dados esportivos. 4. Campeonato Brasileiro de Futebol. 5. Futebol. I. Duarte, Prof. Dr. Denio, orientador. II. Universidade Federal da Fronteira Sul. III. Título.

E-mail: nicholassbrutti@gmail.com

NICHOLAS SANGOI BRUTTI

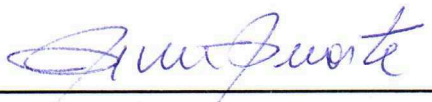
APRENDIZADO DE MÁQUINA APLICADO À PREVISÃO DA EFETIVIDADE DE SUBSTITUIÇÕES DE JOGADORES NO CAMPEONATO BRASILEIRO DE FUTEBOL SÉRIE A

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

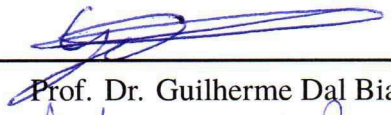
Orientador: Prof. Dr. Denio Duarte

Este trabalho de conclusão de curso foi defendido e aprovado pela banca avaliadora em: 03/07/2019.

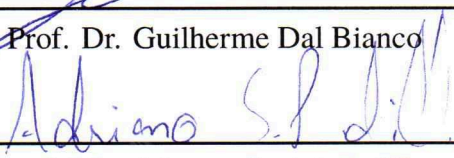
BANCA AVALIADORA



Prof. Dr. Denio Duarte – UFFS



Prof. Dr. Guilherme Dal Bianco



Prof. Me. Adriano Sanick Padilha

RESUMO

As substituições de jogadores são recursos determinantes para o resultado de uma partida de futebol. Dado sua relevância e limitação em partidas oficias, as substituições foram assunto de diversos estudos, com o intuito de fornecer dados de forma estruturada para auxílio na tomada de decisão, por parte das equipes técnicas. Este trabalho, propõe aplicar e comparar algoritmos de aprendizado de máquina, no sentido de classificar a segunda e a terceira substituição do time visitante como efetiva ou não, através da criação de dois modelos distintos. Como conjunto de dados, utilizou-se os dados históricos de cinco anos do Campeonato Brasileiro de Futebol. Os resultados do experimento com 30% dos dados destinados para teste, demonstram que foi possível predizer a efetividade da segunda substituição com 78.39% de acurácia, já a terceira com 86.93%.

Palavras-chave: Aprendizado de máquina. Efetividade de substituições. Análise de dados esportivos. Campeonato Brasileiro de Futebol. Futebol.

ABSTRACT

Substitutions of players are determining resources for the outcome of a football match. Due to the relevance and limitation of substitutions in official matches, several studies have been conducted to propose an optimal way to substitute a player. That is the best moment or the best strategy. This work proposes to apply and compare machine learning algorithms to classify the second and third substitution of the visiting team as effective or not, through the creation of two distinct models. As the input data set, we use data from four years of the Brazilian Soccer Championship (2015-2018). Using 30% of the data set to test the models, the results show that it is possible to predict the effectiveness of the second substitution with 78.39% accuracy and the third with 86.93% accuracy.

Keywords: Machine learning. Substitutition effect. Effectiveness of substitution. Sports data analysis. Brazilian Championship A Series.

LISTA DE ABREVIATURAS

API	Interface de programação de aplicações
CSV	Valores Separados por Vírgula
DEF	Substituição defensiva
DTR	*Decision Tree* (Árvore de decisão)
FN	*False negative* (Falso negativo)
FP	*False positive* (Falso positivo)
HTML	Hipertexto de Marcação de Linguagem
HTTP	Protocolo de Transferência de Hipertexto
JSON	Notação de Objetos do Javascript
KNN	*k-Nearest Neighbors* (k-Vizinhos mais próximos)
OFF	Substituição ofensiva
ORM	Mapeamento objeto-relacional
RFC	*Random Forest* (Floresta aleatória)
SA	Substituição sem alteração evidente
SGBD	Sistema de gerenciamento de banco de dados
SVM	*Support Vector Machine* (Máquinas de vetor de suporte)
TN	*True negative* (Verdadeiro negativo)
TP	*True positive* (Verdadeiro positivo)
XML	Linguagem Extensível de Marcação Genérica

LISTA DE ILUSTRAÇÕES

LISTA DE ALGORITMOS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A análise esportiva é uma área multidisciplinar que busca proporcionar aos técnicos e jogadores informações que contribuam para melhoria contínua do desempenho esportivo. Entre as diversas áreas estudadas, destacam-se a mineração de dados e o aprendizado de máquina que vem ganhando grande notoriedade, impulsionado pelo sucesso no uso em outros esportes, como o basquete e o beisebol (KUMAR, 2013). Com o crescente volume de dados estruturados, semiestruturados e não estruturados, técnicas de mineração são frequentemente adotadas para transformar este extenso volume de dados em informação, contribuindo para o processo de tomada de decisões por parte das comissões técnicas dos clubes (REIN; MEMMERT, 2016). Além disso, a partir destes dados, é possível que seja aplicado técnicas de aprendizado de máquina, para detecção de padrões e previsões.

A Figura 1 apresenta duas reportagens do site da Forbes (www.forbes.com) que discutem a importância do uso da análise de dados para o futebol. Na parte superior da figura, é apresentada manchete de como o Clube Manchester City utiliza os dados para melhorar o desempenho tanto em campo quanto com a torcida. Já, na parte inferior, é apresentada a manchete de como alguns clubes europeus utilizam a análise de dados para contratarem jogadores da categoria de base, ou seja, apostando em jogadores mais jovens para ter um faturamento maior no futuro.

As substituições no futebol são recursos importantes, dado sua limitação em partidas oficiais e a capacidade de mudança tática proporcionada, podendo muitas vezes, influenciar diretamente no resultado final do jogo (MYERS, 2012). Através das substituições, o treinador define explicitamente qual é sua ambição em relação ao jogo, ou seja, se sua equipe será mais ofensiva ou defensiva. Entretanto, é sabido que durante uma partida de futebol as mudanças comportamentais dos times não precisam partir necessariamente da substituição de um jogador. É muito comum no futebol moderno, devido ao equilíbrio das equipes e a forte marcação, a aplicação do conceito de alternâncias táticas.

Figura 1 – Manchetes do site Forbes sobre análise de dados no futebol.

Neste cenário, surgiram diversos estudos utilizando modelos estatísticos e mineração de dados. Um deles tratou sobre a influência das substituições no resultado (GOMEZ; LAGO-PEÑAS; OWEN, 2016), outro avaliou como as variáveis do jogo influenciam no resultado (REY; LAGO-BALLESTEROS; PADRÓN-CABO, 2015) e mais recentemente utilizou-se técnicas de mineração de dados para definição de uma regra de decisão que determina os momentos mais favoráveis para que a substituição ocorra (MYERS, 2012).

Porém, nenhum dos trabalhos selecionados avaliaram a predição da efetividade de substituições, de acordo com o andamento da partida. Este trabalho, portanto explora a utilização de algoritmos supervisionados de aprendizado de máquina, mais precisamente os de classificação, para criação de dois modelos de predição, um para a segunda e outro para a terceira substituição do time visitante, com o objetivo de classificá-las como efetivas ou não. O estudo concentra-se em uma base de dados controlada, referente ao Campeonato Brasileiro de Futebol Série A, do ano de 2015 a 2018. Para determinar os dois modelos mais apropriados para o evento estudado, os algoritmos foram submetidos a testes, e através das métricas de avaliação elegeram-se os dois melhores modelos que obtiveram maior capacidade de predição.

Para o processo de criação dos modelos, foram consideradas *features* do jogo em andamento, como: o tempo das substituições, o saldo de gols no momento da substituição e o tipo tático da substituição, atribuído com base na posição do jogador substituído e do substituto. Além disso, foram utilizadas *features* de trabalhos correlatos, como a força do time (vantagem do time da casa, média diferencial de gols) (SILVA; SWARTZ, 2016). Também novas *features* foram propostas, como: a força defensiva e ofensiva, além da diferença entre as forças. Ambas foram utilizadas no sentido de melhorar a segmentação e valorizar os dados disponíveis.

O trabalho está estruturado da seguinte maneira: o Capítulo 2 apresenta uma breve visão sobre o esporte, e também sobre processo de substituições em comparação aos outros esportes. Aborda também a respeito da competição estudada. O Capítulo 3 discorre sobre o conceito de Aprendizado de Máquina, e sobre os algoritmos de classificação supervisionados. No Capítulo 4 serão apresentados trabalhos relacionados. O Capítulo 5 apresenta a informações sobre como o conjunto de dados foi obtido, quais os atributos criados, entre outros. Em seguida, há o Capítulo 6 que retrata sobre a otimização dos algoritmos e a avaliação do resultado através de métricas. Para finalizar, o Capítulo 7 apresenta a conclusão do estudo desenvolvimento deste trabalho, assim como sugestões de trabalhos futuros.

2 FUTEBOL

O futebol é um esporte amplamente conhecido, sendo considerado o mais popular do mundo (GIULIANOTTI, 2012). Com uma extensa comunidade de espectadores e praticantes, o futebol através de seus grandes torneios é responsável por gerar impactos positivos para economia, promovendo maior desenvolvimento local (ALLMERS; MAENNIG, 2009). Além disso, fornece uma série de benefícios para a saúde dos praticantes (OJA et al., 2015). Um dos motivos que justificam este sucesso, é o fato de o futebol ser um esporte de simples interpretação, o que facilita o entendimento por parte dos espectadores. Outros motivos são explorados em (DUNMORE; MURRAY, 2013).

Uma partida de futebol é composta por dois times. Cada time apresenta no máximo onze jogadores titulares, e no mínimo sete, caso contrário a partida não poderá iniciar. Os times apresentam um banco de reservas com no máximo 12 jogadores suplentes (CBF, 2017). Considerando a situação inicial de um jogo, em cada time há onze jogadores em campo, e um é obrigatoriamente goleiro. A comissão técnica, portanto é responsável por pré-definir estrategicamente o posicionamento dos 10 jogadores restantes, de acordo com o esquema de jogo a ser adotado.

Conforme (DUNMORE; MURRAY, 2013), existem quatro posições básicas:

- Atacante: responsável por propor jogadas ofensivas e finalizações, com o objetivo de marcar gols;
- Defensor: preocupa-se primeiramente em defender seu gol, evitando que o time adversário aproxime-se da área;
- Goleiro: é o jogador mais próximo ao gol de sua equipe. Seu objetivo é evitar que as finalizações dos jogadores adversários se concretizem em gols. É o único capaz de utilizar qualquer parte do corpo para efetuar a defesa, desde que esteja dentro da área delimitada;
- Meio-campista: são os jogadores mais versáteis, contribuem tanto na defesa quanto no ataque.

Para tornar a definição de Meio-campista mais específica de um setor, no meio futebolístico ela é dividida em outras duas posições. São elas, o Volante que é um Meio-campista que atua mais no setor defensivo, e o Meia-atacante cuja sua responsabilidade é atuar mais incisivamente no setor ofensivo.

A posição de cada jogador pode variar de acordo com o andamento do jogo. A partir das circunstâncias da partida e do plantel de jogadores disponíveis, a comissão técnica pode optar por utilizar mudanças táticas (*i.e*, alteração da formação e substituições). Estas modificações tem o intuito de maximizar o desempenho da equipe, seja para reforçar o setor defensivo e manter o placar, ou para priorizar o ataque para ampliar a probabilidade de vitória (DEL CORRAL; BARROS; PRIETO-RODRIGUEZ, 2008).

Uma partida regular de futebol apresenta um total de 90 minutos de jogo. Sendo divididos em dois tempos de 45 minutos. Toda a partida está suscetível a possuir maior tempo de duração, pois em caso de interrupções no jogo, o tempo parado será convertido em acréscimos. A responsabilidade de definir o tempo total de acréscimos da partida é do árbitro previamente escalado.

O objetivo principal do jogo de futebol é que a partir dos chutes em uma bola esférica o time atinja o gol adversário (DUNMORE; MURRAY, 2013), superando o goleiro, bem como, os seus defensores. A cada gol marcado o placar é incrementado em um ponto. Ao final da partida, a equipe que marcou o maior número de gols é decretada vencedora (BRILLINGER, 2010).

Todas as regulamentações, envolvendo o futebol são regulamentadas pela FIFA (Federação Internacional de Futebol[1]), a entidade é responsável por propor leis no âmbito futebolístico (BRILLINGER, 2010).

2.1 CAMPEONATO BRASILEIRO DE FUTEBOL - SÉRIE A

É a principal competição entre clubes de futebol do Brasil, envolve os 20 clubes da elite do futebol nacional. Conhecida popularmente como "Brasileirão", é uma competição organizada pela CBF (Confederação Brasileira de Futebol[2]), que além de possibilitar o título e premiações ao primeiro colocado, permite aos seis primeiros colocados acesso à Taça Libertadores da América. Sendo que os quatro primeiros, têm a vantagem de ingressar diretamente na fase de grupos, enquanto o quinto e sexto colocado participam da fase preliminar (CBF, 2018).

A Libertadores da América é uma competição que envolve os clubes sul-americanos mais bem colocados em seus respectivos campeonatos nacionais, a competição é organizada pela COMNEBOL (Confederação Sul-Americana de Futebol)[3]. Participar desta competição é o desejo de todos os clubes e atletas, dado a visibilidade proporcionada e a chance de participar do Mundial de Clubes, em caso de conquista do título.

Assim como outras em outras ligas nacionais, o Campeonato Brasileiro atualmente segue o modelo de pontos corridos, ou seja, uma competição de longa duração. Neste modelo, ao final do jogo, a equipe vencedora recebe 3 pontos, em caso de empate ambos os clubes recebem 1 ponto, a equipe derrotada não soma pontos. Com um total de 20 equipes participantes, e 38 rodadas disputadas, uma temporada apresenta um total de 380 jogos.

2.2 SUBSTITUIÇÕES

Conforme citado anteriormente, durante uma partida os clubes estão sujeitos a mudanças táticas, seja por iniciativa da equipe técnica ao identificar carências, ou devido a problemas físicos

[1] www.fifa.com
[2] www.cbf.com.br
[3] www.conmebol.com

dos atletas. Uma alternativa utilizada nessas situações, são as substituições. No futebol oficial as substituições têm regras diferenciadas em comparação a outros esportes. Isso porque, há uma limitação no número de substituições. Durante um jogo, são permitidas no máximo três alterações. Além disso, a partir do momento que o jogador foi substituído, ele não poderá reingressar novamente no jogo que encontra-se em andamento.

Estas características reforçam a importância da equipe técnica em acertar suas decisões de alterações de jogadores. Em muitos casos, as substituições são fatores que decidem o jogo, seja de maneira positiva ou negativa (MYERS, 2012), ainda mais por se tratar de um recurso limitado (REY; LAGO-BALLESTEROS; PADRÓN-CABO, 2015).

Nesse contexto, considerando as dificuldades de se estabelecer o momento correto para uma substituição, diversos trabalhos surgiram com objetivo de analisar dados históricos de partidas e determinar uma regra de decisão (MYERS, 2012; SILVA; SWARTZ, 2016). Maiores detalhes serão explanados posteriormente no Capítulo 4.

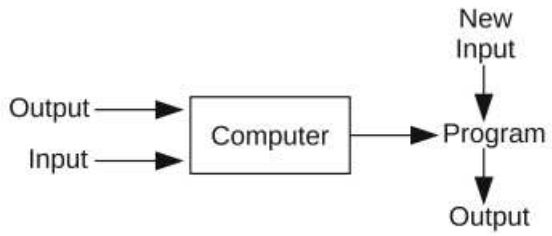
Este trabalho, utiliza conceitos de aprendizado de máquina para prever a efetividade das duas últimas substituições do time visitante. Para confecção do modelo, este trabalho abstrairá as mudanças de esquemas táticos, devido a restrição dessa informação do conjunto de dados de entrada. Portanto, a partir do tempo que as substituições anteriores ocorreram, variáveis circunstâncias do jogo em andamento, considerando a "força" da equipe eo tipo tático da substituição, pretende-se classificar a substituição como efetiva ou não. Para maior entendimento dos conceitos, a seção a seguir apresenta uma revisão sobre o aprendizado de máquina. Os detalhes do projeto do experimento encontram-se descritos no Capítulo 5.

