UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

CRISTIAN FELIPE SCHNEIDER

MACHINE LEARNING APLICADO NA PREVISÃO DE RESULTADOS DE PARTIDAS DE FUTEBOL: UM ESTUDO DE CASO PARA COMPARAÇÃO DE DIFERENTES CLASSIFICADORES

Porto Alegre
2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

MACHINE LEARNING APLICADO NA PREVISÃO DE RESULTADOS DE PARTIDAS DE FUTEBOL: UM ESTUDO DE CASO PARA COMPARAÇÃO DE DIFERENTES CLASSIFICADORES

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Engenheiro Eletricista.

ORIENTADORA: Prof.ª Dra. Mariana R. Mendoza

Porto Alegre
2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

CRISTIAN FELIPE SCHNEIDER

MACHINE LEARNING APLICADO NA PREVISÃO DE RESULTADOS DE PARTIDAS DE FUTEBOL: UM ESTUDO DE CASO PARA COMPARAÇÃO DE DIFERENTES CLASSIFICADORES

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de "Projeto de Diplomação", do Departamento de Engenharia Elétrica e aprovado em sua forma final pela Orientadora e pela Banca Examinadora.

Orientadora: _____

Prof.ª Dra. Mariana Recamonde Mendoza

Doutora pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Banca Examinadora:

Prof.ª Dra. Leia Bernardi Bagesteiro, UFRGS

Doutora pela University of Surrey.– Guildford, Inglaterra

Prof. Dr. Bruno Castro da Silva, UFRGS

Doutor pela University of Massachusetts – Amherst, Estados Unidos

Porto Alegre, janeiro de 2018.

CIP - Catalogação na Publicação

Schneider, Cristian Felipe
Machine learning aplicado na previsão de resultados de partidas de futebol: um estudo de caso para comparação de diferentes classificadores / Cristian Felipe Schneider. -- 2018.
92 f.
Orientadora: Mariana Recamonde Mendoza.

Trabalho de conclusão de curso (Graduação) -- Universidade Federal do Rio Grande do Sul, Escola de Engenharia, Curso de Engenharia Elétrica, Porto Alegre, BR-RS, 2018.

1 machine learning. 2. previsão. 3. futebol. I. Mendoza, Mariana Recamonde, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os dados fornecidos pelo(a) autor(a).

DEDICATÓRIA

Dedico este trabalho aos meus pais, Judithe e Alecio, por todo o apoio e suporte durante todos estes anos, principalmente durante a graduação. Dedico também a minha namorada, Ana Luiza, pelo apoio constante nos momentos mais difíceis. Por fim, dedico este trabalho a todos aqueles que se empenham em distribuir conhecimento de forma livre.

AGRADECIMENTOS

Agradeço à minha orientadora, Profª. Mariana Recamonde Mendoza, pela paciência e dedicação durante a orientação deste trabalho. Agradeço a todos aqueles que, de qualquer forma, me auxiliaram durante os anos de graduação em Engenharia Elétrica. Em especial, agradeço aos colegas da turma de Engenharia Elétrica 2012/2 pelo companheirismo e amizade durante todos estes anos. Em especial, agradeço ao Marcelo, Luiz Artur, Natália, Aline, Daniel, Meire e Ana Luiza pela revisão do texto.

RESUMO

A previsão de eventos esportivos, em especial o futebol, através da utilização de algoritmos de *machine learning* é uma área de pesquisa que vem ganhando destaque, em parte devido à grande popularidade do esporte e o aumento da quantidade de dados disponíveis. O presente trabalho apresenta um estudo do desempenho de diferentes classificadores quando os mesmos são utilizados para prever o resultado de partidas de futebol do campeonato *Premier League*. A partir de dados que retratam partidas realizadas no período 2000-2017, comparou-se o desempenho de dez classificadores, variando-se as *features* utilizadas, otimizando-se seus hiperparâmetros e variando-se a quantidade de partidas utilizadas. Assim, analisando-se a acurácia máxima, obteve-se valores de acurácia iguais a 54.73% ± 2.06% (Regressão Logística), 54.82% ± 2.01 (Análise Discriminante Linear), 54.35% ± 2.12% (*SVM* com *kernel* linear) e 54.34% ± 1.93% (*K-Nearest Neighbors*). Já um classificador *ensemble* composto de seis classificadores selecionados apresentou acurácia de 57% ao analisar um conjunto de dados de teste. Comparando-se com as *baselines* definidas: vitória do mandante (46.56% de acurácia) e time com maior ELO (50.24% de acurácia), os classificadores com melhor desempenho ultrapassaram as *baselines*, chegando a valores mais expressivos do que grande parte da literatura utilizada como base com relação à acurácia. No entanto, observou-se que grande parte dos classificadores apresentaram poucas ou nenhuma previsão de instâncias como sendo pertencentes à classe empate. O classificador que obteve melhor desempenho ao prever empates foi a Análise Discriminante Quadrática. Porém, o mesmo apresentou a menor acurácia dentre os classificadores observados (42.63% ± 2.33%). Desta forma, o aumento do desempenho da acurácia dos classificadores ao prever resultados de partida de futebol está diretamente conectado à necessidade de solucionar o problema relacionado com a previsão de empates.

Palavras-chaves: *Machine learning*, futebol, previsão.

ABSTRACT

The prediction of sport events through machine learning algorithms, especially in football, is a subject which has been in the spotlight lately, mainly because of the increase in the amount of data available and the popularity of the game. This work presents a study focused on the comparison of several classifiers when these are used to predict the result of football matches from the Premier League, an English tournament. Using data from the seasons comprised in the period of 2000-2017, a comparison between ten classifiers was made. A selection of features was conducted, allied with the optimization of the classifiers and the search for the optimal number of matches ignored at the beginning of each season. The best accuracy obtained was 54.73% ± 2.06% (Logistic Regression), 54.82% ± 2.01 (Linear Discriminant Analysis), 54.35% ± 2.12% (SVM with linear linear) and 54.34% ± 1.93% (K-Nearest Neighbors). An ensemble classifier was created using six different classifiers. The ensemble obtained 57% of accuracy when analyzing a new test data set. Comparing these results to the defined baselines (hometeam victory, with 46.56% of accuracy, and victory of the team with the biggest ELO, with 50.24% of accuracy) it was proved that the best classifiers surpassed all the baselines, scoring better than most of the related literature. It was possible to conclude that the majority of the classifiers presented few or zero predictions regarding the draw class. The classifier that better predicted draws was the Linear Discriminant Analysis. However, this classifier presented the worst accuracy among all the classifiers analysed (42.63% ± 2.33%). In this way, in order to achieve greater accuracy, it is necessary to study the problem related to draw predictions.

Keywords: *Machine learning*, football, prediction.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS

ADL: *Análise Discriminante Linear*

ADQ: *Análise Discriminante Quadrática*

CSV: *Comma-Separated Values*

ET: *Extra Trees*

ENS: *Ensemble*

FP: *Falso Positivo*

FN: *Falso Negativo*

IA: *Inteligência Artificial*

KNN: *K-Nearest Neighbors*

MLG: *Modelos Lineares Generalizados*

NBG: *Naive Bayes Gaussiano*

NBM: *Naive Bayes Multinomial*

NFL: *No Free Lunch*

RBF: *Radial Basis Function*

RL: *Regressão Logística*

RF: *Random Forest*

SVM: *Support Vector Machine*

SVML: *Support Vector Machine com kernel linear*

SVMR: *Support Vector Machine com kernel RBF*

SGD: *Stochastic Gradient Descent*

SV: *Support Vectors*

VP: *Verdadeiro Positivo*

VN: *Verdadeiro Negativo*

2 INTRODUÇÃO

O futebol é o esporte mais praticado e assistido no mundo (DOBSON; GODDARD, 2001), sendo a *Premier League* (EPL, *English Premier League*) a liga futebolística com maior audiência, com estimativas de que, cumulativamente, 4.7 bilhões de pessoas assistiram à temporada 2010-11 deste campeonato (TIMMARAJU; PALNITKAR; KHANNA, 2013).

Nos últimos anos, observou-se um aumento na utilização de métodos estatísticos para identificação de fatores determinantes para a previsão de resultados esportivos (HOPKINS, 2010), sendo a *Premier League* um dos campeonatos mais analisados, devido em parte à sua popularidade e audiência, atraindo a atenção também de casas de apostas as quais investem recursos para desenvolvimento de algoritmos de previsão de resultados esportivos (ULMER; FERNANDEZ, 2013).

Assim, observa-se que um dos problemas relacionados a previsões esportivas é a previsão do resultado final de partidas de futebol (vitória do mandante, vitória do visitante ou empate), sabendo que atualmente a acurácia média dos melhores modelos de previsão encontrados na literatura não ultrapassa a figura de 60% (JOSEPH; FENTON; NEIL, 2006). Através da comparação de diversos classificadores, observa-se que a previsão do resultado de partidas de futebol através da utilização de classificadores de *machine learning* esbarra em um problema relacionado com a previsão de empates, visto que diversos autores, como Ulmer e Fernandez (ULMER; FERNANDEZ, 2013) e Trindade (TRINDADE, 2013) observaram acurácia inferior na previsão de instâncias como sendo pertencentes a classe empate quando comparada com a acurácia da classificação de instâncias como vitória do mandante ou visitante. Assim, é possível confirmar a necessidade de mais estudos de forma a possibilitar a replicação ou a solução deste problema ao se utilizar diferentes abordagens.

Desta forma, neste trabalho busca-se discutir alternativas para previsão do resultado final de partidas de futebol utilizando-se métodos de aprendizado de máquina (*machine*

learning). O objetivo principal deste trabalho é realizar uma avaliação do desempenho de diversos algoritmos de classificação quando os mesmos são utilizados para prever resultados de partidas de futebol contidas entre as temporadas 2000-01 e 2016-17 da *Premier League*, comparando os resultados obtidos com a literatura existente.

Este trabalho está organizado da seguinte forma: no Capítulo 2 é apresentado um referencial teórico o qual aborda conceitos fundamentais relacionados com a área de *machine learning*; no Capítulo 3 aborda-se técnicas e conceitos relacionados ao domínio estudado; no Capítulo 4 apresenta-se a metodologia a qual foi seguida de forma a alcançar os resultados apresentados no Capítulo 5, enquanto que conclusões finais deste trabalho são apresentadas no Capítulo 6.

3 REFERENCIAL TEÓRICO

Este capítulo apresenta o embasamento teórico e conceitos utilizados neste trabalho.