3 APRENDIZADO DE MÁQUINA

O aprendizado de máquina é um ramo da Ciência da Computação que busca proporcionar conhecimento aos computadores, através da aplicação de algoritmos sobre um conjunto de dados previamente conhecido. O foco principal dessa área de pesquisa é permitir que computadores aprendam a reconhecer padrões, para tomarem decisões de forma inteligente com base nos dados (HAN; KAMBER; PEI, 2006).

Conforme a Figura 2, o processo de aprendizagem recebe um conjunto de entradas e saídas, eventualmente as saídas podem ser nulas (DUARTE; STÅHL, 2019). Com esses conjunto de dados aplicam-se técnicas de aprendizado para obtenção de um modelo, que represente o fenômeno em questão. Essa etapa é conhecida como treinamento. Posteriormente, há o chamado teste. Neste caso, o modelo recebe novos dados como entrada, e responde de acordo com o conhecimento obtido no processo anterior. Ao final, métricas são aplicadas para verificação do nível de representatividade do modelo.

Figura 2 – Processo de aprendizado



Fonte: (DUARTE; STÅHL, 2019)

Formalmente, o conjunto de treinamento (X) pode ser representado como $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$, onde m é o tamanho do X (conjunto de treinamento), e $y^{(j)}$ é o rótulo de $x^{(j)}$ $(1 \leq j \leq m)$. Através do conjunto X cria-se o modelo (hipótese). Para verificação da hipótese, um novo conjunto de dados X' tal que $\{X' \cap X = \emptyset\}$ é usado como entrada ao modelo. Como saída tem-se o valor predito ($\hat{y}$), podendo ser de diversos tipos, como: contínuo, discreto (classes), *clusters*, entre outros (DUARTE; STÅHL, 2019).

O aprendizado de máquina é dividido em categorias, cada uma com aplicações em contextos diferentes (HAN; KAMBER; PEI, 2006). Com relação a este trabalho, será aplicado especificamente o aprendizado do tipo supervisionado, afinal, os dados apresentam os rótulos. A seção a seguir apresentará maiores informações sobre esta categoria, assim como os algoritmos que serão utilizados no decorrer do estudo.

3.1 APRENDIZADO SUPERVISIONADO

O aprendizado supervisionado pode ser aplicado em situações onde todo o conjunto de dados de treinamento (X) encontra-se "rotulado", ou seja, para toda a entrada existe uma saída correspondente previamente conhecida. As Figuras 3 e 4 representam o funcionamento em uma situação hipotética.

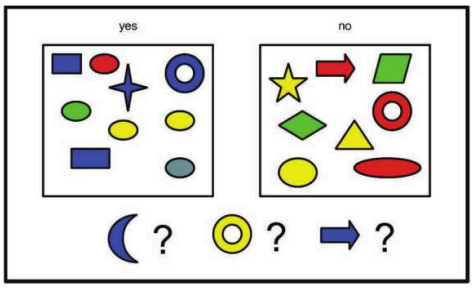


Figura 3 – Representa um conjunto de dados de treinamento rotulado. E um conjunto de teste com 3 elementos inicialmente sem rótulos

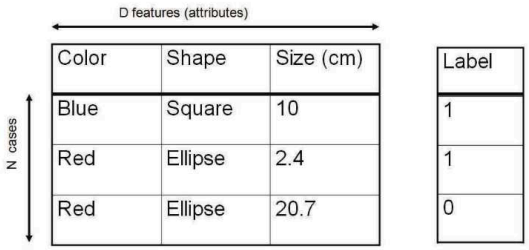


Figura 4 – Conjunto de dados de treinamento visto como uma matriz $N \times D$, com N valores e D *features*. Cada valor com seu respectivo rótulo

Fonte: (MURPHY, 2012)

O aprendizado supervisionado é divido em dois tipos: Regressão e Classificação. A principal diferença está na forma como $\hat{y}$ é calculado. Enquanto a regressão busca ajustar os dados e obter um valor contínuo, a classificação foca em categorizar os dados e separá-los em classes. Um exemplo de classificação binária pode ser observado nas Figuras 3 e 4. Ressalta-se a classificação também pode ser multi-classe.

3.1.1 Classificação

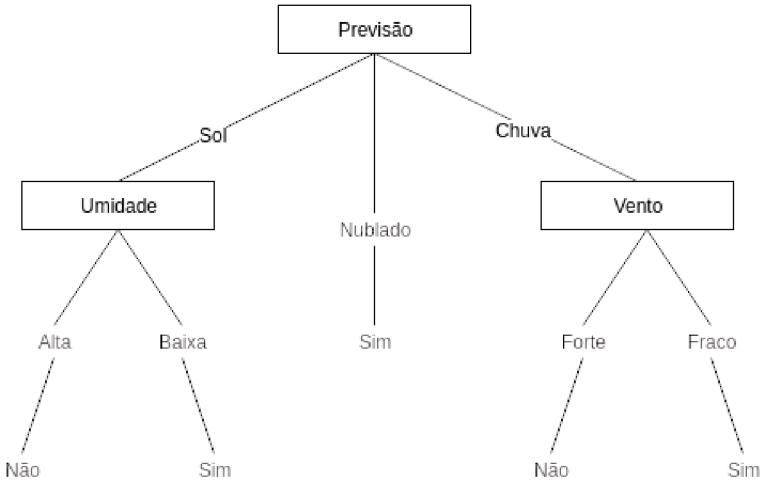
3.1.1.1 Árvore de decisão

A Árvore de Decisão é um dos algoritmos mais populares. Possui aplicações em estudos de diversas áreas devido sua capacidade de representação multidimensional, ausência de configuração prévia e simplicidade para interpretação dos resultados pelos humanos (HAN; KAMBER; PEI, 2006; MITCHELL, 1997). Essas características, tornam o modelo adequado para análises exploratórias.

O próprio nome já sugere como os dados são estruturados. Organizado de forma semelhante a um fluxograma (sem *loops*), os nodos internos representam um teste sobre algum atributo, já os nodos folha uma classificação (resultado). Já o nó que não possui nenhuma aresta incidente é chamado de raiz da árvore.

A classificação ocorre quando se alcança uma folha ao percorrer a árvore a partir da raiz. A cada nó visitado um teste é efetuado para determinar o próximo passo. Diante da resposta obtida a aresta correspondente é explorada, assim sucessivamente até que um nodo folha não seja atingido, encerrando o processo.

Figura 5 – Árvore de decisão para classificação da ocorrência de um jogo de tênis, diante da situação meteorológica



Fonte: Adaptado de (MITCHELL, 1997)

Uma possível entrada para o classificador representado como uma árvore de decisão na Figura 5 seria <*Previsão* = *Sol, Temperatura* = *Quente, Umidade* = *Alta, Vento* = *Forte*>, no caso desta entrada a resposta será negativa. A Árvore de Decisão pode também pode ser vista como uma expressão disjunção de conjunções. Para a mesma árvore de decisão (Figura 5), $(Previsão = Sol \wedge Umidade = Baixa) \vee (Previsão = Nublado) \vee (Previsão = Chuva \wedge Vento = Fraco)$ (MITCHELL, 1997).

Muitos algoritmos surgiram com o intuito de aprender com a Árvore de Decisão, a maioria são variações de um algoritmo de busca gulosa implementado de maneira *top-down*. Entre eles destaca-se o ID3, considerado o mais comum (MARSLAND, 2014; MITCHELL, 1997). O algoritmo ID3 segue a abordagem *top-down*. Primeiramente, busca encontrar o melhor atributo para ser a raiz da árvore. Esse processo envolve testar cada uma das possibilidades, e estatisticamente verificar qual das opções descreve melhor o conjunto de treinamento. A seguir, um descendente do nó raiz é criado para cada um dos valores possíveis. O processo inteiro é então repetido usando os exemplos de treinamento associados a cada nó descendente para selecionar o melhor atributo a ser testado naquele ponto da árvore (MITCHELL, 1997). O método estatístico usado é o ganho de informação (HAN; KAMBER; PEI, 2006).

Para entendimento da escolha do melhor atributo, é preciso introduzir dois novos conceitos a Entropia e o Ganho. Basicamente, a Entropia é um conceito derivado da teoria da informação, que descreve a impureza de uma coleção de dados. No contexto da árvore de decisão, busca-se sempre o atributo com maior Entropia (MITCHELL, 1997; HAN; KAM-

BER; PEI, 2006). Um valor de Entropia igual à zero, não agrega ao conhecimento da árvore, significa que o domínio é homogêneo, pertence totalmente a uma classe. Não há como obter informações extras. Já um atributo que possua Entropia igual a um, entende-se que o conjunto de dados está dividido entre duas classes, esse atributo é extremamente relevante para a árvore e consequentemente fornecerá maior ganho de informação (MARSLAND, 2014).

$$Entropia(p) = -\sum p_i \log_2 p_i, \tag{3.1}$$

O ganho da informação para o atributo A, com relação a um conjunto S de exemplos, segue a Equação 3.2, onde $V(A)$ é um conjunto de todos os possíveis valores para o atributo A, e S_v é o subconjunto de S para qual o atributo A tem valor v (i.e, $S_v = \{s \in S | A(s) = v\}$).

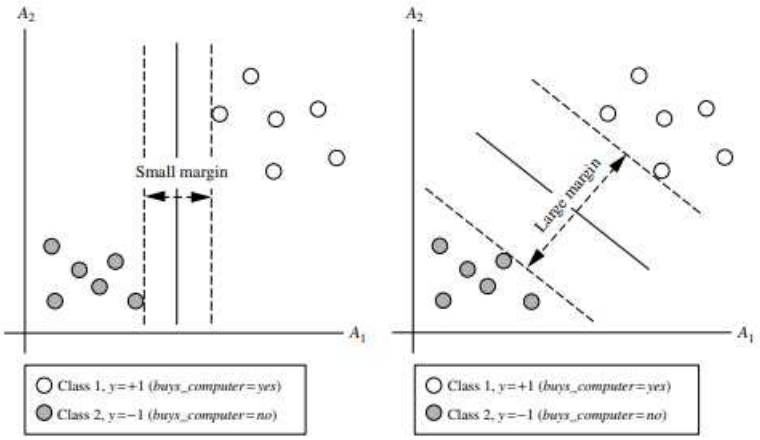
$$Ganho(S, A) = Entropia(S) \; - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropia(S_v) \tag{3.2}$$

3.1.1.2 *Support Vector Machine (SVM)*

O *SVM* é um método de classificação linear e não-linear. O objetivo do método é encontrar um hiperplano ótimo que separe os dados. Para isso, o conjunto de dados de treinamento é transformado para uma dimensão superior (HAN; KAMBER; PEI, 2006). Com uma transformação correta os dados sempre são separáveis. Há dois casos, quando o conjunto é linearmente separável e quando é linearmente inseparável.

A Figura 6 apresenta um caso *linearmente separável*. Observa-se que através da disposição dos dados, que existem vários planos que os separam. Entre estas possibilidades, o algoritmo selecionará o hiperplano que apresenta maior margem (distância) entre os grupos.

Figura 6 – Conjunto de dados linearmente separável, sendo o gráfico da esquerda o hiperplano que apresenta maior margem



Fonte: (HAN; KAMBER; PEI, 2006)

A equação do plano que promove essa separação é descrita pela equação 3.3, o parâmetro W é um vetor de pesos $W = \{w_1, w_2, ..., w_n\}$, já n é o número de *features* e b o *bias*. O conjunto de treinamento X é formado pelas *features*, por exemplo $X = \{x_1, x_2\}$ onde x_1 e x_2 representam as *feature* f_1 e f_2, respectivamente.

$$W \cdot X + b = 0 \tag{3.3}$$

Após o treinamento, para efetuar predições no modelo é possível expressar através da Equação 3.4, como entrada recebe a classe y_i do vetor de suporte X_i, a tupla de teste X^t e l o número de vetores de suporte. As variáveis restantes (α, b_0) são calculadas automaticamente.

$$d(X^t) = \sum_{i=1}^{l} y_i \alpha_i X_i x^t + b_0 \tag{3.4}$$

Já os problemas onde a separação linear é inviável, torna-se necessário a aplicação de uma transformação sobre o conjunto de dados para aumentar a dimensão. Após a transformação, aplica-se o algoritmo para definir o hiperplano no novo espaço (HAN; KAMBER; PEI, 2006). A transformação para o novo espaço envolve a definição do conceito de funções *kernel*. Maiores detalhes do funcionamento do algoritmo são tratados em (HAN; KAMBER; PEI, 2006; MARSLAND, 2014).

3.1.1.3 *K-Nearest Neighbors*

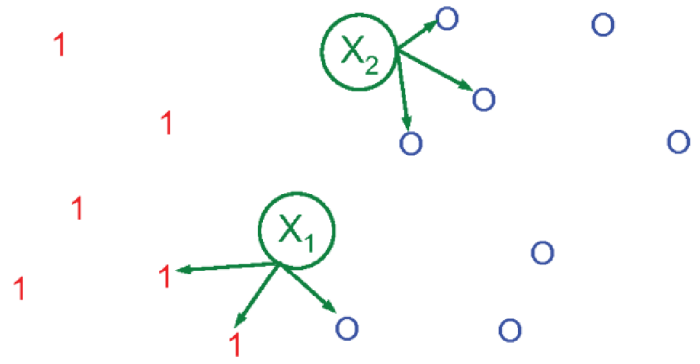
Segundo (HAN; KAMBER; PEI, 2006), o *KNN* é um algoritmo que aprende baseado na analogia dos dados, ou seja, através da similaridade entre a tupla de entrada (teste) e as tuplas de treinamento. Cada tupla formada por n atributos, é representada como um ponto em um espaço n-dimensional. Para determinar em qual a classe que uma tupla do conjunto de teste pertence, é preciso que seja utilizado alguma métrica para encontrar quais são os vizinhos. Neste caso, aplica-se o cálculo da distância, como por exemplo a distância euclidiana. Dado dois pontos ou tuplas , chamados $P_1 = (p_{11}, p_{12}, ..., p_{1n})$ e $P_2 = (p_{21}, p_{22}, ..., p_{2n})$ a distância euclidiana entre estes pontos é calculada conforme na Equação 3.5:

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \tag{3.5}$$

Geralmente, utiliza-se uma técnica de normalização de dados antes de aplicá-los na Equação 3.5. A técnica descrita em (HAN; KAMBER; PEI, 2006) é a normalização *min-max*, que transforma os dados em um valor entre o intervalo $[0, 1]$. Uma vez definida a distância, o algoritmo considera outro parâmetro. Trata-se da variável k que define a quantidade de vizinhos a serem considerados. Para predizer qual a classe da tupla de entrada é verificado qual é classe C mais comum entre seus vizinhos K. Quando $K = 1$, por exemplo, a tupla de entrada recebe a classe do ponto mais próximo no espaço. Já no exemplo exibido na Figura 7 com $K = 3$, as

tuplas de entrada (X_1 e X_2) receberão como rótulo a classe mais comum entre os 3 vizinhos mais próximos. A escolha do parâmetro K não é trivial, caso seja escolhido um K de valor pequeno o método fica sensível a ruídos, e um valor muito grande diminui a precisão (MARSLAND, 2014). O processo para escolha do valor mais adequado é incremental, a cada iteração estima-se o erro associado ao parâmetro, o valor de K que houve menor erro é selecionado. Desta forma aproxima-se do valor mais próximo do ideal. A métrica de distância é crítica, outras abordagens diferentes da distância euclidiana podem ser adotadas para aumentar a performance do modelo (HAN; KAMBER; PEI, 2006).

Figura 7 – Exemplo do funcionamento do algoritmo *KNN* com $K = 3$ em um plano $2D$



Fonte: (MURPHY, 2012)

3.1.1.4 Métodos *Ensemble*

Os métodos *Ensemble* são uma classe específica dentro de algoritmos de classificação. Estes métodos têm a característica de combinar k classificadores, cada um criado a partir de um segmento do conjunto de dados. O rótulo a ser atribuído a uma nova tupla, é definido portanto através de uma votação entre um comitê formado por k classificadores. A classe que recebeu mais votos é adicionada como rótulo da nova tupla. Algoritmos deste tipo tendem a fornecer maior desempenho em comparação aos algoritmos classificadores convencionais (HAN; KAMBER; PEI, 2006; MARSLAND, 2014). A Subseção 3.1.1.4.1 apresenta um algoritmo que pertence a este método, utilizando o conceito de *bagging*.

O *bagging* é o método mais simples de combinação de classificadores. Cria os chamados *bootstraps*, que são amostras diferentes da base de dados com o objetivo de aprender sobre vários aspectos. Portanto, a previsão final para uma nova tupla é definida como a média da previsão de cada uma das hipóteses (MARSLAND, 2014).

3.1.1.4.1 Random Forest

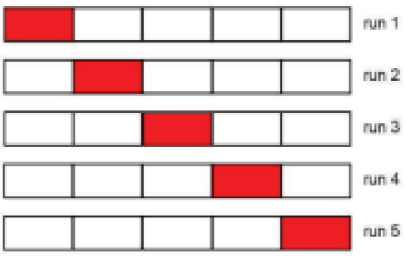
O método *Random Forest* consiste em uma combinação de várias árvores de decisão. É um algoritmo que aplica o método *ensemble* e utiliza o conceito de *bagging*. Algoritmos dessa natureza criam a partir do conjunto de treinamento vários modelos a partir de subconjuntos aleatórios. Os resultados dos vários classificadores são confrontados e se escolhe aquele mais predominante. É uma maneira de garantir maior probabilidade de acerto na decisão, geralmente apresenta um resultado superior em comparação as árvores singulares (HAN; KAMBER; PEI, 2006; MURPHY, 2012).

3.2 VALIDAÇÃO CRUZADA

Um problema que deve ser evitado em aprendizado de máquina é o *overfitting*. Basicamente, significa que a divisão do conjunto de treinamento e teste não ocorreu adequadamente e o algoritmo de aprendizado se especializou somente para aquela parcela do conjunto de dados de treinamento. Em outras palavras, ele não generalizou o evento estudado. Como medida para evitar este tipo de problema existem técnicas de validação cruzada, que fornecem a garantia de generalização através da geração de diversas "fatias"do conjunto de dados. Através de métricas há como definir qual é a divisão mais satisfatória, ou seja, a combinação que garante uma divisão de dados em que o conjunto de treinamento contenha a maior variedade de exemplos para aprender (MARSLAND, 2014).

A Figura 8 apresenta a implementação de um método de validação cruzada, com K = 5, onde cada segmento do conjunto de treino (em vermelho) é testado como conjunto de treinamento.

Figura 8 – Exemplo de validação cruzada



Fonte: (MURPHY, 2012)

3.3 ANÁLISE DE COMPONENTES PRINCIPAIS

Dado um conjunto de treinamento de dimensionalidade d, o *PCA* (*Principal component analysis*) busca por k vetores ortogonais de dimensão n, que melhor podem representar os

dados, onde $k \leqslant n$. O PCA "combina" a essência dos atributos criando um conjunto alternativo de variáveis menor. Os dados iniciais podem então ser projetados neste conjunto limitado. O *PCA* geralmente revela relações que até então não eram visíveis, portanto, permite novas interpretações sobre os dados (HAN; KAMBER; PEI, 2006). Segundo (HARRINGTON, 2012), algumas razões para se reduzir a dimensionalidade dos dados:

- Tornar o conjunto de dados mais fácil de usar;
- Redução do custo computacional de muitos algoritmos;
- Remoção de possíveis ruídos;
- Facilitar a interpretação dos resultados.

Conforme descrito por (HAN; KAMBER; PEI, 2006, p. 102), o procedimento básico do algoritmo *PCA* é o seguinte:

1. Normalização da entrada. Evita que atributos mais frequentes exerçam dominação sobre atributos de ocorrências inferiores;
2. O *PCA* calcula k vetores ortonormais que fornecem uma base para os dados de entrada normalizados. Estes são vetores unitários, que cada um aponta em uma direção perpendicular aos outros. Estes vetores são denominados como componentes principais. Uma característica importante é que os dados de entrada são uma combinação linear entre os componentes principais.
3. Os componentes principais são classificados em ordem decrescente de "significância" ou intensidade. Ou seja, os eixos classificados são de tal ordem que o primeiro eixo mostra a maior variação entre os dados, o segundo eixo mostra a segunda maior variação e assim por diante.
4. O tamanho dos dados é reduzido através da eliminação dos componentes mais fracos, ou seja, aqueles em que há menor variação. Usando os componentes principais mais fortes, deve ser possível reconstruir uma boa aproximação dos dados originais.

3.4 NORMALIZAÇÃO DE DADOS

A normalização dos dados é uma etapa comum na fase de pré-processamento de dados no processo de criação de um modelo de aprendizado de máquina. Em um *dataset* podem haver vários atributos cada um com uma escala específica (e.g. enquanto uma coluna trabalha na ordem de minutos, outra utiliza dias). Nestes casos, aplica-se a normalização de dados, para transformar os dados em um único intervalo de valores (DUARTE; STÅHL, 2019). Onde cada atributo exercerá o mesmo peso (HAN; KAMBER; PEI, 2006). Há diversos algoritmos com

esta finalidade na bibliografia. Entre eles destaca-se o *Min-max* que foi o algoritmo utilizado no Capítulo 5 para transformação dos dados.

O *Min-max*, executa uma transformação linear nos dados originais. Suponha que min_A e max_A sejam os valores mínimo e máximo de um atributo, A. A normalização mapeará um valor v_i em um novo v' no intervalo $[new\_min_A, new\_max_A]$, conforme a Equação 3.6 (HAN; KAMBER; PEI, 2006, p. 114):

$$v' = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A \tag{3.6}$$

3.5 MÉTRICAS DE AVALIAÇÃO

As métricas de avaliação do resultado em algoritmos classificadores, basicamente, são baseadas na contagem da quantidade de classes preditas que são iguais as observadas. Para uma predição p, há quatro possíveis combinações de respostas entre o classificador e a saída esperada, ou seja, ou ambos concordam com a resposta (V e V ou F e F), ou não concordam (V vs. F ou F vs. V). Quando os dois concordam de forma positiva chama-se de verdadeiro positivo (*TP*), e de forma negativa é chamado de verdadeiro negativo (*TF*). Já quando há divergências em que o classificador avaliou como verdadeiro e na realidade o correto é falso chame-se de falso positivo (*FP*), e se o classificador avaliou a entrada como negativa de forma equivocada então há a ocorrência de um falso negativo (*FN*) (DUARTE; STÅHL, 2019).

A definição dos termos *TP*, *TF*, *FP* e *FN* são base para as equações que constituem as métricas de precisão, revocação, *F1-score* e acurácia.

$$precision = \frac{TP}{TP + FP} \tag{3.7}$$

$$recall = \frac{TP}{TP + FN} \tag{3.8}$$

$$F1\text{-}score = \frac{2 \times precision \times recall}{precision + recall} \tag{3.9}$$

A precisão é a relação de exemplos positivos corretos para o número de exemplos positivos reais, enquanto a revocação é a proporção do número de exemplos positivos corretos daqueles que foram classificados como positivos. Uma medida para encontrar um valor adequado para ambos é o *F1-score* (MARSLAND, 2014; DUARTE; STÅHL, 2019). A acurácia, calcula os acertos do classificador conforme a Equação 3.10.

$$\frac{TP + TN}{TP + FP + TN + FN} \tag{3.10}$$

Os quatro algoritmos descritos neste capítulo serão aplicados no contexto deste trabalho. As performances serão comparadas utilizando as métricas descritas na Seção 3.5 no sentido de

encontrar o classificador que melhor descreve o problema, assim como seus hiper-parâmetros de instância. O próximo capítulo, apresenta os trabalhos correlatos.

4 TRABALHOS RELACIONADOS

As substituições são recursos existentes em diversas modalidades esportivas. Porém, cada esporte pode ter uma interpretação diferente sobre sua utilização, conforme definido pela entidade reguladora. O futebol em partidas oficiais, conforme citado anteriormente, apresenta algumas particularidades em comparação aos outros esportes, destaca-se à limitação do número de substituições e a impossibilidade de reingresso de jogadores já substituídos.

Portanto, os trabalhos selecionados tratam exclusivamente do estudo das substituições no contexto do futebol. Não houve distinção quanto as ligas, visto que, as normas são regidas pela FIFA, garantindo que as mesmas regras são válidas independentemente da localização.

(HIROTSU; WRIGHT, 2002) estudaram o problema de definição do melhor momento para substituição. Eles modelaram a partida de futebol como um processo de decisão de *Markov*, contendo quatro estados. Através de programação dinâmica. A partir de dados reais da *Premier League* inglesa o modelo é criado. As probabilidades de transição são estimadas através do método estatístico de máxima verossimilhança. Utilizando programação dinâmica busca-se encontrar o tempo ótimo para uma "mudança tática".

Variáveis são incorporadas ao modelo como: o local da partida (casa ou fora), tempo restante da partida e a formação tática. Uma característica importante deste trabalho é a consideração das chamadas "mudanças táticas", ou seja, ele não trata exclusivamente das substituições de jogadores, mas também de mudanças de aspecto tático. Para promover uma alteração tática não necessariamente precisa acontecer uma substituição. Embora os resultados sejam relevantes, o modelo carece de uma maior validação, pois foram utilizados poucos dados provenientes de partidas hipotéticas.

(DEL CORRAL; BARROS; PRIETO-RODRIGUEZ, 2008) estudaram os aspectos determinantes para realização da primeira substituição. Para tanto, é utilizado um modelo *hazard* Gaussiano inverso, cujo o objetivo é que o modelo estimado identifique as variáveis estatísticas significativas que explicam as substituições dos jogadores durante uma partida de futebol. Foram utilizados os dados da primeira liga espanhola, nas edições de 2004 e 2005. Foram consideradas as seguintes variáveis:

- Variável booleana *HOME* que contém 1 caso a substituição seja do time da casa e 0 caso contrário;
- *CLASSIFICATION* indica a posição do time antes do jogo acontecer;
- *RESULT* consiste na diferença do placar entre a equipe que fez a substituição e a equipe adversária;
- Variáveis booleanas *DEFENSIVE, NEUTRAL* e *OFFENSIVE* que categorizam a substituição;

- *LAST 4 MATCHES POINTS* representa os pontos obtidos nas últimas 4 partidas do time que efetuou a substituição;
- *LAST 4 MATCHES POINTS RIVAL* refere-se aos pontos obtidos nas últimas 4 partidas pela equipe adversária.

Figura 9 – Principais características das primeiras substituições

	Quarter of Hour of Play				
	Halftime	Fourth	Fifth	Sixth	Total
Number of Substitutions	142	194	299	41	676
Home	52%	48%	54%	49%	50%
Last 4 matches points	5.02	5.29	5.15	5.59	5.18
Last 4 matches points rival	4.89	4.95	5.42	5.07	5.17
Classification	9.96	9.90	9.99	8.61	9.85
Opponents' classification	10.57	9.56	10.16	10.05	9.95
Result	−0.57	−0.20	0.35	0.71	0.00
Defensive	12%	12%	21%	24%	16%
Neutral	59%	63%	62%	66%	62%
Offensive	29%	25%	17%	10%	22%

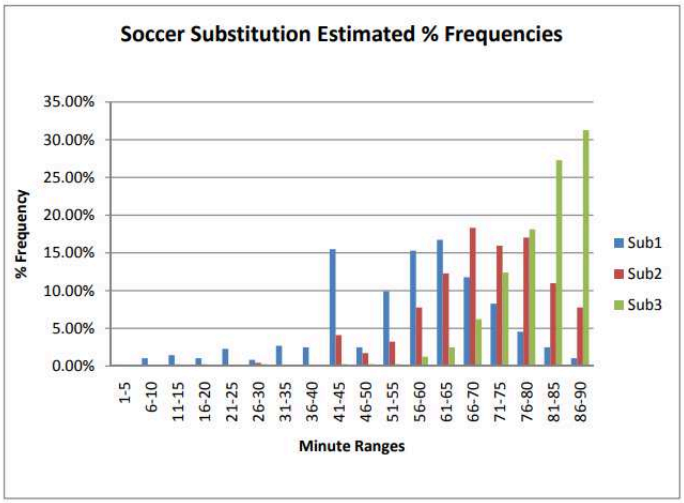
Fonte: (DEL CORRAL; BARROS; PRIETO-RODRIGUEZ, 2008)

Os resultados demonstraram que existem elementos estratégicos importantes que determinam o tempo das substituições. Em particular, foi descoberto que o fator mais importante é a pontuação do jogo no momento da substituição. Também pode-se afirmar que as decisões dos treinadores dependem de sua equipe estar jogando em casa ou fora. Especificamente, as equipes da casa têm uma probabilidade condicional mais alta de fazer sua primeira substituição no intervalo, quando a pressão da torcida é menor.

Já o trabalho de (MYERS, 2012) foca no fator crítico das substituições, o tempo. Historicamente, os treinadores tendem a fazer as substituições de forma mais reativa do que proativa. Esta abordagem nas substituições pode ser equivocada. Se for preciso esperar sinais para identificar que uma substituição precisa ser feita, pode ser que o momento crítico para aquela determinada alteração já tenha passado. (MYERS, 2012) analisou dados históricos de diversas ligas do futebol (*Premier League* inglesa, Série A italiana e *La Liga* espanhola) com o objetivo de propor uma regra de decisão, contendo os melhores intervalos para a ocorrência de uma substituição. Primeiramente ele faz uma análise sobre os dados, para representar a frequência que as substituições ocorrem (veja Figura 10). Além de suas proporções, conforme apresenta na Figura 11.

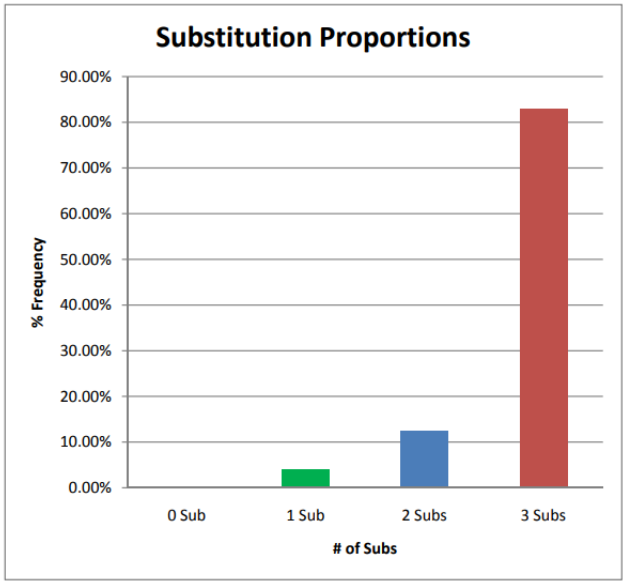
Percebe-se que na Figura 10, a terceira substituição tende a acontecer com maior frequência nos dez minutos finais. Enquanto a segunda acontece com maior frequência na metade do segundo tempo. Já a primeira ocorre em maior quantidade no intervalo de jogo. Na Figura 11 verifica-se que em grande parte dos jogos as três substituições permitidas são utilizadas.