3.1 INTRODUÇÃO A *MACHINE LEARNING*

O termo *Machine Learning* geralmente é utilizado para se referir a mudanças em sistemas que desempenham tarefas associadas à inteligência artificial (IA). Estas tarefas podem incluir diagnósticos, controle, previsões, planejamento, etc. A vantagem de se utilizar agentes que podem se modificar, ao invés de utilizar agentes explicitamente projetados para uma tarefa específica, é baseada no fato de que algumas tarefas são difíceis de serem definidas senão através de exemplos. Além disso, é possível que, escondido em uma grande quantidade de dados, existam importantes correlações as quais podem ser extraídas através de métodos de *machine learning*. Somado a isso, erros humanos de projeto podem prejudicar o desempenho de sistemas projetados para tarefas específicas, sendo que parte destes erros podem ser causados por ambientes em modificação. Sendo assim, sistemas que possuem a habilidade de se modificarem ao longo do tempo podem reduzir estes erros e reduzir a necessidade por manutenções constantes (NILSSON, 1998).

Algumas áreas de aplicação de métodos de *machine learning* são:

- Inteligência Artificial: utilizou-se *machine learning* para estudar o papel de analogias no aprendizado (MICHALSKI; CARBONELL; MITCHELL, 1983) e como ações futuras podem ser previstas através de exemplos passados (KOLODNER, 1993).
- Controle Adaptativo: *machine learning* pode ser utilizado para estudar o problema de controle de processos os quais possuem parâmetros desconhecidos que devem ser estimados durante operação (BOLLINGER; DUFFIE, 1988).

- Estatística: um problema no qual se utilizou métodos de *machine learning* na área de estatística é relacionado com a utilização de amostras extraídas de distribuições de probabilidade desconhecidas para auxiliar na identificação da distribuição probabilística a qual uma nova amostra pertence (NILSSON, 1998).

3.2 APRENDIZADO SUPERVISIONADO

De forma abstrata, é possível explicar a tarefa de aprendizado supervisionado através da frase "utilizar experiência para ganhar *expertise*". Em uma aplicação de aprendizagem supervisionada, a "experiência" deriva de um conjunto de exemplos denominado dados de treinamento, os quais contém informação significativa e completa para aprender a relação entre os valores de variáveis independentes e dependentes, isto é, são conhecidos os valores para variáveis independentes e dependentes de cada exemplo. A *expertise* adquirida a partir deste processo é então utilizada para realizar inferências a respeito de novos exemplos, com valor desconhecido para as variáveis dependentes, denominados dados de teste (SHALEV-SCHWARTZ; BEN-DAVID, 2014). Mais especificamente, aprendizagem supervisionada possui como objetivo a construção de um modelo (também chamado de classificador (SIULY; LI, Y.; ZHANG, Y., 2017)) capaz de aprender o mapeamento entre os valores das variáveis independentes ou preditoras (chamadas de *features*, seguindo o padrão da literatura utilizada) e o valor da variável dependente (classes ou *labels*, no contexto de *machine learning*) das instâncias contidas em um conjunto de treinamento, tal que o classificador resultante possua um poder de generalização suficiente para ser utilizado para predição de instâncias nunca antes vistas (KOTSIANTIS, 2007).

Esta seção detalhará os diferentes classificadores citados neste trabalho.

3.2.1 REGRESSÃO LOGÍSTICA (RL)

Apesar do nome, a regressão logística é um classificador linear, membro do conjunto de modelos de regressão linear chamados Modelos Lineares Generalizados (MLG) (NG, 2008). Nesta seção, serão detalhadas as hipóteses e derivações deste método, assim como diferentes métodos de regularização utilizados para melhorar o desempenho das previsões.

As definições detalhadas nesta seção utilizam x_i como a *i-ésima feature* e θ_j como o *j-ésimo* parâmetro do modelo, sabendo que para uma simples regressão linear a variável dependente (classe prevista) é dada pela Equação (1).

$$y = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n \tag{1}$$

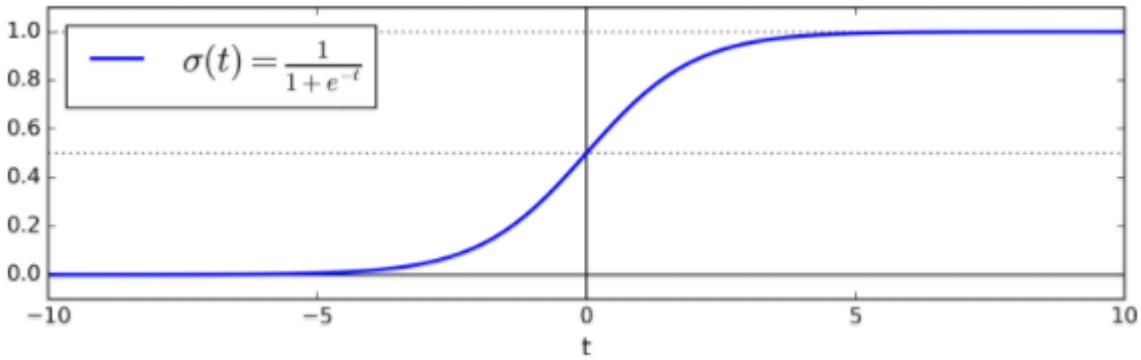
A regressão logística (também chamada de Regressão Logit) é comumente utilizada para estimar a probabilidade de uma instância pertencer a uma classe em particular (ex.: a probabilidade de um *e-mail* ser *spam*). Em uma classificação binária, caso esta probabilidade seja maior do que 50%, o modelo classificará a instância como pertencente à classe em questão (chamada de classe positiva 1). Caso contrário, o modelo classificará a instância como pertencente à classe negativa (0) (GÉRON, 2017).

Para tanto, a regressão logística calcula uma soma ponderada das *features* e distribui a saída seguindo uma distribuição logística seguindo uma curva do tipo sigmóide, demonstrada na e definida pela Equação (2). A saída calculada pela regressão logística é uma estimativa de probabilidade definida pela Equação (3) (GÉRON, 2017).

$$\sigma(\mathrm{t}) = \frac{1}{1 + e^{-t}} \tag{2}$$

$$p = h_\theta(\mathbf{x}) = \sigma(\theta^T \cdot \mathbf{x}) \tag{3}$$

Figura 1 - Função logística.



Fonte: Adaptado de (GÉRON, 2017)

Na regressão logística, o objetivo do treinamento é definir o vetor θ de forma que o modelo estime probabilidades elevadas para o caso de instâncias positivas ($y = 1$, ou seja, se a instância pertence à classe verdadeira) e probabilidades baixas para instâncias negativas ($y = 0$, ou seja, se a instância pertence à classe falsa). Para tanto, a regressão logística utiliza uma função de custo, mostrada na Equação (4) e na Equação (5).

$$c(\theta) = -\log(p), \text{ se y} = 1 \tag{4}$$

$$c(\theta) = -\log(1-p), \text{ se y} = 0 \tag{5}$$

onde p é calculado através da Equação (3).

A função de custo cresce de forma expressiva quando t se aproxima de 0, fazendo com que o custo seja alto quando o modelo estima uma probabilidade próxima de 0 para uma instância positiva, sendo que o mesmo ocorre quando o modelo erroneamente estima uma probabilidade alta (t próximo de 1, ou seja, $\theta^T \boldsymbol{x}$ próximo de 1) para uma instância negativa. O custo sobre todo o conjunto de dados de treinamento é a média do custo sobre todas as instâncias de treinamento, definido pela Equação (6).

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(p^{(i)}) + (1-y^{(i)})\log(1-p^{(i)})\right] \tag{6}$$

onde *y* (classe prevista) assume o valor de 0 se p <0.5 e 1 se p >0.5.

Deve-se notar que não existe uma forma fechada de se calcular a Equação (6). Porém, como a mesma é convexa, a utilização de um algoritmo de otimização garante o encontro do máximo global. A derivação parcial da Equação (6) com relação a θ_j é dada pela Equação (7).

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (\sigma(\theta^T \cdot \mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)} \tag{7}$$

Analisando a Equação (7) é possível observar que para cada instância será calculado o erro da previsão, multiplicando-o pelo valor da *feature j*, calculando então a média de todas as instâncias de treinamento (GÉRON, 2017).

3.2.1.1 REGRESSÃO *SOFTMAX*

O modelo de regressão logística pode ser generalizado para suportar problemas multiclasse através da utilização da regressão *softmax* (também chamada de Regressão Logística Multinomial).

Quando uma instância *x* é fornecida, o modelo de regressão *softmax* calcula $S_k(x)$ para cada classe *k* e então estima a probabilidade para cada classe aplicando a função *softmax* (também chamada de exponencial normalizada).

$$S_k = (\theta^{(k)})^T \cdot \mathbf{x} \tag{8}$$

Assim, ao se possuir todos os valores para todas as classes de cada instância *x*, pode-se estimar as probabilidades p_k para todas as classes.

$$p_{\ k} = \sigma(\mathbf{s}(\mathbf{x}))_{\ k} = \frac{e^{(\mathbf{s}_k(\mathbf{x}))}}{\sum_{j=1}^{K} e^{(\mathbf{s}_j(\mathbf{x}))}} \tag{9}$$

onde *k* é a quantidade de classes, $\mathbf{s}(\mathbf{x})$ é o vetor contendo todos os valores da função *softmax* para cada instância $\mathbf{x}$ e $\sigma(\mathbf{s}(\mathbf{x}))_k$ a probabilidade estimada para que a instância $\mathbf{x}$ pertença à classe *k* dado o valor da função *softmax* (GÉRON, 2017).

3.2.1.2 REGRESSÃO LOGÍSTICA PENALIZADA L1 E L2

A regularização é utilizada em *machine learning* para aumentar o desempenho preditivo de um modelo através da imposição de restrições nos parâmetros do modelo. Um exemplo intuitivo é a hipótese de que grande parte dos componentes de θ (ou seja, os coeficientes de diferentes variáveis da combinação linear) devem ser próximos ou iguais a zero. Esta hipótese pode ser alcançada através da criação de uma função de custo $C(\theta)$[1], mostrada na Equação (10) (GYULA KRISZTIÁN, 2014),

$$C_{\mathrm{L2}} = -l(\theta) + \frac{1}{2}\lambda \| \theta \|_2^2 \tag{10}$$

onde $l(\theta)$ é *log* $L(\theta)$ (log p (y | **X**; θ), sabendo também que ***X*** é o vetor de *features* e y a classe prevista. De forma similar, a Equação (11) apresenta a penalização L1 (GYULA KRISZTIÁN, 2014)

$$C_{\mathrm{L1}} = -l(\theta) + \lambda \sum_{i=1}^{n} | \theta_i | \tag{11}$$

3.2.2 *SUPPORT VECTOR MACHINE* (SVM)

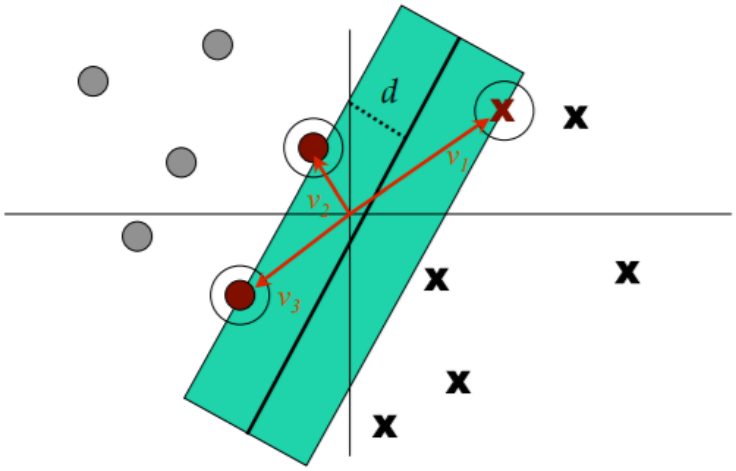
O classificador *Support Vector Machine* (SVM) se baseia na construção de diversos hiperplanos (superfície de decisão) utilizados para separar diferentes amostras de forma a se encontrar o hiperplano ótimo, o qual representa uma determinada instância. Este classificador realiza uma representação de amostras como pontos no espaço, mapeadas de forma que os exemplos pertencentes a classes distintas apresentem uma divisão espacial (margem) bem definida (ZHANG, T., 2001). O hiperplano pode ser definido através da Equação (12),

$$W^T x + b = 0, x \in R^d \tag{12}$$

[1] A biblioteca Scikit-learn utiliza $C_{\mathrm{L2}} = -\lambda l(\theta) + \| \theta \|_2^2$.