Figura 10 – Frequência de substituições no conjunto de dados de estudo



Fonte: (MYERS, 2012)

Figura 11 – Proporção dos times que usam 0, 1, 2 ou 3 substituições



Fonte: (MYERS, 2012)

Posteriormente o autor verificou se algumas hipóteses são satisfeitas. Com o objetivo de determinar os fatores que afetam os tempos de substituições:

- A primeira hipótese é que o tempo das substituições de uma equipe variam de acordo com o placar da partida (perdendo, ganhando ou empatando). O resultado demonstra que o placar é um fator determinante no tempo de uma substituição. Para todas as três substituições há uma diferença significante;
- Outra hipótese é de que o tempo de substituição é influenciado pelo local da partida (casa ou fora). Para as duas primeiras substituições não há diferenças relevantes, sendo um pouco maior na terceira substituição;
- Como última hipótese o autor aferiu se o tempo de substituição difere de acordo com o tipo da liga. No conjunto estudado, constatou-se que há uma diferença significativa somente na primeira substituição.

Posteriormente, o autor aplica uma árvore de decisão para encontrar o tempo ótimo para substituições que provam maior probabilidade de vitória. Foi utilizada a árvore de decisão devido sua facilidade de interpretação. O modelo foi criado na ferramenta *SAS Enterprise Miner*[1], com um total de 485 observações como conjunto de treinamento. Conforme a combinação dos *splits* a seguinte regra de decisão foi obtida como resultado da mineração de dados:

- Se estiver perdendo:
  - Efetuar a 1ª substituição antes do minuto 58;
  - Efetuar a 2ª substituição antes do minuto 73;
  - Efetuar a 3ª substituição antes do minuto 79.
- A regra não é aplicável nas seguintes condições:
  - Em partidas onde há substituições forçadas no primeiro tempo devido a lesões;
  - Caso algum membro das equipes receba cartão vermelho;
  - Em partidas em que ocorre empate e existe tempo extra (prorrogação) conforme o regulamento.

Os resultados do estudo foram satisfatórios. De acordo com (MYERS, 2012), as equipes que utilizaram esta proposta de substituições, obtiveram uma taxa de sucesso de 38% a 47%, quando comparada com 17% a 24% de quando não é seguida.

O trabalho de (REY; LAGO-BALLESTEROS; PADRÓN-CABO, 2015) trata exclusivamente das substituições na Liga dos Campeões da UEFA. Esta competição possui particularidades em relação aos campeonatos abordados nos trabalhos anteriores. Destaca-se o formato

[1] https://www.sas.com

do torneio, que neste caso segue o modelo popularmente conhecido como "*mata-mata*". Basicamente neste modelo há inicialmente a fase de grupos com um total de 32 times, divididos em 8 grupos. Os grupos são definidos a partir de um sorteio, de tal forma que os clubes integrantes não compartilhem o mesmo país de origem. Durante esta etapa cada equipe joga uma partida em casa e outra fora com cada membro do grupo. A vitória incrementa a pontuação em 3 pontos e o empate em 1 ponto. Após todos os jogos, os dois primeiros colocados de cada grupo se classificam para a próxima fase (2018/19 UEFA CHAMPIONS LEAGUE. . ., 2018).

Nesta fase de fato a competição altera de modelo. Novamente é efetuado um sorteio entre os 16 clubes, originando 8 partidas. Cada partida então ocorre duas vezes, em locais alternados (casa, fora), a equipe que somar maior vantagem no placar agregado se classifica para a fase seguinte. E assim suscetivamente, até chegar na final onde restará apenas 2 times. Nesta situação há apenas uma partida em um local predeterminado. Outro fator diferenciado é a possibilidade de existência de prorrogações.

A prorrogação é típica de partidas deste modelo. Elas são aplicadas em casos em que as partidas permanecem empatadas ao término tempo regulamentar, levando em consideração o gol qualificado. Considerando que o jogo já extrapolou o tempo normal, observa-se que provavelmente haverá substituições nesse período, dado ao alto nível de exigência física de jogos em competições desse molde. Portanto, esses dados são relevantes para construção do modelo mais próximo ao real.

As variáveis situacionais que (REY; LAGO-BALLESTEROS; PADRÓN-CABO, 2015) utilizaram para o estudo foram as seguintes:

- O local da partida (casa, fora). Uma variável é definida em 1 caso a substituição seja promovida pelo time local ou 0 caso contrário;
- O *status* da partida foi considerado, em relação ao número de gols marcados pelo time que efetuou a substituição. A variável apresenta três valores possíveis: *winning*, *drawing*, *losing*;
- A qualidade do time foi calculada a partir da diferença do ranking da temporada anterior, obtido no site oficial da UEFA. Uma abordagem utilizando *k-Means* foi aplicada para classificação e agrupamento dos times em *clusters* de acordo com a qualidade (GOMEZ; LAGO-PEÑAS; OWEN, 2016; REY; LAGO-BALLESTEROS; PADRÓN-CABO, 2015)
- Outra variável foi incluída para representar a fase a qual a partida pertence.

Além disso, duas variáveis dependentes foram consideradas. Primeiramente, o tempo em que a substituição aconteceu junto a um rótulo que representa o número da substituição. A outra variável diz respeito a questão tática da substituição. Dado a posição do jogador substituído e o jogador substituto, é possível classificar a alteração como defensiva, ofensiva ou neutra.

Como conjunto de dados de entrada foram utilizadas 677 substituições, ocorridas durante 124 partidas, nas edições de 2013 e 2014 da Liga dos Campeões da UEFA. Observou-se que

o local da partida não influencia no tempo da substituição, mas sim de acordo com o *status* da partida. Não foi encontrado diferença no tempo de substituição de acordo com a fase da competição ou pela qualidade do adversário. O algoritmo utilizado foi a árvore de decisão J48, contido no *software WEKA*[2].

Como resultado foi obtido uma regra de decisão semelhante ao estudo de (MYERS, 2012). Caso o time esteja perdendo:

- Efetuar a 1ª substituição antes do minuto 53;
- Efetuar a 2ª substituição antes do minuto 71;
- Efetuar a 3ª substituição antes do minuto 80;

O autor finaliza elucidando que embora tenha identificado variáveis que implicam diretamente no tempo das substituições, ainda há outras informações abstraídas que poderiam melhorar a qualidade do estudo, como: condições físicas, qualidade do campo, táticas adversárias e aspectos culturais (REY; LAGO-BALLESTEROS; PADRÓN-CABO, 2015). Orienta-se que os estudos futuros busquem agregar esse conhecimento na etapa de construção do modelo.

O trabalho mais recente na área foi publicado por (SILVA; SWARTZ, 2016). Foi apresentada uma análise alternativa sobre a regra encontrada por (MYERS, 2012). Adotou-se uma abordagem utilizando à Regressão logística bayesiana que é caracterizada por aproveitar o conhecimento a priori. Também, foram utilizados maior quantidade de dados, e um atributo que define a "*força*"do time. O intuito de determinar a força do clube surgiu da hipótese de que um clube com um elenco mais forte, provavelmente possui maiores chances de diminuir a diferença de gols. Portanto, as substituições em times mais fortes em tese pode apresentar maior ganho em qualidade ao time.

Como resultado (SILVA; SWARTZ, 2016) encontraram que com equipes equiparadas há uma vantagem de gols para o time visitante no segundo tempo. Adicionalmente, observaram que não há tempo discernível durante o segundo tempo, quando houve algum benefício através de substituições.

Apesar de existirem diversos trabalhos relacionados à estudos sobre as substituições no futebol em diversas ligas, nenhum deles estudou à previsão da efetividade de substituições, considerando as variáveis circunstanciais do jogo em andamento e os dados históricos. Neste contexto, este trabalho apresenta as etapas de construção de dois modelos de aprendizado de máquina para predição da efetividade de segunda e terceira substituição do time visitante, utilizando como base de dados os dados históricos de partidas do Campeonato Brasileiro de Futebol Série A (2015-2018). Além disso, através do modelo pretende-se identificar as *features* que determinam a efetividade ou não de uma substituição. O próximo capítulo apresentará como o experimento foi projetado, e o que se entende por efetividade no contexto de times visitantes em uma competição de pontos corridos.

[2] https://www.cs.waikato.ac.nz/ml/weka/

5 PROJETO DO EXPERIMENTO

Este capítulo apresenta detalhes sobre todas as etapas que compõem o desenvolvimento de um projeto de aprendizado de máquina. Inicialmente, é descrito como os dados históricos foram obtidos. Em seguida, são apresentados os atributos compostos criados de acordo com os dados disponíveis no conjunto de dados original. Posteriormente, é apresentado o critério para escolha dos algoritmos classificadores e seus hiper-parâmetros.

5.1 EXTRAÇÃO DOS DADOS BRUTOS

Para obtenção dos dados relacionados as partidas de futebol de forma automatizada, houve a necessidade do desenvolvimento de um *scraper*. A extração de dados da *web* (*web scraping*) pode ser definida como o processo de recuperação e combinação de conteúdos de interesse das páginas *web* de maneira sistemática. Para tanto, o programa *(scraper)* é responsável por simular a interação de um usuário convencional com a página, fornecendo a capacidade de localização e extração de informações de quantos sites forem necessários (GLEZ-PEÑA et al., 2013).

Conforme (GLEZ-PEÑA et al., 2013) o conceito de *web scraping* pode ser dividido em três fases distintas:

1. Conexão: o protocolo *HTTP* é utilizado para estabelecer a comunicação entre o *scraper* e o servidor. Através dos verbos *GET* e *POST* o *scraper* interage com o destino (i.e., requisita e envia dados);

2. Extração: utiliza técnicas como o *parsing* do código *HTML* da página e expressões regulares para conseguir identificar as informações relevantes e extraí-las;

3. Geração da saída: este é o principal objetivo dos *scrapers*, transformar os dados extraídos das páginas em uma estrutura de dados organizada (VARGIU; URRU, 2013), que permita a consulta e o armazenamento de forma satisfatória. O formato de saída geralmente é *CSV*, *XML* ou *JSON*.

A extração do conjunto de dados para este trabalho seguiu a abordagem citada acima. Como fonte dos dados, foi utilizado o portal de acompanhamento de jogos em tempo real da UOL[1]. Através da inspeção da requisição efetuada para a página, foi possível identificar que a solicitação dos dados era encaminhada até uma *API* via *XMLHttpRequest*, cuja resposta é um *JSON*. Desta forma, o *scraper* foi desenvolvido para extrair as informações relevantes do *JSON* retornado pela *API*.

Obrigatoriamente, as requisições para a *API* exigem somente três atributos: o nome do time da casa, o nome do time visitante e a data que o jogo aconteceu. Para viabilizar a extração

[1] https://www.uol.com.br

de forma automatizada pelo *scraper* foi obtido uma planilha auxiliar do portal *Football-Data*[2]. Neste arquivo encontram-se disponíveis os atributos necessários para consulta à *API*, além das *odds* (probabilidades) relacionadas as partidas, de acordo com as casas de aposta, porém como não há ligação com este trabalho, estes dados foram desconsiderados. O arquivo apresenta os registros de todas as 1520 partidas ocorridas entre as temporadas 2015 e 2018 da Série A do Campeonato Brasileiro de Futebol.

Desta forma, o *scraper* foi desenvolvido com o objetivo de percorrer esta coleção de jogos, extrair as informações relevantes para o contexto do trabalho e finalmente disponibilizar a saída para que seja dado prosseguimento a próxima fase do processo de aprendizado de máquina.

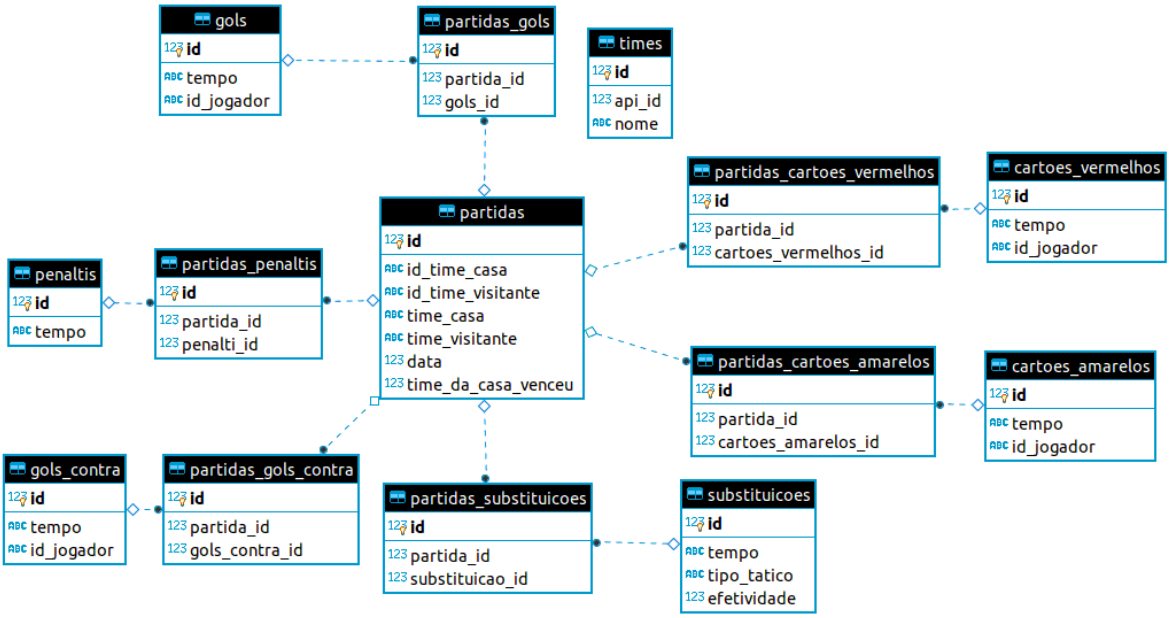
O *scraper* foi implementado na linguagem de programação *Python* na versão 3.7. Para efetuar as requisições *HTTP* e leitura do retorno foram utilizadas as bibliotecas *requests*[3] e *json*, respectivamente. Como saída é gerado um arquivo em formato *JSON*. Paralelamente, é populado um banco de dados relacional embutido (*SQLite*). Para facilitar a compreensão do código e permitir a alteração do SGBD de maneira prática, utilizou-se uma biblioteca auxiliar de *ORM* chamada *peewee*[4].

O modelo entidade relacionamento apresentado na Figura 12 apresenta como foi organizado o banco de dados relacional a partir da extração dos dados pelo *scraper*. Foram extraídos os dados inclusos na chave eventos do arquivo *JSON* recuperado pelo *scraper*. Os eventos são: os gols contra e a favor, cartões vermelhos e amarelos, os pênaltis e as substituições. Para complementar a informação das substituições com o tipo tático, foi necessário extrair o posicionamento do jogador substituído e o substituto, para tanto, foi preciso recuperar estes dados localizados na seção de escalação. Ressalta-se que os atributos do tipo tático e a efetividade não constam no conjunto de dados brutos, detalhes de como eles foram criados estão na Seção 5.2. Estes atributos, são gerados no momento da extração dos dados, por isso já estão inclusos neste diagrama. A tabela partidas armazena o total de 1518 partidas extraídas, os relacionamentos com essa tabela indicam os eventos que vieram a ocorrer em cada partida.

Do total de 1520 partidas realizadas neste intervalo, o *scraper* foi capaz de recuperar 1518, resultando em um arquivo de saída *JSON* de aproximadamente 4.3 *MB*, e um arquivo de extensão *.db* com 1,3 MB. Considerando que houve a ocorrência de uma anulação por *WO*[5], nota-se que apenas os dados de uma partida não foram extraídos com êxito, devido a uma falha na consulta a *API*. Porém, ao observar as substituições, foi identificado que em alguns casos faltava o preenchimento do atributo relacionado ao posicionamento, tanto no jogador substituto quanto o substituído. Embora existam técnicas para imputação de dados, optou-se pela exclusão destas partidas do conjunto de dados. Aplicou-se um filtro e foram selecionadas somente as partidas onde o time visitante realizou as 3 substituições. Desta forma, após

[2] http://www.football-data.co.uk/
[3] http://python-requests.org
[4] http://docs.peewee-orm.com
[5] Partida prevista para 11/12/2016 entre Chapecoense *vs.* Atlético-MG, válida pela 38ª rodada. Não ocorreu em virtude do desastre aéreo que vitimou a delegação da Associação Chapecoense de Futebol.