A margem é definida através de exemplos de treinamento os quais são chamados de vetores de suporte (*support vectors*, SV), sendo estas as amostras de mais difícil classificação.

Figura 2 - Exemplo de margem definida através de três vetores de suporte (v1, v2 e v3).



Fonte: (BERWICK, 2003).

De forma a se encontrar os hiperplanos com margem máxima, Cortes e Vapnik (1995) sugerem a utilização de *kernels,* os quais modificam a função que define a margem ótima de separação das classes no hiperplano. Assim, define-se o *kernel* linear através da Equação (13)

$$K(x_i, x_j) = x_i^T x_j \tag{13}$$

considerando-se duas amostras com vetores x_i e x_j. Outro *kernel* existente é o *Radial Basis Function* (RBF), o qual apresenta uma simplificação significativa nas computações necessárias para busca do hiperplano e margem ótimos, sendo o mesmo definido pela Equação (14) (CORTES; VAPNIK, 1995).

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \tag{14}$$

3.2.2.1 MARGENS NÃO SEPARÁVEIS

Nem sempre será possível encontrar uma margem que separe completamente as classes. Desta forma, busca-se encontrar a margem a qual minimiza o erro associado a

incompleta separação das margens. Para tanto, considera-se que um ponto possui margem da forma $1-\xi_i$, de forma que seja possível penalizar estes pontos multiplicando-os por um fator de controle C o qual controla o balanço entre o tamanho da margem e o erro. Assim, este problema de otimização é descrito pela Equação (15) (GYULA KRISZTIÁN, 2014).

$$\min_{W,b} \frac{1}{2} \| W \|^2 + C \sum_{i=1}^{m} \xi_i \tag{15}$$

3.2.3 *RANDOM FOREST* (RF)

Recentemente pode-se observar grande interesse em métodos *ensemble* os quais combinam os resultados de diferentes classificadores em uma única previsão. Porém, para este conceito ser válido, deve-se ter certeza de que os diferentes classificadores utilizados são independentes entre si. Para tanto, uma técnica popular de *ensembles* é chamada *bagging*, onde diversos classificadores geram resultados em amostras do conjunto de treinamento. Após, o voto da maioria dos resultados é utilizado como resultado final do classificador *ensemble* (GYULA KRISZTIÁN, 2014).

Em *Random Forests*, adiciona-se uma camada extra de aleatoriedade em um classificador do tipo árvore de decisão *bagging* (BREIMAN, 1996) durante a construção da árvore.

3.2.3.1 ÁRVORES DE DECISÃO

O classificador árvore de decisão classifica novas instâncias através da estratégia dividir para conquistar, onde um problema maior é dividido em sub-problemas menores, sendo esta estratégia aplicada recursivamente (GAMA, 2004). A capacidade de discriminação de uma árvore é gerada através da divisão espacial das *features*, sendo que a cada sub-espaço é associada uma classe (SILVA, 2005a).

3.2.3.2 O ALGORITMO *RANDOM FOREST*

O classificador *Random Forest* é uma combinação de árvores de decisão de forma que cada árvore depende do valor de um vetor aleatório amostrado independentemente o qual possui a mesma distribuição para todas as árvores da floresta (conjunto de árvores). Assim, o erro de generalização da floresta converge ao ponto em que se aumenta a quantidade de árvores. O erro de generalização de uma floresta depende da força individual de cada árvore e da correlação entre as mesmas (BREIMAN, 2001a). Ao se utilizar uma amostra aleatória de *features* para dividir cada nó é criado um erro que geralmente é menor do que o observado por outros classificadores semelhantes, como o classificador *Adaboost* (FREUND; SCHAPIRE, 1999), sendo também mais robustas quanto à presença de ruído nos dados (BREIMAN, 2001b).

Dado um *ensemble* de classificadores $h_1(\mathbf{x}), h_2(\mathbf{x}), ..., h_K(\mathbf{x})$ e considerando-se também que o conjunto de treinamento foi amostrado aleatoriamente da distribuição do vetor aleatório **X**,Y, define-se a função de margem do classificador *Random Forest* através da Equação (16)

$$mg(\mathbf{X}, Y) = med_K I(h_K(\mathbf{X}) = Y) - \max_{j \neq Y} med_K I(h_K(\mathbf{X}) = j) \quad (16)$$

onde I(.) denota a função indicativa. A margem mede o quanto a média de votos, no ponto **X**,Y e para uma classe em específica, ultrapassa a média de votos para todas as demais classes. Assim, a margem é diretamente proporcional ao intervalo de confiança da classificação (BREIMAN, 2001b).

Em classificadores do tipo *Extremely Randomized Trees* (*Extra Trees*, ET), a aleatoriedade é utilizada em um passo extra: em vez de se analisar pelos limites mais discriminantes, os limites os quais determinam as margens também são analisados de forma aleatória para cada amostra de *features*, sendo que os melhores limites aleatórios os quais definem as margens são definidos como regra de divisão dos nós.

3.2.3.3 ENTROPIA E ÍNDICE GINI

De acordo com Silva (2005), a entropia é o cálculo do ganho de informação baseado em uma medida utilizada na teoria da informação. A entropia caracteriza a (im)pureza dos dados. Assim, em um conjunto de dados, é uma medida da falta de homogeneidade dos dados de entrada em relação a sua classificação. A entropia é definida pela Equação (17),

$$Entropia(S) = \sum_{i=1}^{C} -p \log_2 p_i \tag{17}$$

onde p_i é a proporção dos dados em S os quais pertencem à classe i e C é o número de classes. Assim, do ponto de vista da entropia, uma árvore de decisão tem o objetivo de diminuir a entropia. Ou seja, a árvore de decisão deve diminuir a aleatoriedade da classe prevista (SILVA, 2005).

Ainda de acordo com Silva, o Índice Gini mede o grau de heterogeneidade dos dados. O Índice de Gini é definido pela Equação (18),

$$Gini(\mathrm{p}) = 1 - \sum_{i=1}^{C} p_i{}^2 \tag{18}$$

onde p_i é a frequência relativa de cada classe em cada nó e C é o número de classes (SILVA, 2005).

3.2.4 *K-NEAREST NEIGHBORS*

De acordo com Silva, o classificador KNN é um classificador baseado na analogia, onde o conjunto de treinamento é formado por vetores de n-dimensões, sendo que cada elemento deste conjunto representa um ponto no espaço n-dimensional (2005).

De forma a classificar um elemento o qual não pertence ao conjunto de treinamento, o classificador KNN procura K elementos no conjunto de treinamento, sendo que estes K elementos devem estar próximos do elemento desconhecido. Os K elementos próximos são

denominados K-vizinhos mais próximos (do inglês *K-Nearest Neighbors*). Desta forma, verifica-se quais são as classes dos K-vizinhos, sendo que a classe mais frequente é atribuída ao elemento desconhecido. A métrica mais comum para determinação dos K-vizinhos mais próximos é a distância euclidiana, definida pela Equação (19) (SILVA, 2005).

$$d(x,y) = \sqrt{(\mathrm{x}_1 - y_1)^2 + (\mathrm{x}_2 - y_2)^2 + \ldots + (\mathrm{x}_n - y_N)^2} \tag{19}$$

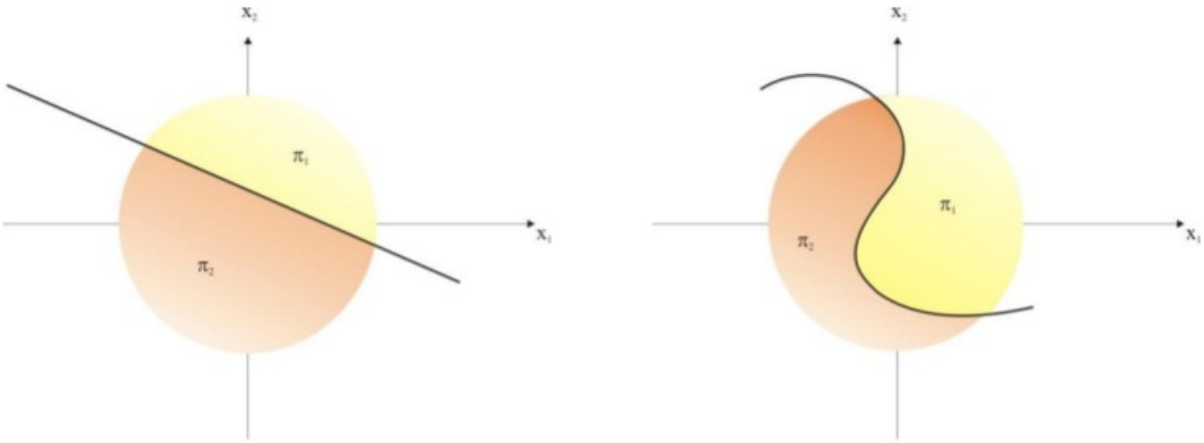
3.2.5 ANÁLISE DISCRIMINANTE

A Análise Discriminante é uma técnica de estatística multivariada utilizada para discriminar e classificar objetos (VARELLA, 2004). De acordo com (KHATTREE; NAIK, 2003), a Análise Discriminante estuda a separação de objetos de uma população em classes, onde a discriminação é a primeira etapa a qual procura por características capazes de serem utilizadas apara alocar objetos em diferentes grupos pré-definidos. Assim, as mesmas regras que servem para alocar objetos podem ser utilizadas para separar objetos (VARELLA, 2004). Assim, a tarefa de discriminação visando posterior classificação consiste em obter funções matemáticas capazes de classificar uma instância *X* em uma das várias populações com base em medidas de um número *p* de *features* com o objetivo de minimizar a probabilidade de classificação errônea. Em resumo, o problema consiste em obter-se uma combinação de características observadas as quais apresentem o maior poder de discriminação entre as populações. Assim, para esta combinação se dá o nome de função discriminante (VARELLA, 2004).