Figura 12 – Modelo Entidade Relacionamento



estes procedimentos o conjunto ficou com dados de 1326 partidas, conforme pode ser visto na Figura 13. Neste conjunto, há um total de 3978 substituições para análise e utilização nos algoritmos de aprendizado de máquina.

5.2 CRIAÇÃO DE ATRIBUTOS

Após a extração dos dados, iniciou-se a etapa de criação de novos atributos. O objetivo é aproveitar as informações disponibilizadas no *dataset* para gerar novos atributos que contribuirão para a criação do conjunto de treinamento e de teste, além disso, espera-se que eles forneçam maior capacidade de classificação e representatividade ao modelo de aprendizado de máquina. As Equações de 5.1 a 5.4 foram propostas por (SILVA; SWARTZ, 2016), enquanto as Equações de 5.5 a 5.8 foram criadas intuitivamente, de acordo com os dados disponíveis.

5.2.1 Vantagem do time da casa

É um atributo contínuo, calculado com o objetivo de mensurar a força dos times quando jogam em casa em determinada edição do Campeonato Brasileiro de Futebol. Ele não é explicitamente utilizado no conjunto de dados, na realidade, é um indicador auxiliar para o cálculo dos atributos descritos nas eqs. (5.3) e (5.4). O cálculo funciona da seguinte maneira, dado o ano de edição, soma-se todas as ocorrências de gols efetuados pelo time da casa. O mesmo é efetuado para os gols do time visitante. Na sequência efetua-se uma subtração entre os gols do time da casa e os gols do time visitante, e depois há a divisão pelo total de jogos da

edição. Este atributo é definido como *HTA*.

$$HTA = \frac{(total\ de\ gols\ em\ casa\ -\ total\ gols\ fora\ de\ casa)}{total\ de\ partidas} \tag{5.1}$$

5.2.2 Média diferencial de gols

Este atributo é responsável por representar a média diferencial de gols de uma determinada edição do Campeonato Brasileiro de Futebol. O cálculo necessita de duas informações prévias: o time T ao qual será submetido à análise e a edição (E) que deseja-se avaliar. Para montagem da equação, primeiramente é calculado a quantidade de gols que o time T efetuou na edição E. Depois, calcula-se a quantidade de gols sofridos pelo time T na edição E. No final, é subtraído o valor de gols marcados e sofridos, e dividido pelo total de jogos.

$$D = \frac{(total\ de\ gols\ marcados - total\ de\ gols\ sofridos)}{total\ de\ partidas} \tag{5.2}$$

5.2.3 Identificação do provável ganhador

Este atributo é criado baseado nos dois anteriores, descritos nas Equações 5.1 e 5.2. Utilizando os resultados da média diferencial de gols e do *HTA*, é possível criar uma regra que defina o time favorável a vencer, considerando as forças dos dois clubes envolvidos e a qualidade do time da casa em partidas realizadas em seu estádio. Para explicação da regra, suponha que D_j seja o time da casa e D_k o visitante. Se $D_j - D_k + HTA \geqslant 0$, caso positivo, significa que o time da casa é favorável a vencer a partida, logo provavelmente suas substituições serão bem sucedidas. Então a *feature* chamada *RC* recebe 1, e a outra *feature RV* permanece inalterada, em 0. Caso o retorno seja negativo, significa que o time favorável a conquistar a vitória é o time visitante.

$$RC = \begin{cases} 1, & \text{se } D_j - D_k + HTA \geqslant 0 \\ 0, & \text{caso contrário} \end{cases} \tag{5.3}$$

$$RV = \begin{cases} 1, & \text{se } D_j - D_k + HTA < 0 \\ 0, & \text{caso contrário} \end{cases} \tag{5.4}$$

5.2.4 Força defensiva dos times

Como uma medida para determinar a força defensiva dos times participantes de uma partida, foi criada uma nova *feature*. A proposta é que clubes com maior força defensiva,

consigam maior taxa de sucesso em substituições deste tipo. O cálculo foi realizado conforme demonstrado na Equação 5.5:

$$F_{[V,C]D} = \frac{total\ de\ gols\ sofridos}{total\ partidas} \tag{5.5}$$

Através desta fórmula, os clubes com maior força defensiva terão o valor próximo de zero.

5.2.5 Força ofensiva dos time

Semelhante a equação anterior, esta tem o objetivo de indicar a força ofensiva dos clubes. A hipótese é de que clubes com maior força ofensiva, provavelmente terão sucesso ao realizar substituições ofensivas. O cálculo pode ser visualizado na Equação 5.6:

$$F_{[V,C]O} = \frac{total\ de\ gols\ marcados}{total\ partidas} \tag{5.6}$$

Portanto, clubes visitantes que têm maior poderio ofensivo, terão o valor de F_{VO} próximo a 1.

5.2.6 Diferença entre as forças dos times

Foram criados dois atributos para armazenar as diferenças. O primeiro intitulado *DOD* (i.e. diferença ofensiva para defensiva), descrito na Equação 5.7. O intuito desta variável é salvar a diferença entre F_{CD} e F_{VO}. Uma diferença grande pode indicar que o nível de disparidade entre os dois adversários, é significativo. A hipótese é que esta informação também agregue no desenvolvimento e desempenho do trabalho. Já o segundo com características comuns ao anterior, é chamado de *DDO* (i.e. diferença defensiva para ofensiva), ele encontra-se descrito na Equação 5.8. Este valor foi calculado pela subtração de F_{VD} e F_{CO}.

$$DOD = F_{CD} - F_{VO} \tag{5.7}$$

$$DDO = F_{VD} - F_{CO} \tag{5.8}$$

5.2.7 Efetividade da substituição

A efetividade da substituição é um atributo binário que representa o sucesso ou não da substituição. Para o time visitante, a efetividade da substituição está associada a dois fatores: ao tipo tático da substituição e ao modelo de competição adotada. Por padrão, o conjunto de dados não apresenta a informação do tipo tático da substituição, logo foi desenvolvido um método que faz o papel de atribuir um rótulo para a substituição que pode receber um dos três valores:

ofensiva (*OFF*), defensiva (*DEF*) e sem alteração (*SA*). Essa codificação foi baseada no trabalho de (REY; LAGO-BALLESTEROS; PADRÓN-CABO, 2015). Basicamente, há um método intermediário que recebe a posição de um jogador e atribui um índice ascendente de acordo com o grau de ofensividade. As posições foram subdivididas em seis setores, o Algoritmo 1 apresenta como a divisão ocorreu. O algoritmo recebe como parâmetro uma estrutura *s* que contem a posição do jogador a ser substituído e retorna o índice da substituição. O Goleiro é codificado como 0, posições defensivas (Zagueiro, Lateral-direito e Lateral-esquerdo) como 1, Volantes como 2, Meias armadores como 3, Meias-atacantes 4 e Atacantes 5. O valor -1 é o rótulo atribuído quando a informação encontra-se ausente, como foi visto na Seção 5.1 isto aconteceu para 128 substituições, distribuídas entre 113 partidas.

Algoritmo 1 – Atribuição do rótulo contendo o grau de ofensividade da substituição

```
    Entrada: Substituição s
    Saída: índice da substituição
 1  início
 2      selecione s.posição :
 3          caso 'Goleiro' : retorna 0;
 4          caso 'Zagueiro', 'Lateral-esquerdo', 'Lateral-direito' : retorna 1;
 5          caso 'Volante' : retorna 2;
 6          caso 'Meia' : retorna 3;
 7          caso 'Meia-atacante' : retorna 4;
 8          caso 'Atacante' : retorna 5;
 9          senão : retorna -1;
10      fim
11  fim
```

O índice implementado tem como objetivo identificar o tipo da substituição. Se o índice atribuído ao jogador substituto for maior que o do substituído, sabe-se que esta substituição foi do tipo ofensiva (e.g. zagueiro vs. atacante, volante vs. meia-atacante, etc...). Caso contrário, a substituição é considerada defensiva. Já quando o valor do índice é igual, a substituição é classificada como sem alteração (*SA*).

Para a classificação da substituição como efetiva ou não efetiva, conforme demonstrado no Algoritmo 2, é avaliado se no intervalo entre o momento que a substituição ocorreu e o final do jogo houveram gols por parte de qualquer uma das equipes. Independentemente do tipo da substituição, se no intervalo houveram gols do time visitante, a substituição é classificada como positiva, afinal houve um incremento favorável no placar. Já se a substituição foi do tipo *SA* ou *DEF* e o time adversário não marcou gols, a substituição também é positiva. Isto significa que o fortalecimento do setor defensivo ou a alteração para recuperação da condição física surtiram efeito. Nos outros casos, a substituição é classificada como negativa.

Algoritmo 2 – Atribuição do rótulo que define a substituição do time visitante como efetiva ou não efetiva

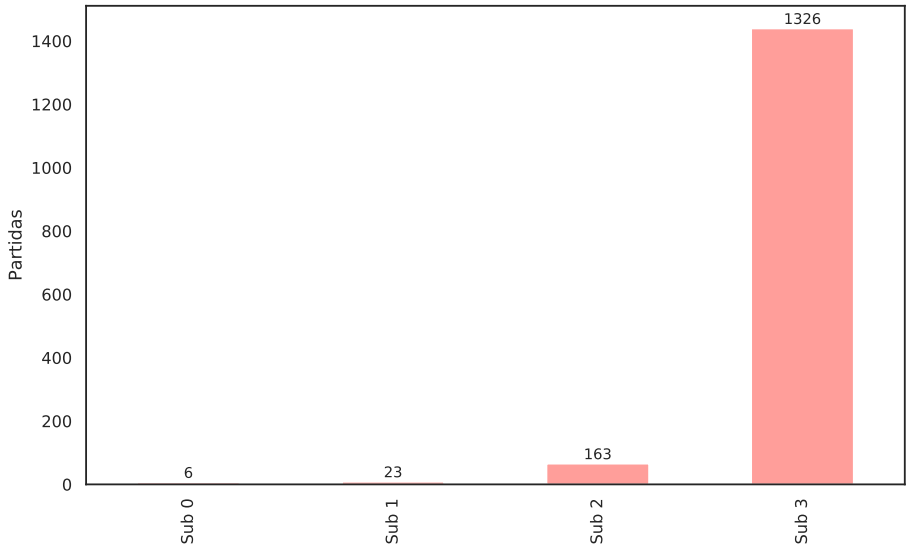
```
   Entrada: Substituição s
   Saída: Retorna 1 caso a substituição s seja efetiva, 0 caso contrário
 1 início
 2 |   gols_favoraveis = procuraGolsNoIntervalo(s.tempo, 90, time_visitante);
 3 |   gols_adversario = procuraGolsNoIntervalo(s.tempo, 90, time_casa);
 4 |   se quantidade(gols_favoraveis) > 0 então
 5 |   |   retorna 1;
 6 |   senão
 7 |   |   se (s.tipo == 'SA' ou s.tipo == 'DEF') e quantidade(gols_adversario) == 0 então
 8 |   |   |   retorna 1;
 9 |   |   senão
10 |   |   |   retorna 0;
11 |   |   fim
12 |   fim
13 fim
```

5.3 ANÁLISE EXPLORATÓRIA DOS DADOS

Esta seção tem como finalidade investigar as substituições promovidas pelo time visitante durante as partidas de futebol ocorridas nas edições de 2015 à 2018 do Campeonato Brasileiro de Futebol, Série A.

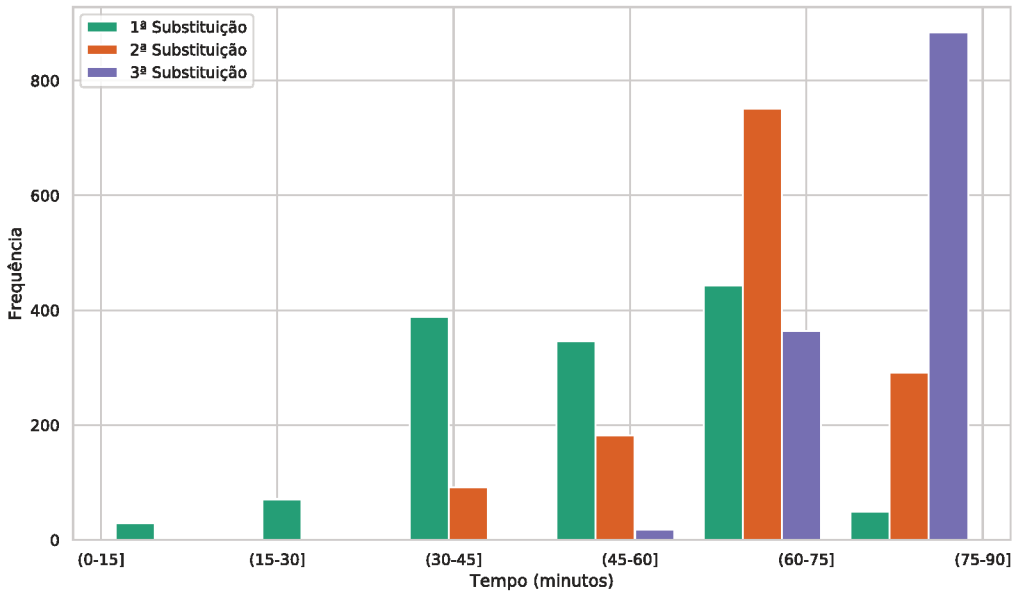
A Figura 13 apresenta que em aproximadamente 87% dos jogos selecionados, a equipe visitante efetua as três substituições. Em contra partida aproximadamente 11% dos jogos realizaram duas substituições, enquanto 1% realizou apenas uma substituição. Um dado significativo, que reforça a importância dada pelos clubes para este recurso. Esta valorização é observada também em estudos envolvendo outras ligas nacionais (MYERS, 2012).

Figura 13 – Proporção das substituições



A Figura 14 apresenta a distribuição das substituições de acordo com os intervalos de tempo, cada um de 15 minutos. Nota-se que o clube visitante, objeto do estudo, tende a realizar a terceira e a segunda substituição nos 30 minutos finas da partida. Já a primeira substituição se encontra distribuída durante toda a partida, havendo quase que uma uniformidade do minuto 30 aos 75. Comportamento semelhante ao encontrado por (GOMEZ; LAGO-PEÑAS; OWEN, 2016) que analisou um conjunto de dados distinto.

Figura 14 – Histograma das substituições



Uma hipótese pela qual há o predomínio das duas últimas substituições na fase final do jogo, é que isto esteja diretamente relacionado ao modelo de competição adotado. Sabe-se que em competições de pontos corridos com o calendário extenso, o empate fora de casa dependendo da qualidade e a tradição do adversário é visto como um resultado aceitável, afinal conquista-se um ponto. Pode-se interpretar que estas substituições ocorrem com um caráter mais defensivo, no sentido de evitar gols do adversário. Mas, há outra hipótese: as substituições também podem ocorrer com a finalidade de tornar o time mais ofensivo. O treinador ao observar que há a chance de conquistar a vitória, prioriza o fortalecimento do setor ofensivo, retirando jogadores fisicamente esgotados e membros do setor defensivo, inserindo no jogo atletas com características mais ofensivas.