3.2.5.1 REGIÕES DE ALOCAÇÃO

As regiões de alocação são conjuntos de valores separados por uma fronteira definida por uma função discriminante, sendo que esta função é obtida através de amostras de treinamento (VARELLA, 2004).

Figura 3 - Exemplo de regiões de alocação.



Fonte: (VARELLA, 2004)**.**

3.2.5.2 REGRAS DE CLASSIFICAÇÃO

Para que uma classificação seja eficiente, a mesma deve resultar em pequenos erros (VARELLA, 2004). Deve-se considerar as probabilidades a priori e os custos de má classificação. Adicionalmente, as regras de classificação devem considerar as variâncias das populações. Assim, quando uma regra de classificação assume que as variâncias são iguais, a função discriminante é denominada linear. Em contraste, quando as variâncias são consideradas diferentes, denomina-se a função discriminante como quadrática (VARELLA, 2004). A Equação (20) apresenta a Função Discriminante de Fisher,

$$D(X) = L'\mathrm{X} = [\mu_1 - \mu_2]' \sum^{-1} X \tag{20}$$

onde X é o vetor aleatório de características (*features*) das populações, L é o vetor discriminante, μ é o vetor de médias p-variado e Σ é a matriz de covariâncias das populações (VARELLA, 2004).

3.2.6 NAIVE BAYES

Os métodos Naive Bayes de classificação são um conjunto de algoritmos de aprendizado supervisionado o qual se baseia na aplicação do Teorema de Bayes com a hipótese *naive* (ingênua) de independência entre cada par de *features* (ZHANG, H., 2004). Assim, dada uma classe *y* e m vetor de *features x*, o Teorema de Bayes define a relação:

$$P(y \mid x_1, ..., x_N) = \frac{P(y)P(x_1, ..., x_N \mid \mathrm{y})}{P(x_1, ..., x_N)} \tag{21}$$

Através da hipótese *naive* define-se que (ZHANG, H., 2004b)

$$P(x_i \mid \mathrm{y}, x_1, ..., x_{i-1}, x_{i+1}, ..., x_N) = P(x_i \mid \mathrm{y}) \tag{22}$$

A principal diferença entre os diferentes classificadores do método Naive Bayes é relacionada com as hipóteses feitas sobre as distribuições de $P(x_i \mid \mathrm{y})$. O clasificador Naive Bayes Gaussiano considera como gaussiana a distribuição, enquanto que o classificador Naive Bayes Multinomial considera que os dados são distribuidos multinomialmente (PEDREGOSA *et al.*, 2011).

3.2.7 COMPARAÇÃO ENTRE CLASSIFICADORES

Diversos métodos de validação estatística podem ser utilizados a fim de comprovar os resultados obtidos e, assim, selecionar o modelo com melhor desempenho para o conjunto de dados de interesse de acordo com a métrica de interesse. Esta seção detalha os fundamentos teóricos dos métodos empregados neste trabalho.

3.2.7.1 VALIDAÇÃO CRUZADA *K-FOLD*

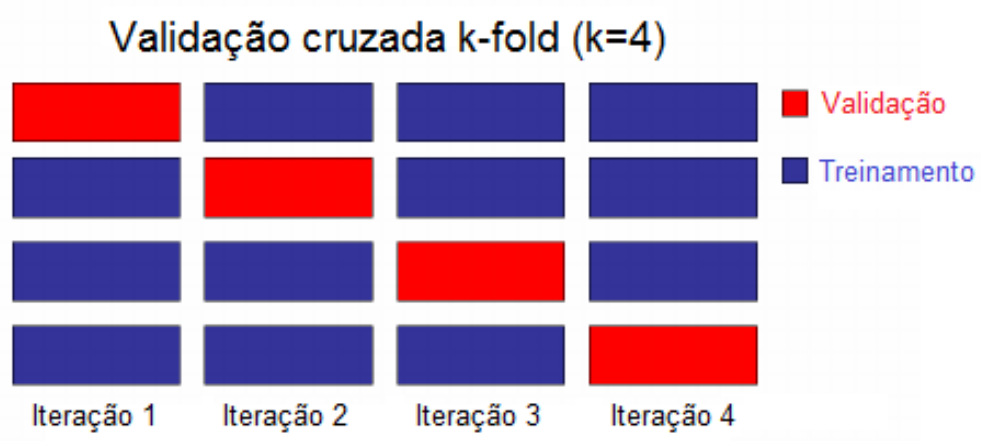
Como observado por Nilsson (1998), quando o treinamento e avaliação de um algoritmo são realizados utilizando o mesmo conjunto de dados, obtém-se um resultado demasiadamente otimista a respeito de seu desempenho. A validação cruzada apresenta uma

alternativa para resolver este problema, partindo do princípio de que o poder preditivo de um algoritmo deve ser testado em dados que não foram utilizados em seu treinamento (ARLOT; CELISSE, 2010).

Na maioria das aplicações, existe um número limitado de dados disponíveis. Assim, criou-se o conceito da divisão dos dados: parte dos dados são utilizados para treinar o algoritmo (instâncias de treinamento) e o restante dos dados (instâncias de teste) são utilizados para avaliar o desempenho do algoritmo, desde que todas as instâncias sejam independentes e identicamente distribuídas. Assim, ao se realizar esta divisão entre treinamento e teste diversas vezes, obtém-se uma estimativa mais robusta do poder de generalização de um classificador, sendo possível utilizar esta avaliação no resultado de diversos algoritmos para fins de comparação (ARLOT; CELISSE, 2010).

Na validação cruzada *k-fold* (*k-fold cross-validation*), o conjunto de dados original é particionado aleatoriamente em k subconjuntos com dimensões idênticas ou semelhantes. Dentre os k subconjuntos, um é definido como o subconjunto de teste, enquanto que os demais são utilizados como os subconjuntos de treinamento. Este processo é repetido k vezes, sendo cada subconjunto utilizado apenas uma vez como dados de teste para a validação. A média dos k resultados representa uma estimativa única e mais robusta do desempenho do classificador. Em aplicações de *machine learning*, é comum a utilização de 10 *folds* ($k = 10$). Alguns autores sugerem ainda o uso de validação cruzada repetida, estratégia na qual a validação cruzada *k-fold* é repetida N vezes, cada qual com uma configuração distinta (em essência, composição em termos de instâncias) para os *k-folds* gerados no início do processo (REFAEILZADEH; TANG; LIU, 2008).

Figura 4 - Exemplo de validação cruzada k-fold com k = 4.



Fonte: Adaptado de (BISGIN *et al.*, 2011).

3.2.7.2 VALIDAÇÃO CRUZADA *K-FOLD* ESTRATIFICADA

Para cenários que apresentam desbalanço de classes ou um conjunto limitado de dados para treinamento de modelos, uma variação da validação cruzada denominada validação cruzada *k-fold* estratificada é recomendada. Esta variação consiste em garantir que a divisão dos subconjuntos (*k-folds*) seja realizada de forma que os mesmos representem adequadamente a mesma distribuição de classes do conjunto original de dados, ou seja, apresentem a mesma proporção de instâncias por classe observada para o conjunto total de instâncias (REFAEILZADEH; TANG; LIU, 2008).

3.2.8 TEOREMA *NO-FREE-LUNCH*

Para cada classificador, existe uma tarefa na qual o mesmo falhará, mesmo que esta tarefa seja desempenhada com êxito por outro classificador (SHALEV-SCHWARTZ; BEN-DAVID, 2014). Em termos gerais, este teorema diz que é impossível a construção de um algoritmo global (independente do conjunto de dados e do tipo de problema a ser resolvido) o qual apresenta alto desempenho independente da métrica analisada, onde "alto" desempenho é fortemente baseado no contexto da aplicação (OMEZ; ROJAS, 2016). A implicação do Teorema *No-Free-Lunch* na área de *machine learning* é a necessidade de se testar e

experimentalmente validar múltiplos classificadores para um problema particular, a fim de identificar o que possui melhor potencial no cenário de interesse.

3.2.9 DESBALANÇO DE CLASSES

Um conjunto de dados é denominado desbalanceado caso contenha ao menos uma classe representada por uma quantidade demasiadamente reduzida de instâncias quando comparada às demais classes. Nestas situações, classificadores podem apresentar acurácia alta com relação à classe majoritária e ao mesmo tempo apresentar uma acurácia extremamente inferior com relação à classe minoritária. Isto decorre da influência que a classe majoritária exerce nos critérios tradicionais de treinamento, visto que a maioria dos classificadores baseia-se na diminuição da taxa de erro em suas previsões, assumindo que todas as classificações errôneas custam o mesmo. Assim, uma avaliação de desempenho baseada exclusivamente na acurácia pode prover medidas enviesadas do poder preditivo de um classificador para dados desbalanceados. Porém, em diversas aplicações reais, esta hipótese não se confirma, como, por exemplo, em aplicações de diagnósticos médicos (GANGANWAR, 2012).

O estudo de métodos de *machine learning* utilizando conjunto de dados desbalanceados com classes binárias é um tópico popular e conhecido, porém o estudo de problemas multiclasses ainda é um problema de pesquisa em aberto (KOÇO *et al.*, 2013).

3.3 INDICADORES DE DESEMPENHO

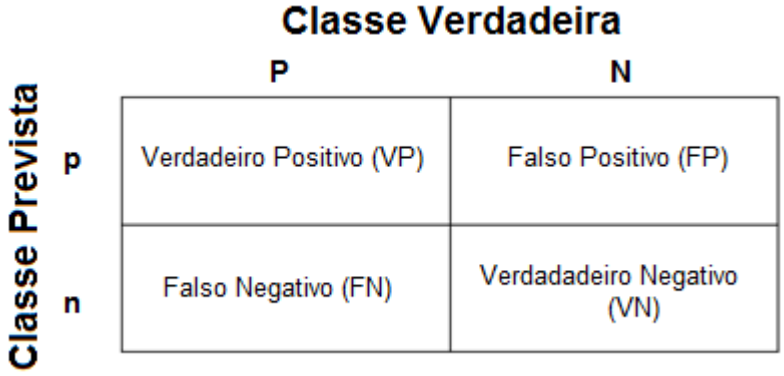
Diversos indicadores de desempenho podem ser utilizados a fim de estimar e comparar os resultados de diferentes classificadores, comparando a saída predita com a saída esperada. Esta seção detalha os indicadores utilizados neste trabalho. Deve-se ter em mente que todas as

métricas apresentadas se comportam de maneira diferente quando utilizadas em conjuntos de dados balanceados ou desbalanceados (SAITO; REHMSMEIER, 2015).

3.3.1 DEFINIÇÕES GERAIS UTILIZADAS NOS INDICADORES DE DESEMPENHO

A divisão das previsões em uma matriz de confusão binária pode ser observada na Figura 5.

Figura 5 - Definição dos agrupamentos de resultados em uma matriz de confusão para classificação binária.