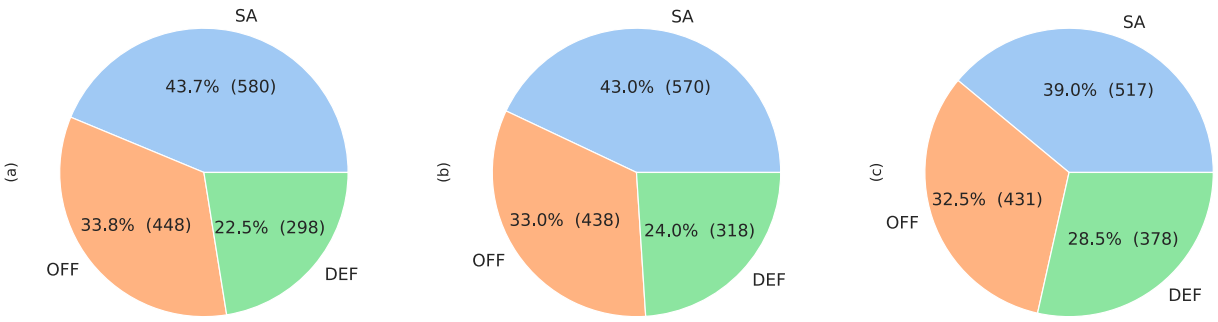
Portanto, para melhor interpretar as circunstâncias em que estas substituições foram efetuadas, é necessário que seja avaliado a alteração tática promovida pelo treinador. Embora, muitas vezes, não se concretize de fato.

A Figura 15 apresenta o tipo tático de cada uma das três substituições possíveis. Percebe-se que a alteração do tipo "SA"ocorre com maior frequência. Alterações deste tipo, teoricamente não modificam a tática da equipe, pois trata-se da substituição de um atleta por outro cuja posição

de origem é a mesma, ou o setor de atuação é o mesmo (e.g. Zagueiro vs. Lateral-direito, ambos atuam no setor defensivo). Uma análise minuciosa deve ser aplicada para verificar a porcentagem destas substituições que foram forçadas devido a lesões. No conjunto de dados obtido, esta informação não consta de forma trivial. Para identificar se uma substituição originou-se de uma lesão, seria necessário avaliar as informações publicadas em tempo real pelos repórteres que acompanham o jogo, se no conteúdo delas há a menção que determinado atleta sofreu uma contusão e que o técnico realizará uma substituição. Como esta análise ultrapassa a delimitação do problema, ela foi desconsiderada no estudo.

A Figura 15 apresenta o tipo tático das substituições efetuadas pelos times visitantes. Esta análise leva em consideração todos os jogos em que houveram as três alterações. A Figura 15 (a) representa as proporções do tipo tático da primeira substituição, a Figura 15 (b) da segunda substituição e a Figura 15 (c) da terceira. Pode-se ressaltar que a substituição do tipo tático defensivo (*DEF*) ocorreu com maior probabilidade na terceira substituição. Aliás, é na terceira substituição que há a informação mais relevante. De acordo com os dados, é nesta substituição que os times priorizaram alterações que visavam tornar o time mais agressivo ou defensivo, abdicando de alterações do tipo *SA*. Este fato, expõe o real interesse do time adversário do minuto T_3 (minuto em que a terceira substituição aconteceu) até o final da partida. A decisão é formulada levando em consideração as circunstancias da partida (i.e. placar, quantidade de jogadores em campo...) e até mesmo o placar dos jogos dos clubes adversários na tabela de classificação (e.g. clubes com chances reais de rebaixamento, dependendo da situação em que se encontram, tendem a priorizar a vitória a qualquer custo).

Figura 15 – Substituições e tipo tático



A Figura 16, representa a distribuição da efetividade das substituições de número dois (a) e três (b), promovidas pelo time visitante. Observa-se que em ambas as substituições, há o predomínio da classe efetiva (1). Embora pareça existir uma relação bicondicional, da forma $p \Leftrightarrow q$, sabe-se que esta afirmação é falsa, basta um contraexemplo para descartá-la. Supondo que a segunda substituição do time visitante foi do tipo *OFF* e esta alteração resultou em um gol favorável, logo ela foi efetiva. Porém, supondo que a terceira substituição seja *DEF*, e o time sofreu um ou mais gols no intervalo entre o minuto da substituição e o final do jogo, esta substituição é definida como não efetiva. Portanto, não há uma garantia que defina que se a segunda substituição foi efetiva a terceira também será. O que pode acontecer, é que dado o

tipo da segunda substituição e da próxima, exista uma probabilidade desta nova substituição ser efetiva ou não.

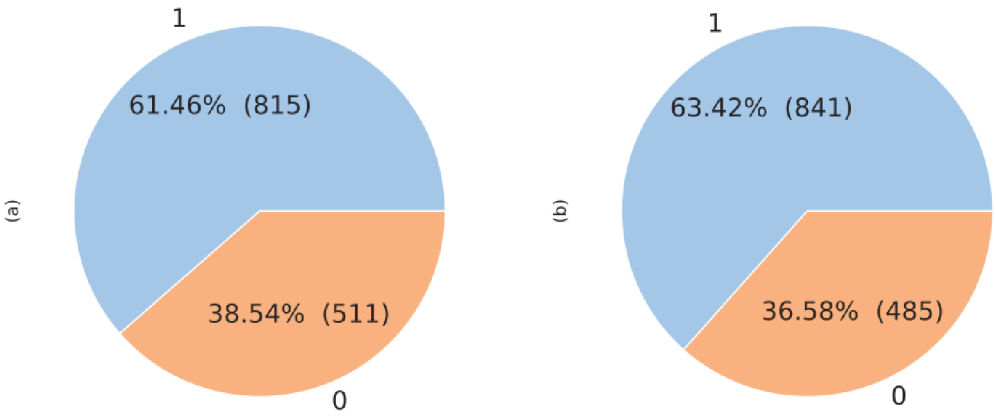


Figura 16 – Distribuição da efetividade das duas últimas substituições permitidas para o time visitante

5.4 ORGANIZAÇÃO DO CONJUNTO DE DADOS E PRÉ-PROCESSAMENTO

Esta etapa foi destinada ao tratamento dos dados extraídos, com o objetivo de organizá-los em uma estrutura de tal forma que seja possível prosseguir para as fases posteriores de treinamento e teste dos algoritmos. Para tanto, foi necessário realizar o pré-processamento de alguns dados, que encontravam-se codificados como variáveis nominais (i.e. o tipo da substituição). Os modelos não trabalham com codificações deste tipo, portanto foi efetuado uma transformação destes dados para valores contínuos. O tipo tático das substituições antes expressado por: *DEF, OFF, SA*, foi convertido em 3 classes (0, 1 e 3). Além disso, foi realizado a discretização das variáveis numéricas que armazenam os tempos em que as substituição ocorreram. O tempo foi dividido em valores categóricos, ou seja, seis classes foram criadas, cada uma responsável por representar um período de 15 minutos do jogo. Para as substituições que ocorreram no intervalo de jogo, foi atribuído o tempo de 45 minutos. A escolha do valor de 15 minutos para o intervalo, justifica-se pelo fato de oferecer maior representatividade do tempo na definição da efetividade ou não de uma substituição, ele é um valor adequado pois um número menor resultaria em um grande volume de classes, o que levaria um aprendizado específico de determinada situação, induzindo ao *overfitting*. Um valor maior para o intervalo, também causaria interferência no aprendizado e a predição seria afetada.

Para representar cada uma das três substituições efetuadas pelo time visitante, existem três tuplas, cada uma delas com quatro atributos: o saldo de gols no momento da substituição (G_n), o tipo da substituição (S_n), o tempo discretizado (T_n) e a efetividade (y_n), com $0 < n <= 3$. O saldo de gols é definido como a diferença entre os gols marcados pelo time visitante e o time

da casa. O tempo é uma informação que por padrão já consta no *dataset* extraído, apenas foi aplicado uma transformação. Já o tipo da substituição e a efetividade foram atributos criados obedecendo os critérios apresentados na Seção 5.2, assim como os atributos de força e a diferença entre elas. Além disso, também há um atributo que contém o saldo de gols no final da partida.

Para conversão do texto categórico contido no atributo tipo da substituição em dados numéricos compreensíveis para o modelo de aprendizado de máquina, utilizou-se a classe *LabelEncoder* contida na biblioteca *scikit-learn*. Para a conversão do tempo optou-se por resolver nativamente pela linguagem de programação. Desta forma, o conjunto de dados principal foi formulado com os seguintes atributos:

- A coluna G_n representa o saldo de gols antes da substituição n ser efetuada;
- A coluna S_n representa o tipo tático da substituição n, codificado da seguinte maneira: 0 para substituições defensivas, 1 para ofensivas e 2 para as que não promovem alteração tática;
- A coluna T_n é composta pela classe relativa ao tempo em que a substituição n aconteceu;
- O atributo F_{VD}, representa a força defensiva do time visitante;
- O atributo F_{VO}, representa a força ofensiva do time visitante;
- O atributo F_{CD}, representa a força defensiva do time da casa;
- O atributo F_{CO}, representa a força ofensiva do time da casa;
- Os atributos R_C e R_V são dados binários, responsáveis por informar se o time da casa é favorável a vencer ou o visitante, respectivamente;
- Já os atributos DOD e DDO representam a diferença entre as forças dos dois clubes, ataque$_{[VO,CO]}$ *vs.* defesa$_{[VD,CD]}$;
- As variáveis y_n apresentam a informação se a substituição n foi efetiva (1) ou não (0).

As seções 5.4.1 e 5.4.2, apresentam como este conjunto foi segmentado para a criação dos dois modelos distintos, o primeiro sendo responsável por predizer se a 2ª substituição será efetiva. E um segundo modelo, que irá predizer a efetividade ou não da 3 substituição.

5.4.1 Estrutura dos dados do modelo I

O primeiro modelo tem a finalidade de prever a efetividade da 2ª substituição da equipe visitante. Neste caso, o rótulo a ser predito é o y_2. A ideia é que através das informações previamente conhecidas referentes a primeira substituição e o saldo de gols atual G_2, o modelo classifique a substituição candidata S_2 que ocorrerá no tempo T_2, como efetiva ou não efetiva.

A Tabela 1 apresenta um extrato de como o *dataset* para este modelo ficou organizado. Os atributos retidos para a criação do modelo foram, portanto, o saldo de gols antes da primeira substituição G_1, o tipo tático da substituição S_1, a classe referente ao tempo em que a primeira substituição ocorreu T_1, o mesmo é feito para esses atributos referentes à segunda substituição, alterando apenas o índice para dois.

Além disso, há os outros atributos, que envolvem dados históricos do retrospecto dos dois clubes participantes do jogo. Trata-se da força defensiva e ofensiva do time visitante, representado respectivamente por F_{VD} e F_{VO}. Há também os atributos F_{CD} e F_{CO} que têm a mesma finalidade, porém relacionado ao time da casa. Os atributos R_C, R_V, DOD e DDO completam o conjunto de dados.

Tabela 1 – *Dataset* estruturado para o modelo I

G_1	G_2	S_1	S_2	T_1	T_2	F_{VD}	F_{VO}	F_{CD}	F_{CO}	R_C	R_V	DOD	DDO	y_2
0	0	2	2	2	2	0.55	0.51	0.55	0.41	0	1	-0.04	0.14	1
0	0	2	1	2	2	0.72	0.43	0.72	0.44	1	0	-0.29	0.28	1
0	0	2	1	2	2	0.70	0.38	0.70	0.43	1	0	-0.32	0.27	0

5.4.2 Estrutura dos dados do modelo II

O segundo modelo tem a finalidade de prever a efetividade da 3ª substituição da equipe visitante. Neste caso, o rótulo a ser predito é o y_3. A ideia é que através das informações previamente conhecidas referentes a primeira e a segunda substituição, o modelo classifique a substituição candidata S_3 que ocorrerá no tempo T_3, como efetiva ou não efetiva. A Tabela 2 demonstra como o *dataset* é formado. Neste caso, os atributos retidos para construção do modelo foram praticamente os mesmos da Tabela 1, com a alteração do rótulo a ser predito e com o acréscimo da tupla com a informação sobre a terceira substituição (G_3, S_3 e T_3).

Tabela 2 – *Dataset* estruturado para o modelo II

G_1	G_2	G_3	S_1	S_2	S_3	T_1	T_2	T_3	F_{VD}	F_{VO}	F_{CD}	F_{CO}	R_C	R_V	DOD	DDO	y_3
0	0	0	2	2	2	2	2	3	0.55	0.51	0.55	0.41	0	1	-0.04	0.14	1
1	1	1	2	2	2	1	3	3	0.72	0.43	0.72	0.44	1	0	-0.29	0.28	0
0	0	0	2	1	0	2	2	3	0.70	0.38	0.70	0.43	1	0	-0.32	0.27	1

5.4.3 *Feature selection*

Segundo (KOTSIANTIS; KANELLOPOULOS; PINTELAS, 2006), a etapa de *feature selection* é responsável por identificar e remover *features* consideradas irrelevantes ou redundantes. Esta medida permite que os modelos diminuam de complexidade, tornando-os mais rápidos e efetivos na predição. Através da redução de ruído nos dados, evita-se o *overfitting*.

Neste trabalho, a técnica utilizada para selecionar as *features* mais relevantes encontra-se disponível na biblioteca *scikit-learn*, trata-se do método *SelectKBest*. Este método, recebe dois parâmetros. O primeiro é a função *score*, e em segundo uma variável *k*, que define a quantidade de principais atributos que deseja-se selecionar. Em ambos os modelos, a função *score* utilizada foi a *f_classif*. Para o modelo I o atributo *k* foi fixado em 6. Já para o modelo II foram consideradas todas as *features*.

Para estipular o valor possível de *k*, foi realizado uma análise sobre a correlação entre cada uma das variáveis, em cada um dos conjuntos de dados. Foram elaborados dois mapas de calor, conforme pode ser visualizado nos Apêndices B e C. Analisando o Apêndice B, basicamente, as seis *features* mais influentes para o modelo I são as que possuem maior valor na linha respectiva a variável a ser predita (y), nestes casos há uma correlação positiva. O retorno obtido pelo método *SelectKBest* coincide com esta análise. Além disso, foram gerados diversos modelos com diferentes valores de *k*. Conforme pode ser visualizada na Tabela 4 o valor de *k* que obteve maior média de acertos foi 6. Portanto para o modelo I foram consideradas as *features* descritas na Tabela 3.

Tabela 3 – Seis *features* com melhor colocação no modelo I

Feature	*Score*
FVO	16.73
RV	14.62
RC	14.62
DOD	14.51
G1	10.06
S2	8.12

Tabela 4 – Teste de predição do modelo I com *k features*

Total de features	*Score médio*
2	0.5912
3	0.5940
4	0.5921
5	0.5765
6	0.7291
7	0.7251
8	0.6717
9	0.6709
10	0.6686
11	0.6694
12	0.6388
13	0.6398
14	0.6338

Para o modelo II, optou-se por manter todas as *features*. Pois, esta combinação obteve nos testes de validação cruzada um valor médio de *score* superior, comparado a outras execuções com o valores menores para o parâmetro *k*, conforme exibido na Tabela 5. A princípio, todas as variáveis tem relevância para classificação do evento estudado. Isto será melhor explorado após a execução e otimização, onde através do método de floresta aleatória ou árvore de decisão, será verificado o grau de importância dado a cada variável para o processo de classificação.

Tabela 5 – Teste de predição do modelo II com *k features*

Total de features	*Score médio*
2	0.6037
3	0.5969
4	0.5973
5	0.5846
6	0.5690
7	0.5755
8	0.5793
9	0.5672
10	0.5650
11	0.5685
12	0.5685
13	0.5698
14	0.5673
15	0.5685
16	0.6932
17	0.6956

O próximo capítulo tratará da fase de execução (treinamento, otimização) dos dois modelos classificadores. Para desenvolvimento desta etapa foi utilizado a linguagem de programação *Python* com as bibliotecas *scikit-learn* e *pandas*. A *scikit-learn* biblioteca é amplamente conhecida e fornece a maioria dos algoritmos de aprendizado de máquina já implementados e testados, além de possuir uma excelente documentação e outras funções auxiliares que contribuem para simplificar e padronizar o desenvolvimento de aplicações desta finalidade. Já a biblioteca *pandas* é de grande utilidade para manuseio dos dados, seja na leitura de arquivos ou operações mais avançadas, a classe *DataFrame* inclui muitas funcionalidades.