Para problemas multiclasse, também é possível compilar os resultados em uma matriz de confusão, como mostra a Figura 6.

Figura 6 - Matriz de confusão para problema com 3 classes.

Previsão / Classe Verdadeira	A	B	C
A	AA	AB	AC
B	BA	BB	BC
C	CA	CB	CC

Fonte: Adaptado de (LI, L.; WU; YE, 2015).

Neste caso, os eventos AA representam as instâncias corretamente previstas como pertencentes à classe A, enquanto o mesmo é válido para os eventos BB e CC, sendo estes respectivos às classes B e C, enquadrando-se na definição de verdadeiros positivos. Os eventos BA e CA são os falsos positivos referentes à classe A. A mesma interpretação é

realizada para os eventos BB e CB, e BC e CC, para as classes B e C, respectivamente. A avaliação de métricas como precisão e *recall* é feita por classe e depois sumarizadas por algum tipo de média. A acurácia e taxa de erro são calculadas de forma global.

3.3.2 ACURÁCIA

A acurácia (PAULINO *et al.*, 2011), apresentada na Equação (23), é a proporção dos verdadeiros positivos com relação ao total de previsões realizadas,

$$A = \frac{VP + VN}{VP + FP + VN + FN} \tag{23}$$

onde *A* é a acurácia, *VP* é o total de verdadeiros positivos, *VN* é o total de verdadeiros negativos, *FP* é o total de falsos positivos e *FN* é o total de falsos negativos (SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006).

3.3.3 TAXA DE ERRO

A taxa de erro, apresentado na Equação (24), é a proporção de instâncias classificadas incorretamente pelo modelo, isto é, o número de falsos positivos e falsos negativos com relação ao total de previsões realizadas,

$$TE = \frac{FP + FN}{VP + VN + FP + FN} \tag{24}$$

onde *TE* é a taxa de erro (SAITO; REHMSMEIER, 2015). A taxa de erro também pode ser calculada a partir da acurácia, como mostra a Equação (25).

$$TE = 1 - A \tag{25}$$

3.3.4 PRECISÃO

A precisão *P*, apresentada na Equação (26), é a proporção dos verdadeiros positivos com relação ao total de verdadeiros positivos e falsos positivos (SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006).

$$P = \frac{VP}{VP + FP} \tag{26}$$

3.3.5 RECALL

O *recall R*, apresentado na Equação (27), é a proporção dos verdadeiros positivos com relação ao total de verdadeiros positivos e falsos negativos, refletindo a completude do modelo quanto à identificação dos exemplos positivos.

$$R = \frac{VP}{VP + FN} \tag{27}$$

3.3.6 *F1-SCORE*

O *F1-score* (F_1), apresentado na Equação (28), representa uma média harmônica entre precisão e recall, refletindo em uma única medida o desempenho do modelo em predizer corretamente todas as instâncias verdadeiramente positivas e evitar a predição de falsos positivos (SAITO; REHMSMEIER, 2015).

$$F_1 = \frac{2\,P\,R}{P+R} \tag{28}$$

3.4 SELEÇÃO DE *FEATURES*

Seleção de *features* é o processo de identificar e remover *features* irrelevantes e redundantes do conjunto de dados utilizado (YU; LIU, 2004). A seleção de *features* pode também ser utilizada para reduzir a dimensão dos dados analisados e auxiliar na velocidade e

eficácia dos algoritmos de mineração de dados. O fato de que diversas *features* são interdependentes pode afetar o desempenho de classificadores baseados em aprendizado supervisionado. Este problema pode ser resolvido através da construção de novas *features* a partir do conjunto básico de *features* presente no conjunto de dados original (MARKOVITCH; ROSENSTEIN, 2002).

4 REVISÃO DE LITERATURA

Neste capítulo será detalhada a revisão de literatura relacionada com o domínio estudado.

4.1 PREMIER LEAGUE

A *Premier League* é uma liga profissional de futebol realizada na Inglaterra onde 20 times competem pelo título. Os últimos três colocados de cada temporada são rebaixados e substituídos pelos três melhores clubes de ligas inferiores. Na *Premier League*, cada clube joga contra todos os outros duas vezes: uma vez em seu estádio e uma vez no estádio do adversário. Assim, a *Premier League* apresenta 38 rodadas, com 10 partidas por rodada, totalizando 380 partidas. Uma temporada normal da *Premier League* é iniciada em Agosto e finalizada em Maio do ano seguinte (TIMMARAJU; PALNITKAR; KHANNA, 2013).

4.2 RANKINGS ESPORTIVOS

De acordo com (LASEK; SZLÁVIK; BHULAI, 2013), classificar diferentes participantes de eventos esportivos é importante pois existe uma necessidade de indicar a qualidade de cada um dos competidores baseando-se apenas em suas performances individuais.

Os principais modelos de classificação utilizados no futebol são variações do sistema ELO, originalmente utilizado no xadrez, mas modificado e utilizado amplamente para a classificação de indivíduos e grupos em diversos esportes.

4.2.1 SISTEMA ELO

O sistema de classificação ELO, criado pelo físico e enxadrista húngaro Arpad Emrick Elo, é um classificador amplamente utilizado em esportes. De acordo com (LASEK;

SZLÁVIK; BHULAI, 2013), neste sistema classificatório a pontuação de um competidor é atualizada iterativamente após cada evento. Assim, a pontuação atualizada de um competidor é a atualização da sua pontuação antiga baseada no resultado da partida e na expectativa anterior à partida (resultado esperado). A Equação (29) apresenta o cálculo da pontuação de um indivíduo após uma partida,

$$R_A' = R_A + K\,(S_A - P_A) \tag{29}$$

onde R_A' é a pontuação atualizada do Competidor A, R_A é a pontuação antiga do Competidor A, S_A é o resultado real da partida (da perspectiva do Competidor A contra o Competidor B), P_A é o resultado esperado (baseado nas pontuações de ambos os competidores antes da partida) e K é uma constante positiva definida pela importância da partida. S_A assume o valor 0 caso o Competidor A seja derrotado pelo Competidor B, 0.5 em caso de empate e 1 em caso de vitória do Competidor A sobre o Competidor B. O sistema classificatório ELO original assume que o desempenho dos competidores segue uma distribuição normal com média R_A. Porém, ainda de acordo com os autores, é comum assumir que o desempenho dos competidores segue uma distribuição logística. Assim, P_A é calculado de acordo com a Equação (30),

$$P_A = \frac{1}{1 + e^{-a\,(R_A + R_B)}} \tag{30}$$

onde a é um fator de escalonamento.

A base do sistema de classificação ELO é a existência de uma autocorreção na pontuação após uma partida. Por exemplo, um competidor que alcançar um resultado melhor do que o esperado receberá uma "bonificação" enquanto que um time que for derrotado por um adversário com desempenho inferior ao esperado receberá uma "penalização" em sua pontuação. Por fim, observa-se a importância da escolha do número de resultados passados na

primeira iteração deste classificador. Uma prática comum, de acordo com os autores, é inicializar a pontuação de todos os competidores com o mesmo valor, visto que o importante é a diferença entre as pontuações e não os pontos em si.

4.2.1.1 SISTEMA ELO APLICADO AO FUTEBOL

O sistema de classificação ELO aplicado ao futebol é uma variação do sistema ELO clássico. De acordo com (FOOTBALLDATABASE.COM, 2017), esta variação leva em consideração o time mandante, a diferença de gols final de uma partida, a diferença de pontos (diferença de ELO) dos times e apresenta um ajuste de peso baseado na importância da partida. A Equação (31) descreve o cálculo da nova pontuação do Time A após uma partida contra o Time B,

$$R_A' = R_A + K\,G\,(W - W_E) \tag{31}$$

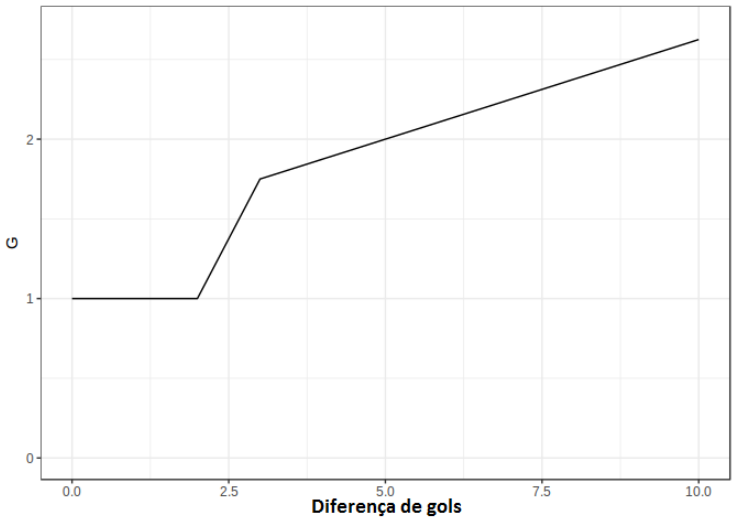
onde R_A' é a pontuação atualizada do Time A, R_A é a pontuação antiga do Time A, W é o resultado real da partida (da perspectiva do Time A contra o Time B), W_E é o resultado previsto, K é uma constante positiva definida pela importância da partida e G é um índice que traduz a diferença de gols de ambos os times na partida atual. W assume o valor 0 caso o Time A seja derrotado pelo Time B, 0.5 em caso de empate e 1 em caso de vitória do Time A sobre o Time B. G é calculado da seguinte maneira:

- Se a partida terminar empatada ou se um time vencer por apenas um gol de diferença, G = 1;
- Se um time vencer por dois gols de diferença, G = 1.5;
- Se um time vencer por mais de dois gols de diferença,

$$G = \frac{11 + N}{8} \tag{32}$$

onde *N* é a diferença de gols. É importante notar que *G* não depende do time analisado, sendo que o mesmo valor de *G* é utilizado para atualizar as pontuações tanto do Time A quando do Time B.

Figura 7 - G em função da diferença de gols.



Fonte: Adaptado de (LACY, 2017).

O resultado esperado W_E é um valor entre 0 e 1, onde 0.5 representa que um empate é esperado. A Equação (33) apresenta o cálculo de W_E,

$$W_E = \frac{1}{10^{-\frac{dr}{400}} + 1} \tag{33}$$

sendo *dr* a diferença entre as pontuações de ambos os times (com uma bonificação para o time mandante), definido pela Equação (34),

$$dr = R_A - (R_B + 100) \tag{34}$$

caso o Time B seja o mandante da partida. Assim, W_E é considerado como uma provável derrota caso resulte em um valor menor do que 0.5. Se W_E resultar em um valor maior do que 0.5, o resultado esperado é uma vitória.

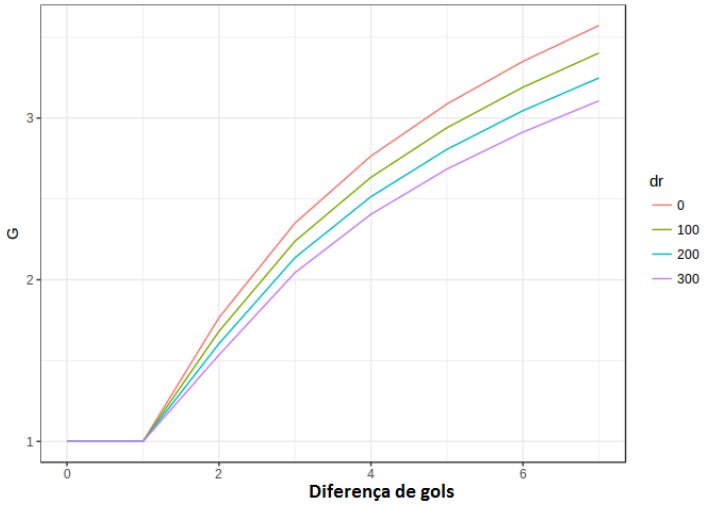
Outra forma de se calcular *G* é proposta por (LACY, 2017). Nesta variação, *G* segue uma curva logarítmica, bonificando mais expressivamente vitórias com diferenças de gols

menores quando comparado ao cálculo de *G* da Equação (32). O cálculo de *G* para esta variação é mostrado na Equação (35),

$$G = \log(1.7N)\frac{2}{2+0.001dr} \tag{35}$$

onde *N* é a diferença de gols. A Figura 8 mostra *G* em função da diferença de gols utilizando diferentes valores de *dr*.

Figura 8 - Curva de G em função da diferença de gols para vários valores de dr.



Fonte: Adaptado de (LACY, 2017).

4.3 MODELOS DE PREVISÃO DE PARTIDAS DE FUTEBOL

Esta seção detalha os modelos de previsão de partidas de futebol existentes na literatura estudada e a metodologia seguida nos respectivos modelos. As equações utilizadas para cálculo das *features* foram obtidas pelos respectivos autores de forma empírica. Assim, há margem para otimização das mesmas. Porém, neste trabalho optou-se pela utilização das mesmas em suas formas originais, conforme desenvolvidas pelos respectivos autores.

4.3.1 MODELOS DE PREVISÃO BASEADOS EM MACHINE LEARNING

Ulmer e Fernandez (2013) implementaram diversos métodos para tentar prever resultados da *Premier League* (campeonato de futebol da Inglaterra, equivalente a Série A brasileira). Para tanto, os autores utilizaram dados de 10 temporadas (3800 partidas) na etapa de treinamento dos modelos e 2 temporadas (760 partidas) na etapa de teste. Primeiramente, os autores criaram um *ranking* contendo todos os times analisados, classificando-os de maneira semelhante à utilizada no sistema ELO. Outra característica utilizada foi a "forma" atual dos times, que é baseada no desempenho do mesmo nas 7 últimas partidas (excluindo as primeiras 7 partidas de cada temporada da análise). Cada partida analisada para o cálculo da forma apresentou um peso diferente, sendo que a partida mais recente apresentou o maior peso e a partida mais antiga apresentou o menor peso. Além do desempenho dos times, os autores também levaram em consideração o local da partida, baseando-se na hipótese de que times mandantes possuem vantagem. Assim, dentre todos os modelos implementados pelos autores, os classificadores que apresentaram melhor desempenho foram *Support Vector Machine (SVM)*, *Random Forest (RF)* e *one-vs-all Stochastic Gradient Descent (SGD)*, com taxas de erro entre 48% e 52%. A Tabela 1, Tabela 2, Tabela 3 e a Tabela 4 apresentam a matriz de confusão de cada um destes classificadores.

Tabela 1 - Matriz de confusão do classificador SVM linear.

	Previsão de empate	**Previsão de vitória**	**Previsão de derrota**
Empate	0	142	152
Vitória	0	307	162
Derrota	0	152	317

Tabela 2 - Matriz de confusão do classificador SVM com *kernel* RBF (*Radial Basis Function*).

	Previsão de empate	**Previsão de vitória**	**Previsão de derrota**
Empate	42	126	126
Vitória	34	275	160
Derrota	40	136	293

Tabela 3 - Matriz de confusão do classificador RF.

	Previsão de empate	Previsão de vitória	Previsão de derrota
Empate	12	142	140
Vitória	13	302	154
Derrota	23	134	312

Tabela 4 - Matriz de confusão do classificador SGD.

	Previsão de empate	Previsão de vitória	Previsão de derrota
Empate	3	91	200
Vitória	12	249	208
Derrota	12	68	389

Como pode ser observado, todos os classificadores utilizados pelos autores apresentaram um desempenho inferior na previsão de empates quando comparado às demais classe, justificado pela menor incidência deste resultado em jogos da Premier League. Esta mesma limitação foi observada no trabalho de Trindade (2013), o qual aplicou os métodos *Maximum Likelihood Estimator (MLE)* e *Multilayer Perceptron (MP)* para previsão de resultados em jogos da Série A do Campeonato Brasileiro, obtendo taxas de erro entre 81.63% e 77.55% para a classe "Empate". O autor implementou seu modelo baseando-se em um vetor de *features* contendo informações sobre partidas em um intervalo de 10 anos do Campeonato Brasileiro Série A (2003-2013). Assim, o vetor criado pelo autor leva em consideração o local da partida (para determinar o time mandante), o resultado da partida (vitória, empate ou derrota) e o ano, ponderando assim os resultados em função do tempo, dando maior peso para as partidas mais recentes. A Tabela 5 e a Tabela 6 apresentam a matriz de confusão obtida pelo autor para cada classificador.

Tabela 5 - Matriz de confusão do classificador MLE.

	Previsão de empate	Previsão de vitória	Previsão de derrota
Vitória	8	71	18
Empate	9	25	15
Derrota	4	19	21

Tabela 6 - Matriz de confusão do classificador MP.

	Previsão de empate	Previsão de vitória	Previsão de derrota
Vitória	7	78	12
Empate	11	27	11
Derrota	4	23	17

Joseph, Fenton e Neil (2006) realizaram uma comparação entre Redes Bayesianas (RB) e diversas metodologias de aprendizado de máquina (MC4*,* Árvores de Decisão*, Naive Bayes, Data Driven Bayesian e K-Nearest Neighbors*). Os dados utilizados compreendem todas as partidas do clube Tottenham Hotspur Football Club entre 1995 e 1997. Os autores utilizaram como *features* a presença (ou ausência) e a posição de alguns jogadores importantes da equipe, a qualidade do adversário e o local da partida. Os resultados encontrados pelos autores mostraram que as redes Bayesianas, quando estas são construídas utilizando um domínio de características selecionado adequadamente, apresentaram menor taxa de erro (40.79%) quando comparadas com outros métodos de aprendizado de máquina.

Aslan e Inceoglu (2007) chegaram à conclusão de que deve-se estudar o problema da previsão de resultados em jogos de futebol com base no fato de que existe uma maior incidência de resultados positivos para times que jogam como mandante. O trabalho destes autores foi baseado na pesquisa de Cheng e colaboradores (2003), que propôs a utilização de um algoritmo de redes neurais híbrido baseado em *Back Propagation (BP)* e *Learning Vector Quantization (LVQ)*, utilizando a temporada 2001-2002 do campeonato Serie A Italiano para testar a acurácia das previsões do modelo. As características utilizadas como *features* das redes neurais com BP foram:

- Razão de vitórias: razão entre a quantidade de partidas que um determinado time venceu e o total de partidas. O mesmo conceito foi aplicado para empates e derrotas;

- Média de gols: total de gols de um determinado time dividido pelo total de partidas. O mesmo conceito foi aplicado para gols concedidos;
- Forma: a forma de um time foi calculada de acordo com a Equação (36),

$$\text{Forma} = 3R(n-1) + 2R(n-2) + R(n-3) \tag{36}$$

onde *R(n-i)* representa o resultado do partida i, sendo *R(n)* a partida atual. Caso o time tenha vencido a partida *n-i*, *R(n-i)* será 3. Em caso de empate, *R(n-i)* será 1 e em caso de derrota, *R(n-i)* será 0.

- Time mandante e time visitante.

Na etapa de treinamento, o modelo previu corretamente 73% das partidas analisadas, sendo que os autores consideraram as demais partidas como resultados incomuns, popularmente conhecido como "zebra" no Brasil. Por fim, os autores testaram o modelo utilizando 153 jogos, compreendidos entre as rodadas 18 e 34 da Serie A italiana. A taxa de erro do modelo implementado foi de 48%. Este resultado foi comparado com outras metodologias, utilizadas como *baseline*, como a previsão baseada no modelo ELO, que apresentou taxa de erro de cerca de 53%. A previsão baseada apenas na razão entre gols concedidos e marcados apresentou taxa de erro de cerca de 51%. As previsões baseadas nas análises comparativas das últimas 6 partidas de cada time apresentaram taxa de erro de cerca de 66%.

Por fim, uma importante conclusão levantada por Lasek, Szlávik e Bhulai (2013) diz respeito à superioridade de um *ensemble* de classificadores quando comparados com a utilização de algoritmos individuais.

A Tabela 7 apresenta um resumo dos principais métodos discutidos nesta seção com suas respectivas características e resultados.

Tabela 7 - Resumo dos métodos e resultados encontrados pelos autores estudados nas etapas de testes.

Autor	Método	Partidas	Acurácia (%)
(ULMER; FERNANDEZ, 2013)	SVM Linear	760	51
	One-vs-all SGD	760	52
	One-vs-all SGD	342	51
	One-vs-all SGD	266	49
	One-vs-all SGD	190	48
	One-vs-all SGD	114	49
	SVM RBF	760	48
	Random Forest	760	50
(JOSEPH; FENTON; NEIL, 2006)	Rede Bayesiana Expert	114	59
(CHENG *et al.*, 2003)	Rede LVQ + BP	153	52
	ELO clássico	153	47
	Razão entre gols	153	49
(TRINDADE, 2013)	MLE	190	56
	MP	190	54

4.3.2 MODELO DE PREVISÃO BASEADO NA EXPECTATIVA DE GOLS

Um modelo clássico de previsão de resultados é o modelo de gols esperados. Este modelo de previsão foi inicialmente estudado por Mahrer (1982), que adotou um modelo de Poisson independente com média variável baseada na qualidade dos times. Assim, considerando que o time mandante i enfrenta o time visitante j e o placar observado é (x_{ij},y_{ij}), pode-se assumir que X_{ij} segue uma distribuição de Poisson com média $\alpha_i\beta_j$. Por consequência, Y_{ij} também segue uma distribuição de Poisson com média $\gamma_i\delta_j$, considerando-se X_{ij} e Y_{ij} independentes. Pode-se considerar α_i como a força ofensiva do time i quando o mesmo joga como mandante, β_j como a força defensiva do time j quando o mesmo joga como visitante, γ_i como a força defensiva do time i quando o mesmo joga como mandante e δ_j como a força ofensiva do time j quando o mesmo joga como visitante.