6 EXECUÇÃO E RESULTADOS

6.0.1 Treinamento

Para o treinamento de ambos os modelos, foi utilizado 70% do conjunto de dados, os outros 30% restantes foram destinados para o teste. Entre as diversas técnicas existentes para esta finalidade, a técnica usada para segmentação dos conjuntos de treino e teste foi a estratificação aleatória (*StratifiedShuffleSplit*). Como medida para evitar o *overfitting*, esta técnica possibilita que o modelo treine com uma maior variedade de exemplos, garantindo o balanceamento de classes. Esta função no *scikit-learn* recebe quatro parâmetros, sua explicação e os valores adotados para este experimento encontram-se descritos abaixo.

- *n_folds*: refere-se a quantidade de "partes"que o conjunto será dividido entre treino e teste. O valor definido para este parâmetro foi 10;
- *test_size*: valor real na escala de 0 até 1, que representa a porcentagem dos dados destinados ao teste. O valor para este parâmetro é 0.3;
- *train_size*: valor real na escala de 0 até 1, que representa a porcentagem dos dados destinados ao treinamento. O valor para este parâmetro é 0.7;
- *random_state*: é a chamada semente utilizada para geração de números pseudo-randômicos que são responsáveis por embaralhar os dados. Este atributo foi fixado no valor 42, para garantir que os dados não se alterem durante a etapa de construção do modelo.

Em seguida, foram escolhidos os quatro algoritmos classificadores para criação do modelo, tratam-se do *Random Forest Classifier*, *Support Vector Machine*, *K-Nearest Neighbors* e *Decision Tree*.

Inicialmente, com o objetivo de avaliar e selecionar os algoritmos que melhor se adéquam para solução do problema, foi realizado uma etapa de comparação. Por enquanto, nesta fase, cada classificador foi instanciado com os hiper-parâmetros padrões dos algoritmos. A comparação ocorreu através de um laço de repetição entre os quatro modelos, onde foi calculado um índice chamado *score* para cada uma das 10 partes do conjunto de treino. Exclusivamente, para o classificador *KNN* o conjunto de dados foi normalizado através do algoritmo *Min-max*. A métrica utilizada para o cálculo do *score* foi a acurácia. Ao final de cada iteração do laço, calcula-se a média do *score* e o desvio padrão para o modelo em questão.

Os gráficos do tipo *box plot* contidos nas Figuras 17 e 18 representam justamente de forma gráfica a média do *score* atingindo por cada classificador. A sigla *RFC* refere-se ao *Random Forest Classifier*, *DTR* a *Decision Tree Classifier*, SVM ao *Support Vector Machine*, o KNN ao *K-Nearest Neighbors Classifier*. A partir das figuras, constata-se que os três algoritmos que se sobressaíram e obtiveram melhores resultados da média para o modelo I foram: *RFC*, *SVM* e *KNN*. Enquanto para o modelo II os melhores algoritmos foram: *RFC*, *SVM* e *DTR*.

Tabela 6 – Tabela de resultados de execução dos modelos com *cross-validation*

Classificador	Modelo I		Modelo II	
	Média	Desvio padrão	Média	Desvio padrão
RFC	0.707789	0.017409	0.831910	0.017853
DTR	0.683166	0.019606	0.780653	0.011841
SVM	0.783166	0.016051	0.797990	0.010789
KNN	0.742211	0.017631	0.768342	0.014856

Figura 17 – Comparação dos classificadores no modelo I

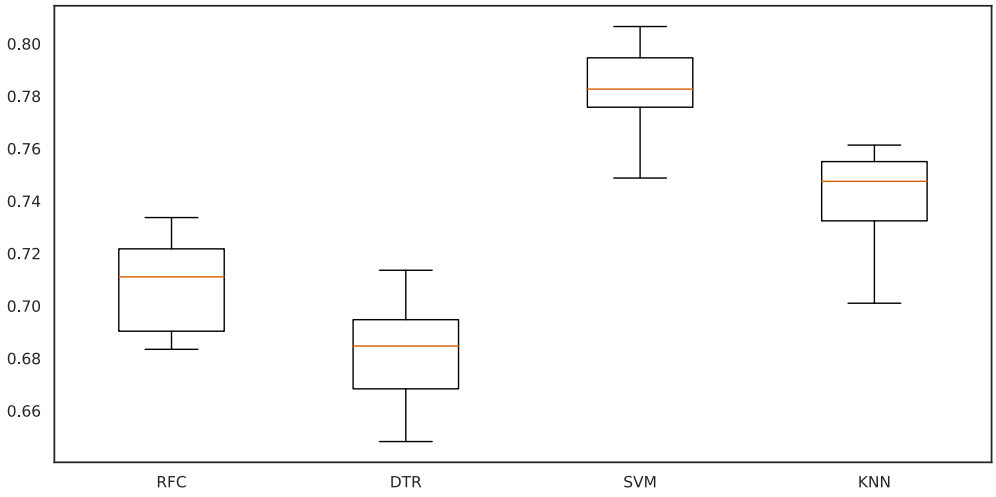
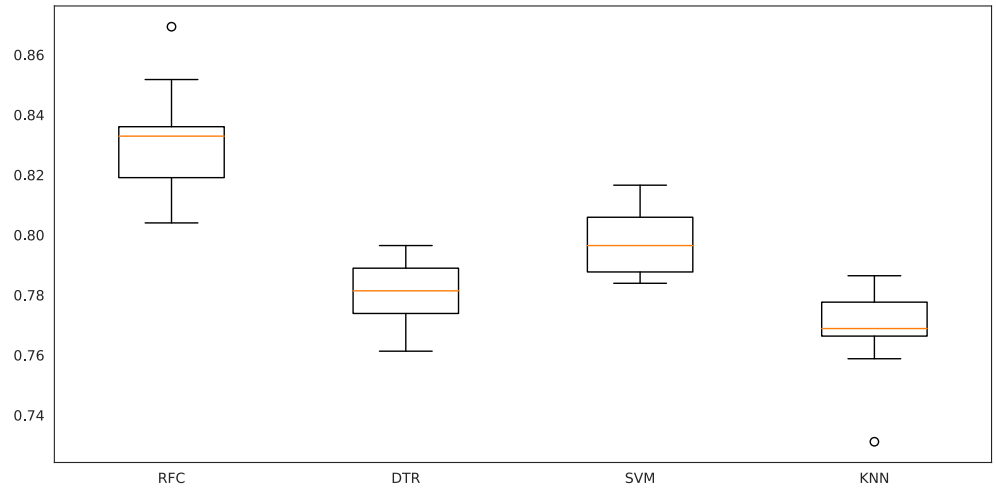


Figura 18 – Comparação dos classificadores no modelo II



6.0.2 Otimização

Cada classificador possui determinados hiper-parâmetros no momento em que é instanciado. Um grande desafio é encontrar os valores mais adequados para estes hiper-parâmetros de tal forma que crie o melhor modelo possível, de acordo com o conjunto de dados de treino disponível. Neste sentido, a biblioteca *scikit-learn* fornece o *GridSearchCV*, que implementa uma busca do tipo força-bruta. Este método recebe como parâmetro diversos dados, entre eles destacam-se: o *param_grid* que armazena o dicionário de dados com as chaves e os valores possíveis, e também o parâmetro *cv* que define a estratégia de validação cruzada a ser adotada. De posse deste dicionário de dados com o nome dos atributos e os valores que se deseja testar, através da busca e validação cruzada, o algoritmo retorna os hiper-parâmetros do melhor modelo produzido com os dados informados.

A função *GridSearchCV*, foi utilizada neste estudo para encontrar os melhores hiper-parâmetros para os três classificadores selecionados de cada modelo. Para cada classificador, criou-se um dicionário de dados contendo o nome do hiper-parâmetro e uma lista de possíveis valores. O Apêndice A descreve os hiper-parâmetros utilizados, bem como uma breve descrição do que cada um significa. Enquanto as Tabelas 7 e 8 apresentam os valores que foram testados e o resultado obtido para cada um dos modelos.

Tabela 7 – Especificação dos hiper-parâmetros aplicados ao *GridSearchCV* e o retorno obtido para o modelo I

Classificadores	Hiper-parâmetros	*Melhores parâmetros*
RFC	*'n_estimators': [200, 500, 1000], 'max_features': ['auto', 'sqrt', 'log2'], 'max_depth' : [4,5,6,7,8], 'criterion' :['gini', 'entropy'],*	*'criterion': 'entropy', 'max_depth': 5, 'max_features': 'sqrt', 'n_estimators': 1000*
SVM	*'kernel': ['linear', 'rbf'], 'C': [1, 0.25, 0.5, 0.75], 'gamma': [1, 2, 3, 'auto'], 'decision_function_shape': ['ovo', 'ovr'], 'shrinking': [True, False]*	*'C': 1, 'decision_function_shape': 'ovo', 'gamma': 'auto', 'kernel': 'rbf', 'shrinking': True*
KNN	*'n_neighbors': [3,5,11,19], 'weights':['uniform', 'distance'], 'metric':['euclidean', 'manhattan']*	*'metric': 'manhattan', 'n_neighbors': 19, 'weights': 'uniform'*

Tabela 8 – Especificação dos hiper-parâmetros aplicados ao *GridSearchCV* e o retorno obtido para o modelo II

Classificadores	Hiper-parâmetros	*Melhores parâmetros*
RFC	*'n_estimators': [200, 500, 1000], 'max_features': ['auto', 'sqrt', 'log2'], 'max_depth' : [4,5,6,7,8], 'criterion' :['gini', 'entropy'],*	*'criterion': 'entropy', 'max_depth': 6, 'max_features': 'sqrt', 'n_estimators': 1000*
SVM	*'kernel': ['linear', 'rbf'], 'C': [1, 0.25, 0.5, 0.75], 'gamma': [1, 2, 3, 'auto'], 'decision_function_shape': ['ovo', 'ovr'], 'shrinking': [True, False]*	*'C': 1, 'decision_function_shape': 'ovo', 'gamma': 1, 'kernel': 'rbf', 'shrinking': True*
DTR	*'max_depth': [3, None], 'min_samples_leaf': [1, 5, 10], 'criterion': ['gini', 'entropy']*	*'criterion': 'gini', 'max_depth': 3, 'min_samples_leaf': 5*

6.0.3 Resultados

Para avaliação dos modelos I e II foram utilizadas as métricas de acurácia, precisão, revocação e *F1-score*, descritas na Seção 3.5. A validação ocorreu exclusivamente com 30% do conjunto de dados, o que significa que há dados de 398 jogos. Estes dados compõem o conjunto de teste, são independentes e não participaram da fase de treinamento do modelo. Esta medida é importante para não tornar o aprendizado tendencioso, afinal, não faz sentido testar com exemplos que foram ensinados ao algoritmo, visto que ele foi construído para acertar estes exemplos. Não foi preciso implementar estas métricas pois a biblioteca *scikit-learn* já possui estas funcionalidades disponíveis no pacote *metrics*. Optou-se pela utilização da função *classification_report* que concentra todos estes indicadores em apenas uma chamada.

O resultado de cada métrica em cada um dos classificadores pode ser visualizado nas Tabelas 9 e 10. O método de segmentação por estratificação realizou a criação do conjunto de teste automaticamente. Para o modelo I, o conjunto é composto por 153 tuplas da classe 0 (substituições não efetivas) e 245 tuplas da classe 1 (substituições efetivas). Já para o modelo II há uma pequena diferença, o conjunto é composto por 146 tuplas da classe 0 (substituições não efetivas) e 252 tuplas da classe 1 (substituições efetivas).

Tabela 9 – Tabela de resultados de execução do modelo I após o *tuning* dos hiper-parâmetros

Classificador	Precisão (Avg)	Revocação (Avg)	*F1-Score* (Avg)	Acurácia
SVM	0.78	0.76	0.76	78,39%
KNN	0.78	0.75	0.76	78,14%
RFC	0.78	0.75	0.75	77,89%

Tabela 10 – Tabela de resultados de execução do modelo II após o *tuning* dos hiper-parâmetros

Classificador	Precisão (Avg)	Revocação (Avg)	*F1-Score* (Avg)	Acurácia
DTR	0.88	0.84	0.85	86.93%
RFC	0.87	0.83	0.85	86.43%
SVM	0.81	0.72	0.74	78.39%

6.0.4 Discussão dos resultados

Considerando o valor da acurácia, percebe-se na Tabela 9 que o algoritmo *SVM*, foi o melhor colocado para o modelo I, ele alcançou uma taxa de revocação minimamente superior aos outros algoritmos. E no modelo II, o algoritmo *RFC* foi quem obteve os melhores valores.

Ao analisar as métricas por classe do algoritmo melhor colocado no modelo I, nota-se que ele foi mais eficiente na classificação de dados pertencentes à classe 1. Havendo uma taxa de precisão de 80%, revocação de 87% e o *F1-score* de 83%. Valores mais expressivos do que na predição de tuplas da classe 0, onde se obteve 76% de precisão, 64% de revocação e 70% de *F1-score*. Isto se deve ao fato de que no conjunto de dados há o predomínio de dados pertencentes à classe 1. Em cerca de 61,46% (815 tuplas) do conjunto de dados a substituição 2 foi classificada como efetiva (1), enquanto 38,54% (511 tuplas) ela foi negativa. No conjunto de teste aplicado ao modelo, haviam 398 tuplas. Sendo 153 delas pertencentes à classe 0, e as outras 245 tuplas restantes à classe 1.

No modelo II ao consultar as métricas por classe do melhor algoritmo encontrado observa-se o mesmo cenário, exceto na métrica de precisão que na classe 0 foi superior a classe 1. Para a classe 0 foi encontrado 89% de precisão, 73% de revocação e 80% de *F1-score*. Referente a classe 1 a precisão encontrada foi de 86%, de revocação 95% e de *F1-score* 90%. Das 398 tuplas utilizadas como conjunto de teste, 146 delas pertenciam à classe 0, já as outras 252 tuplas à classe 1.

Embora o classificador *SVM* tenha obtido melhores resultados em ambas as métricas, a extração das *features* mais importantes torna-se difícil, pois, foi utilizado o *kernel* diferente do linear. Neste caso, não é possível extrair as *features* mais importantes nativamente na biblioteca na *scikit-learn*. Portanto, utilizou-se a *Random Forest* para estudo dos fatores de influência no sucesso ou não das substituições.

Analisando a importância das *features* atribuídas pelo classificador *Random Forest* no modelo I, disponível na Figura 19, é possível perceber que a *feature* que obteve maior relevância na classificação é o tipo da substituição candidata (S_2), para predição da efetividade da segunda substituição. Em seguida, é a força ofensiva do visitante (*FVD*) e a diferença ofensiva defensiva (*DOD*).

O mesmo é observado no modelo II para a terceira substituição, conforme a Figura 20. O tipo da substituição S_3 é determinante para classificar S_3 como efetiva ou não, seguido pela diferença defensiva ofensiva (*DDO*) e a força ofensiva (*FCO*) do time da casa.