Uma variação do modelo baseado na expectativa de gols é utilizado pela casa de apostas online Pinnacle, uma das maiores empresas do ramo. De acordo com (CRONIN, 2017), é possível calcular a força ofensiva do time i a partir da Equação (37),

$$\alpha_i = \frac{\frac{GM_i}{TPM_i}}{\frac{TGM}{TP}} \tag{37}$$

onde GM_i é o total de gols do time i como mandante, TPM_i é o total de partidas do time i como mandante, TGM é o total de gols de todos os mandantes no campeonato analisado e TP é o total de partidas do campeonato até o instante de interesse. De maneira similar, pode-se definir a força defensiva do time j através da Equação (38)

$$\beta_j = \frac{\frac{GSV_j}{TPVj}}{\frac{TGSV}{TP}} \tag{38}$$

onde GSV_j é o total de gols sofridos pelo time j como visitante, TPV_j é o total de partidas do time j como visitante, $TGSV$ é o total de gols sofridos por todos os visitantes no campeonato analisado e TP é o total de partidas do campeonato até o instante de interesse. O cálculo dos coeficientes γ_i e δ_j é realizado de forma similar.

Assim, de acordo com o autor, pode-se calcular a quantidade provável de gols que o time i marcará através da Equação (39),

$$EG_i = \alpha_i \, \beta_j \frac{TGM}{TP} \tag{39}$$

onde EG_i é a expectativa de gols do time i ao enfrentar o time j. Por consequência, pode-se calcular a expectativa de gols do time j de maneira similar,

$$EG_j = \gamma_i \, \delta_j \frac{TGV}{TP} \tag{40}$$

onde EG_j é a expectativa de gols do time j ao enfrentar o time i e TGV é o total de gols de todos os visitantes do campeonato analisado até o instante de interesse. Por fim, pode-se definir a distribuição de probabilidade para k e n gols (para cada time) aplicando as

respectivas expectativas de gols como valores médios da distribuição de Poisson, como mostra a Equação (41) e a Equação (42),

$$f_i\ (\mathrm{k};\mathrm{EG_i}) = \frac{e^{-\mathrm{EG_i}}\ \mathrm{EG_i}^{\mathrm{k}}}{\mathrm{k!}} \tag{41}$$

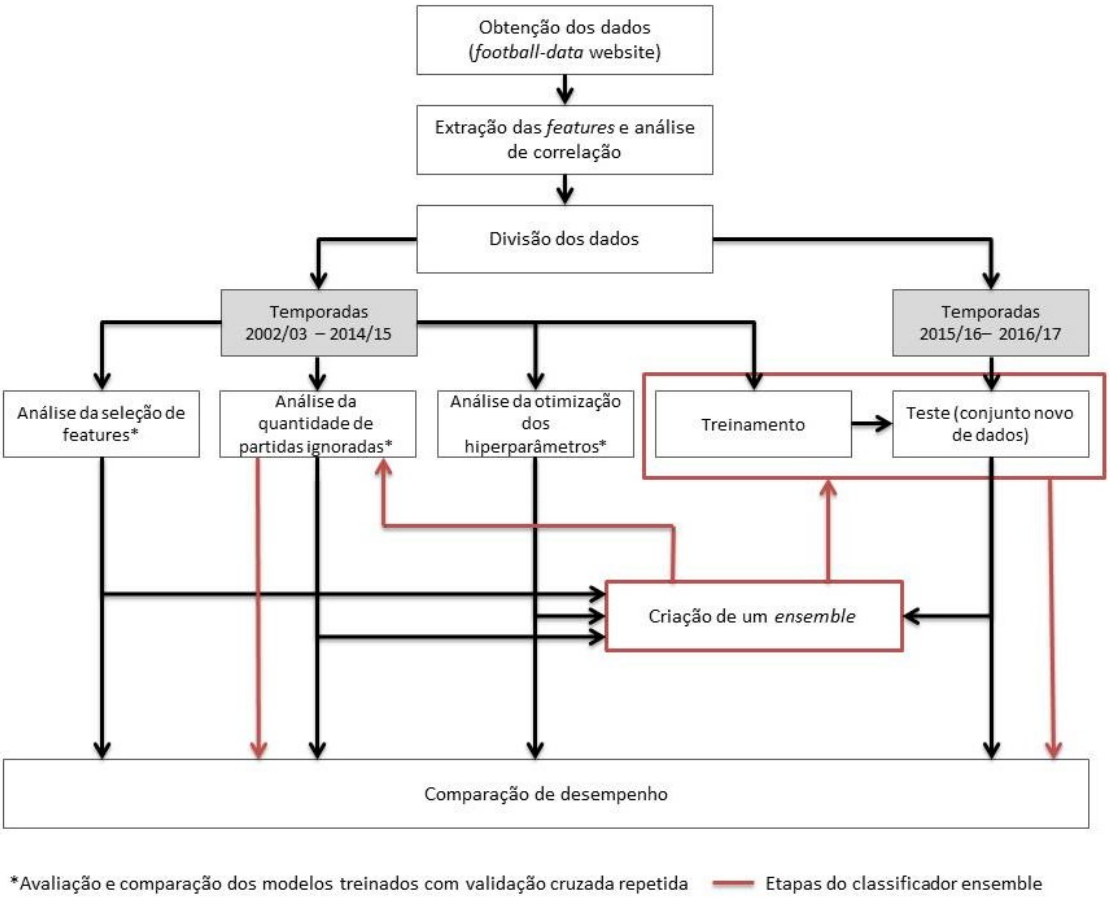
$$f_j\ (\mathrm{n};\mathrm{EG_j}) = \frac{e^{-\mathrm{EG_j}}\ \mathrm{EG_j}^{\mathrm{n}}}{\mathrm{n!}} \tag{42}$$

Definindo-se um intervalo de gols para *n* e *k*, pode-se agrupar as probabilidades dos eventos onde $k > n$ (probabilidade do time *i* ganhar a partida), $k < n$ (probabilidade do time *j* ganhar a partida) e $k = n$ (probabilidade de um empate ocorrer).

5 METODOLOGIA

Este capítulo apresenta a metodologia utilizada no presente trabalho. A Figura 9 apresenta um fluxograma que resume as etapas a seguir.

Figura 9 - Fluxograma da metodologia utilizada.



5.1 CONJUNTO DE DADOS E FERRAMENTAS UTILIZADAS

O conjunto de dados utilizado compreende todas as partidas realizadas no período de 2000 a 2017 no campeonato britânico *Premier League*, totalizando 6460 partidas. Estes dados foram obtidos gratuitamente no website *football-data.co.uk*. Para cada partida, o conjunto de dados utilizado apresenta detalhes como os times envolvidos (time mandante e visitante), o resultado final, o local da partida, o total de chutes no alvo, total de chutes em geral, total de cartões, escanteios e faltas.

Toda a codificação foi implementada em Python, utilizando bibliotecas adicionais como a biblioteca pandas (leitura de arquivos do tipo CSV), numpy (cálculos em geral), matplotlib (gráficos) e scikit-learn (*machine learning*) (PEDREGOSA *et al.*, 2011), entre outras.

5.2 EXTRAÇÃO DE *FEATURES*

A partir dos dados coletados foram extraídas *features* a serem utilizadas no treinamento dos classificadores selecionados. Utilizaram-se 21 *features*, descritas a seguir:

1. ELO do time mandante;
2. ELO do time visitante;
3. Forma do time mandante;
4. Forma do time visitante;
5. Probabilidade de Poisson de vitória do time mandante;
6. Probabilidade de Poisson de vitória do time visitante;
7. Probabilidade de Poisson de empate;
8. Força ofensiva do time mandante;
9. Força defensiva do time mandante;
10. Força ofensiva do time visitante;
11. Força defensiva do time visitante;
12. Média de vitórias do time mandante jogando como mandante;
13. Média de empates do time mandante jogando como mandante;
14. Média de vitórias do time visitante jogando como visitante;
15. Média de empates do time visitante jogando como visitante;
16. Média de gols marcados pelo time mandante jogando como mandate;
17. Média de gols sofridos pelo time mandante jogando como mandante;

18. Média de gols marcados pelo time visitante jogando como visitante;
19. Média de gols sofridos do time visitante jogando como visitante.
20. Expectativa de gols do time mandante.
21. Expectativa de gols do time visitante

As médias de gols, de vitórias e empates foram escolhidas de acordo com o conhecimento do esporte. As demais *features* foram escolhidas devido à sua ampla utilização na literatura.

Nenhuma *feature* foi normalizada. Assim, deve-se observar que classificadores como a Regressão Logística e o *K-Nearest Neighbors* podem sofrer alterações expressivas em suas métricas de desempenho devido a este fato.

Analisou-se também a correlação de Pearson entre as features, de forma a evitar a utilização de features fortemente correlacionadas..

5.2.1 ELO DO TIME MANDANTE E VISITANTE

Optou-se por utilizar a variação do ELO aplicado ao futebol, detalhada na Seção 4.2.1.1, aliada à variação do cálculo de *G* proposta por Lacy (2017). Adicionalmente, optou-se pela inicialização de todos os ELOs dos times participantes da primeira temporada analisada (2000/2001) com o valor 1500. Para garantir que o ELO dos times analisado refletirá a sua qualidade, a temporada 2000/2001 e a temporada 2001/2002 (760 partidas) não foram utilizadas nas etapas de validação cruzada, treinamento e teste dos classificadores, sendo utilizadas apenas para computar os ELOs e aproximá-los de valores que retratem mais fielmente a qualidade dos times.

Ao final de uma temporada, os ELOs de todos os times sofrem uma correção de forma a trazê-los de volta à média, de acordo com a Equação (43),

$$\mathrm{ELO_C} = 0.8\ \mathrm{ELO_{FT}} + 0.2 \cdot 1500 \tag{43}$$

onde ELO_C é o ELO corrigido e ELO_{FT} é o ELO original do time ao final da temporada.

No cálculo do ELO é importante manter a pontuação total (somatório de todos os ELOs) constante de forma a evitar a inflação/deflação das pontuações. Assim, um problema comum que surge no cálculo do ELO para um período maior do que uma temporada é relacionado com rebaixamentos e promoções. Na *Premier League*, os três times com o pior desempenho (em pontos) são rebaixados, dando lugar aos três melhores times da divisão inferior. Este fato faz com que alguma correção deva ser implementada, pois os times que foram rebaixados irão apresentar um ELO inferior a 1500, o que impossibilita dar aos times promovidos o valor inicial de 1500, como é feito na inicialização das pontuações.

Diversos autores propuseram metodologias para solucionar este problema, mas neste trabalho optou-se pelo desenvolvimento de uma metodologia própria. Para tanto, o ELO dos times rebaixados é somado e sua média é calculada, de acordo com a Equação (44),

$$\mathrm{ELO_R} = \frac{\mathrm{ELO_{18}} + \mathrm{ELO_{19}} + \mathrm{ELO_{20}}}{3} \tag{44}$$

onde ELO_R é a média dos ELOs dos três últimos colocados (times rebaixados) ao final de uma temporada, ELO_{18} é o ELO do décimo oitavo colocado ao final de uma temporada, ELO_{19} é o ELO do décimo nono colocado ao final de uma temporada e ELO_{20} é o ELO do vigésimo colocado. Assim, propõe-se atribuir a cada um dos times promovidos da divisão inferior o valor ELO_R como ELO inicial.