Figura 19 – Classificador *RandomForest* modelo I

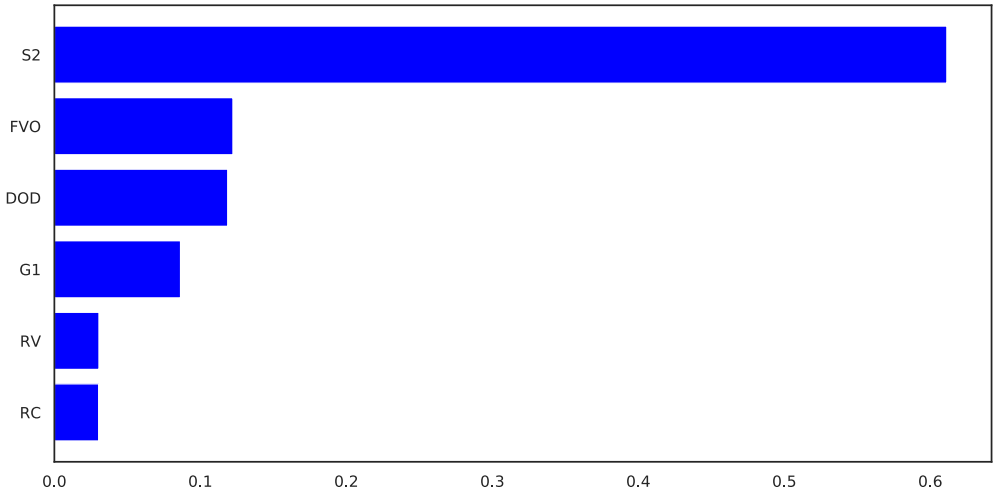
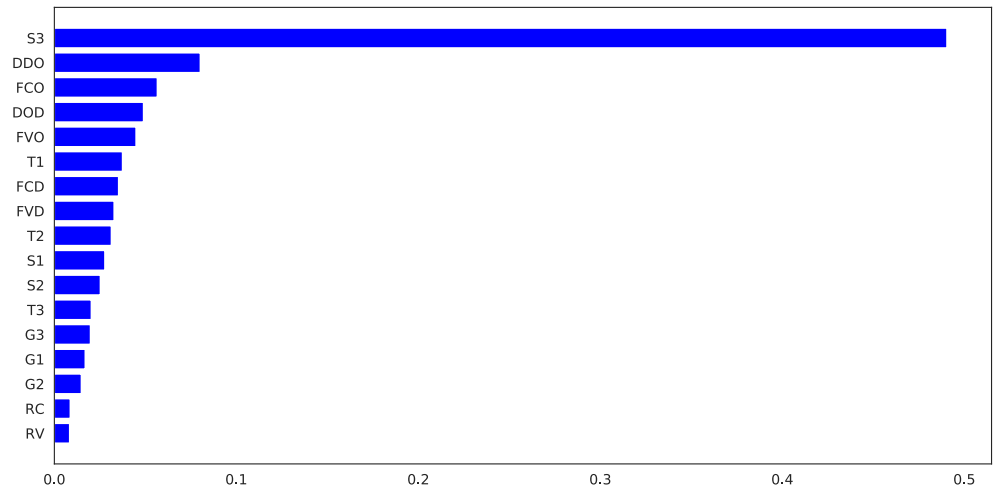


Figura 20 – Classificador *RandomForest* modelo II



Ao consultar o conjunto de dados, é possível compreender o por que de o atributo *S3* ser de grande importância para a predição do modelo II. Basicamente, quando a terceira substituição é ofensiva, das 431 ocorrências, em apenas 55 (12.7%) a substituição é considerada efetiva, enquanto as 376 (87.23%) restantes são não efetivas.

Como é possível verificar no histograma da Figura 14 que trata da distribuição das substituições, geralmente a terceira substituição ocorre na etapa final do jogo, nos últimos 15 minutos. Como dificilmente ocorrem gols por parte do time visitante neste período, as substituições ofensivas não são efetivas.

Ressalta-se a importância do processo de discretização aplicado nas *features* T_1, T_2 e T_3, as quais armazenam os tempos respectivos de cada uma das substituições das três substituições, propostas pelo clube visitante. Se o tempo fosse discretizado utilizando um número menor de classes, isto certamente, resultaria em problemas na predição de substituições ofensivas. Por exemplo, se existissem somente duas classes para representação do tempo (*i.e.*, classe 0 (0-45) e classe 1 (45-90)) o classificador consideraria que toda a substituição ofensiva ocorrida na classe 1, por consequência na segunda etapa, não é efetiva, o que é um equivoco. Embora existam outras *features* em conjunto, neste caso, elas não exercem o mesmo poder de decisão atribuído ao fator tempo.

7 CONCLUSÃO

Este trabalho de conclusão de curso teve como objetivo explorar as substituições propostas pelo time visitante nos anos de 2015 a 2018 do Campeonato Brasileiro de Futebol. Através dos dados históricos disponíveis no conjunto de dados, avaliou-se a possibilidade da previsão da efetividade da segunda e terceira alteração dos times visitantes, a partir da construção de alguns modelos classificadores de aprendizado de máquina.

Para tanto, na etapa de modelagem foram utilizadas *features* relacionadas ao andamento da partida de futebol, além de atributos compostos, construídos com base nos históricos das partidas. Pode-se citar a regra criada para rotulação de substituições como ofensivas, defensivas ou sem alteração tática evidente. Além disso, outra regra foi criada para atribuição do rótulo (efetivo ou não efetivo), assumindo hipóteses sobre a ambição do time visitante em partidas de competições de pontos corridos, e considerando o tipo tático da substituição.

Dois modelos foram produzidos, o primeiro sendo responsável pela previsão da efetividade da segunda substituição e um segundo modelo por realizar a previsão da terceira substituição. Através das métricas de acurácia, precisão, revocação e *F1-Score*, foi possível avaliar os classificadores quanto a sua qualidade.

Percebeu-se que a *feature* mais importante para avaliar a qualidade ou não da substituição está relacionada ao tipo tático da substituição candidata (ofensiva, defensiva ou sem alteração). Entre os três algoritmos testados em cada modelo, o que obteve a melhor acurácia para o modelo I foi o *Support Vector Machine (SVM)* com acurácia de 78.39%. Para o modelo II o algoritmo *Decision Tree (DTR)* alcançou a acurácia de 86.93%.

7.0.1 Trabalhos futuros

Como possíveis trabalhos futuros, pode-se apontar:

- Um aspecto que não foi considerado neste trabalho foram estatísticas da partida em andamento, além de dados relacionados aos jogadores envolvidos na substituição (e.g. desempenho em jogos anteriores). Desta forma, pode-se modificar o algoritmo de atribuição de rótulo para que ele consiga maior capacidade de representatividade com situações reais;

- Estender o modelo para previsão da efetividade para o time da casa, e avaliar a importância das *features*;

- A forma de definição do tipo tático da substituição pode ser melhorado, pois nem sempre a posição de origem é reflexo de sua posição no jogo. O cruzamento entre dados obtidos com mapas de calor e a posição de origem, podem trazer maior precisão na interpretação do posicionamento real;

- Além disso, outra possibilidade é agregar mais dados a este modelo, para que seja avaliado seu desempenho na predição de outras edições do Campeonato Brasileiro. Ou até mesmo em outras ligas nacionais.

REFERÊNCIAS

2018/19 UEFA CHAMPIONS LEAGUE Regulations. 2018. Disponível em: <https://www.uefa.com/MultimediaFiles/Download/Regulations/uefaorg/Regulations/02/55/82/79/2558279_DOWNLOAD.pdf>. Acesso em: 29 nov. 2018.

ALLMERS, Swantje; MAENNIG, Wolfgang. Economic impacts of the FIFA soccer World Cups in France 1998, Germany 2006, and outlook for South Africa 2010. **Eastern economic journal**, Springer, v. 35, n. 4, p. 500–519, 2009.

BRILLINGER, David R. Soccer/world football. **Encyclopedia of Operations Research and Management Science. New York: Wiley**, 2010.

CBF. **Regulamento Específico do Campeonato Brasileiro de Futebol, Série-A 2018**. 2018. Disponível em: <https://conteudo.cbf.com.br/cdn/201712/20171218105845_0.pdf>. Acesso em: 29 nov. 2018.

_____. **Regulamento Geral das Competições**. 2017. Disponível em: <https://conteudo.cbf.com.br/cdn/201712/20171218105845_0.pdf>. Acesso em: 29 nov. 2018.

DEL CORRAL, Julio; BARROS, Carlos Pestana; PRIETO-RODRIGUEZ, Juan. The determinants of soccer player substitutions: a survival analysis of the Spanish soccer league. **Journal of Sports Economics**, Sage Publications Sage UK: London, England, v. 9, n. 2, p. 160–172, 2008.

DUARTE, Denio; STÅHL, Niclas. Machine learning: a concise overview. In: DATA Science in Practice. [S.l.]: Springer, 2019. p. 27–58.

DUNMORE, Thomas; MURRAY, Scott. **Soccer for dummies**. [S.l.]: John Wiley & Sons, 2013.

GIULIANOTTI, Richard. Football. **The Wiley-Blackwell Encyclopedia of Globalization**, Wiley Online Library, 2012.

GLEZ-PEÑA, Daniel et al. Web scraping technologies in an API world. **Briefings in bioinformatics**, Oxford University Press, v. 15, n. 5, p. 788–797, 2013.

GOMEZ, Miguel-Angel; LAGO-PEÑAS, Carlos; OWEN, L Adam. The influence of substitutions on elite soccer teams' performance. **International Journal of Performance Analysis in Sport**, Taylor & Francis, v. 16, n. 2, p. 553–568, 2016.

HAN, Jiawei; KAMBER, Micheline; PEI, Jian. Data mining: concepts and techniques, (the morgan kaufmann series in data management systems). **pp-230-240**, 2006.

HARRINGTON, Peter. **Machine learning in action**. [S.l.]: Manning Publications Co., 2012.

HIROTSU, Nobuyoshi; WRIGHT, Michael. Using a Markov process model of an association football match to determine the optimal timing of substitution and tactical decisions. **Journal of the Operational Research Society**, Taylor & Francis, v. 53, n. 1, p. 88–96, 2002.

KOTSIANTIS, SB; KANELLOPOULOS, Dimitris; PINTELAS, PE. Data preprocessing for supervised leaning. **International Journal of Computer Science**, Citeseer, v. 1, n. 2, p. 111–117, 2006.

KUMAR, Gunjan. **Machine Learning for Soccer Analytics**. Set. 2013. Tese (Doutorado). DOI: 10.13140/RG.2.1.4628.3761.

MARSLAND, Stephen. **Machine Learning: An Algorithmic Perspective, Second Edition**. 2nd. [S.l.]: Chapman & Hall/CRC, 2014. ISBN 1466583282, 9781466583283.

MITCHELL, Thomas M. **Machine Learning**. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.

MURPHY, Kevin P. **Machine Learning: A Probabilistic Perspective**. [S.l.]: The MIT Press, 2012. ISBN 0262018020, 9780262018029.

MYERS, Bret R. A proposed decision rule for the timing of soccer substitutions. **Journal of Quantitative Analysis in Sports**, De Gruyter, v. 8, n. 1, 2012.

OJA, Pekka et al. Health benefits of different sport disciplines for adults: systematic review of observational and intervention studies with meta-analysis. **Br J Sports Med**, BMJ Publishing Group Ltd, British Association of Sport e Exercise Medicine, v. 49, n. 7, p. 434–440, 2015.

REIN, Robert; MEMMERT, Daniel. Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science. **SpringerPlus**, SpringerOpen, v. 5, n. 1, p. 1410, 2016.

REY, Ezequiel; LAGO-BALLESTEROS, Joaquín; PADRÓN-CABO, Alexis. Timing and tactical analysis of player substitutions in the UEFA Champions League. **International Journal of Performance Analysis in Sport**, Taylor & Francis, v. 15, n. 3, p. 840–850, 2015.

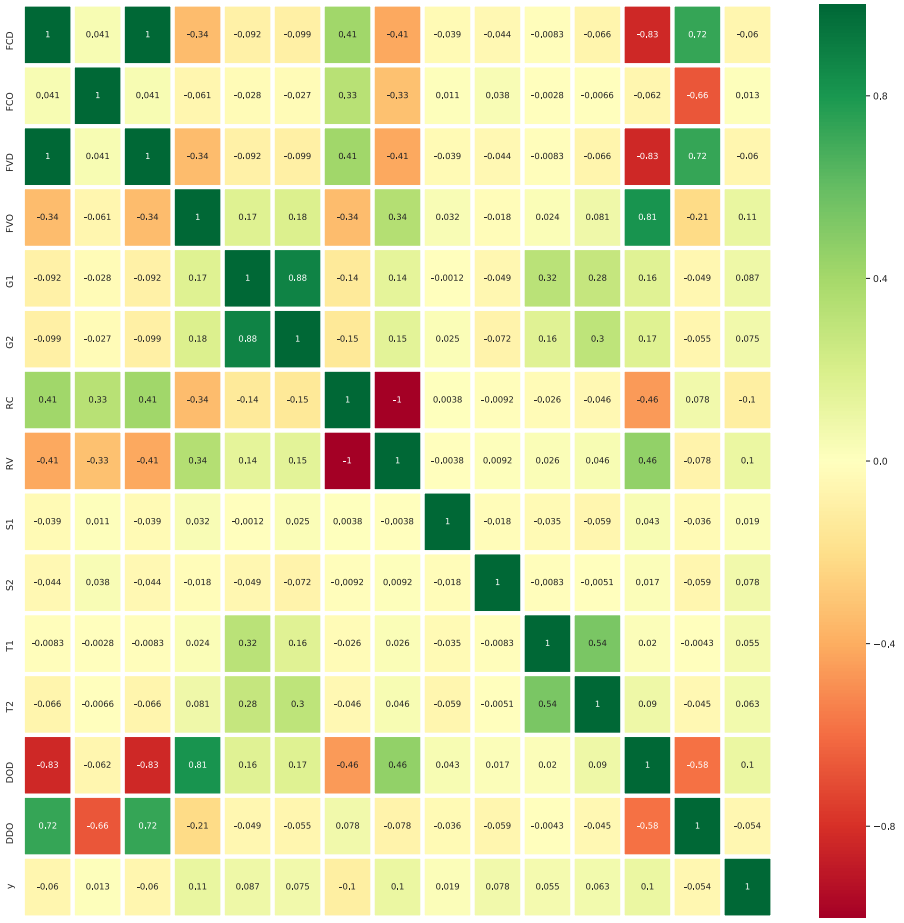
SILVA, Rajitha M; SWARTZ, Tim B. Analysis of substitution times in soccer. **Journal of Quantitative Analysis in Sports**, De Gruyter, v. 12, n. 3, p. 113–122, 2016.

VARGIU, Eloisa; URRU, Mirko. Exploiting web scraping in a collaborative filtering-based approach to web advertising. **Artif. Intell. Research**, v. 2, n. 1, p. 44–54, 2013.

APÊNDICE A – PARÂMETROS DOS CLASSIFICADORES

- *Random Forest Classifier:*
  - *n_estimators*: define o número de árvores;
  - *max_features*: o número de recursos a serem considerados ao procurar a melhor divisão;
  - *max_depth*: a profundidade máxima da árvore;
  - *criterion*: define a função que irá medir a qualidade de uma divisão.
- *Support Vector Machine:*
  - *kernel*: define a função utilizada para mapeamento;
  - *C*: parâmetro que define a penalidade do erro;
  - *gamma*: coeficiente de *kernel*;
  - *decision_function_shape*: permite agregar os resultados dos classificadores "um contra um" a uma função de decisão;
  - *shrinking*: define se será usado a heurística *shrinking* ou não.
- *KNeighborsClassifier:*
  - *n_neighbors*: define o número de vizinhos, por padrão é utilizado 5;
  - *weights*: função peso utilizada na previsão;
  - *metric*: define a métrica utilizada para o calculo da distância de um ponto a outro.
- *Decision Tree:*
  - *max_depth*: profundidade máxima da árvore;
  - *min_samples_leaf*: mínimo de amostras necessárias para estar em um nó folha;
  - *criterion*: define a função que medirá a qualidade da divisão.

APÊNDICE B – MAPA DE CALOR MODELO I



APÊNDICE C – MAPA DE CALOR MODELO II