5.2.2 FORMA DO TIME MANDANTE E VISITANTE

Para o cálculo da forma atual de um time, a qual descreve de forma numérica o desempenho do time analisado com relação às últimas três partidas, optou-se por seguir a metodologia empregada por Cheng e colaboradores (2003), sendo o cálculo da forma descrito pela Equação (36) na Seção 4.3.1.

5.2.3 PROBABILIDADES DE POISSON

Para o cálculo da probabilidade de Poisson de vitória, empate e derrota de ambos os times, assim como para o cálculo da força ofensiva, força defensiva e expectativa de gols de ambos os times, será utilizada a metodologia descrita na Seção 4.3.2.

5.2.4 MÉDIA DE VITÓRIAS, EMPATES E DERROTAS COMO MANDANTE E VISITANTE

O cálculo da média de vitórias de um time específico jogando como mandante é mostrado na Equação (45), enquanto que a Equação (46) apresenta o cálculo da média de empates como mandante,

$$\mathrm{MV_M} = \frac{\mathrm{TVM}}{\mathrm{TPM}} \tag{45}$$

$$\mathrm{ME_M} = \frac{\mathrm{TEM}}{\mathrm{TPM}} \tag{46}$$

onde MV_M é a média de gols como mandante de um time, TVM é o total de vitórias como mandante, TPM é o total de partidas como mandante, ME_M é a média de empates como mandante e TEM é o total de empates como mandante. Os cálculos relativos ao time visitante são realizados de forma similar. A média de derrotas não foi utilizada por ser colinear com a média de vitórias.

5.2.5 MÉDIA DE GOLS MARCADOS E SOFRIDOS

O cálculo da média de gols marcados e sofridos por ambos os times é mostrado pela Equação (47), Equação (48), Equação (49) e Equação (50),

$$\mathrm{MGM_M} = \frac{\mathrm{TGM_M}}{\mathrm{TP_M}} \tag{47}$$

$$\mathrm{MGS_M} = \frac{\mathrm{TGS_M}}{\mathrm{TP_M}} \tag{48}$$

$$\mathrm{MGM_V} = \frac{\mathrm{TGM_V}}{\mathrm{TP_V}} \tag{49}$$

$$\mathrm{MGS_V} = \frac{\mathrm{TGS_V}}{\mathrm{TP_V}} \tag{50}$$

onde MGM_M é a média dos gols marcados por um time quando o mesmo joga como mandante, TGM_M é o total de gols marcados por um time quando o mesmo joga como mandante, TP_M é o total de partidas de um time como mandante, MGS_M é a média de gols sofridos de um time quando o mesmo joga como mandante, TGS_M é o total de gols sofridos de um time quando o mesmo joga como mandante, MGM_V é a média de gols marcados por um time quando o mesmo joga como visitante, TGM_V é o total de gols marcados por um time quando o mesmo joga como visitante, TP_V é o total de partidas de um time como visitante, MGS_V é a média de gols sofridos por um time quando o mesmo joga como visitante e TGS_V é o total de gols sofridos por um time como visitante.

5.3 CLASSIFICADORES UTILIZADOS

Esta seção detalha a como se realizou a comparação entre os diferentes classificadores utilizados. Utilizaram-se os classificadores descritos a seguir:

1. Regressão logística (RL);
2. Análise Discriminante Linear (ADL);
3. Análise Discriminante Quadrática (ADQ);
4. *K-Nearest Neighbors (KNN);*
5. *Naive* Bayes Gaussiano (NBG)*;*
6. *Naive* Bayes Multinomial (NBM);
7. *Support Vector Machine* com *kernel* linear (SVML);

8. *Support Vector Machine* com *kernel* RBF (SVMR);
9. *Random Forest* (RF);
10. *Extra Trees* (ET);

Estes classificadores foram escolhidos por representarem uma gama diversa de algoritmos de previsão, além de estarem disponíveis através da biblioteca *scikit-learn*.

5.4 COMPARAÇÃO DE DESEMPENHO ENTRE OS CLASSIFICADORES

Esta seção detalhará a metodologia de comparação das métricas propostas entre os diferentes classificadores.

5.4.1 COMPARAÇÃO EM FUNÇÃO DA QUANTIDADE DE PARTIDAS IGNORADAS

Através da Equação (36), que apresenta o cálculo da *feature* Forma, é possível concluir que deve-se ignorar no mínimo três partidas de maneira que a Forma possa ser calculada a cada nova temporada. Assim, analisou-se a quantidade de partidas ignoradas em função das métricas de desempenho dos classificadores, procurando observar a influência da variação das métricas de desempenho em função da quantidade de partidas ignoradas, tanto com uma configuração padrão dos hiperparâmetros dos classificadores (configuração padrão fornecida pela biblioteca *scikit-learn*) quanto com uma configuração otimizada para cada métrica analisada.

5.4.1.1 COMPARAÇÃO COM CONFIGURAÇÃO PADRÃO DE HIPERPARÂMETROS

Primeiramente comparou-se o desempemho (acurácia e *F1-score*[2]) dos múltiplos classificadores através do método de validação cruzada *k-fold* repetida, adotando-se 10 *folds* (k = 10) e 4 repetições (N = 4) na divisão dos dados, totalizando 40 amostras. Além disso,

[2] Para classes sem previsões o *F1-score* é definido como zero pela biblioteca *scikit-learn*.

utilizou-se a configuração padrão dos hiperparâmetros dos classificadores, de acordo com a biblioteca *scikit-learn*.

Nesta etapa de comparação, utilizaram-se as partidas compreendidas no período entre 2002/2003 – 2014/2015, totalizando 4940 partidas (13 temporadas). De forma a analisar a variação das métricas de interesse em função da quantidade de partidas ignoradas no início de cada temporada, comparou-se o desempenho dos classificadores ao se ignorar as primeiras 3 partidas (totalizando 4550 partidas analisadas), 10 partidas (totalizando 3640 partidas analisadas) e 19 partidas (totalizando 2470 partidas analisadas).

De forma a possibilitar a análise da existência de diferenças significativas entre os desempenhos dos classificadores em função da quantidade de partidas ignoradas, empregou-se o teste estatístico de Mann-Whitney.

5.4.1.2 OTIMIZAÇÃO DOS HIPERPARÂMETROS DOS CLASSIFICADORES

De forma a encontrar os hiperparâmetros os quais retornam as maiores métricas (acurácia e *F1-score*), variaram-se os hiperparâmetros dos classificadores de forma a encontrar a maior métrica de interesse, com exceção dos classificadores ADL, ADQ e NBG, visto que os mesmos não apresentam hiperparâmetros os quais possam ser otimizados. Além disso, deve-se notar que os classificadores foram otimizados para duas situações: acurácia e *F1-score*. Ou seja, existem dois conjuntos de hiperparâmetros otimizados para cada classificador os quais otimizam a métrica em questão.

Para otimização do classificador RL, analisou-se o desempenho sem nenhuma penalização, com a penalização L1 e com a penalização L2. Além disso, variou-se o valor de C no intervalo de 1 até 10000, buscando-se o valor máximo das métricas analisadas.

Para otimização do classificador SVML, variou-se o hiperparâmetro C no intervalo de 0.001 até 10000, buscando-se o valor máximo das métricas analisadas.

Para otimização do classificador SVML, variou-se o hiperparâmetro C no intervalo de 0.001 até 10000, procurando pela convergência das métricas analisadas. Já o hiperparâmetro γ foi analisado utilizando-se valores no intervalo 0.00001 até 0.1.

Para otimização do classificador KNN, variou-se a quantidade de vizinhos no intervalo de 1 até 10000, buscando-se o valor máximo das métricas analisadas.

Para otimização do classificador NBM, variou-se o valor de α no intervalo de 1 até 10000, buscando-se o valor máximo das métricas analisadas.

Para otimização do classificador RF, analisou-se o desempenho ao se utilizar o critério de Índice Gini e o critério da Entropia para divisão dos nodos. Além disso, variou-se a quantidade de *features* utilizadas em cada nodo no intervalo de 1 até 17 *features*, buscando-se o valor máximo das métricas analisadas.

Para otimização do classificador ET, analisou-se o desempenho ao se utilizar o critério de Índice Gini e o critério da Entropia para divisão dos nodos. Além disso, variou-se o valor de *features* no intervalo de 1 até 17, buscando-se o valor máximo das métricas analisadas.

5.4.1.3 COMPARAÇÃO COM CONFIGURAÇÃO DE HIPERPARÂMETROS OTIMIZADOS

Repetiu-se o procedimento descrito na Seção 5.4.1.1 Entretanto, utilizou-se a configuração otimizada dos hiperparâmetros, descrita na Seção 5.4.1.2, tanto com relação a acurácia quanto ao *F1-score*.

5.4.2 SELEÇÃO DE *FEATURES*

Utilizou-se o algoritmo *Random Forest* (com 10000 árvores e utilizando o critério de Índice Gini) para analisar a importância das *features* para a tarefa de classificação. O processo de seleção de *features* foi realizado com base em um processo iterativo, no qual após o cálculo da medida de relevância para cada *feature*, as *features* com menor importância são

retiradas uma a uma, repetindo-se os processos de validação e análise dos resultados de forma a analisar diferentes combinações de variáveis e suas relações com as métricas de desempenho dos classificadores. Este processo não utilizou validação cruzada, utilizando em vez disso todas as temporadas compreendidas entre a temporada 2002/2003 e 2014/2015 (incluindo ambas as temporadas), totalizando 2470 partidas (ignorou-se as primeiras 19 partidas de cada temporada). Nesta etapa, utilizou-se a configuração otimizada dos hiperparâmetros dos classificadores, tanto para a acurácia quanto para o *F1-score*, utilizando a respectiva configuração, visto que ambas as métricas foram analisadas. Esta etapa foi executada utilizando validação cruzada *k-fold* repetida com 10 *folds* ($k = 10$) e 4 repetições ($N = 4$), totalizando 40 amostras por classificador e por quantidade de *features* utilizada.

5.4.3 DESEMPENHO EM UM CONJUNTO DE DADOS NOVOS

De forma a se observar o comportamento dos classificadores em um conjunto de dados novos, utilizaram-se as temporadas entre 2002/2003 – 2014/2015 (2470 partidas, visto que ignoraram-se as primeiras 19 partidas de cada temporada nesta etapa) como conjunto de dados de treinamento e as temporadas 2015/2016 – 2016/2017 (380 partidas, ignorando-se também as primeiras 19 partidas de cada temporada) como conjunto de dados de teste. Assim, esta etapa não contou com a utilização do método de validação cruzada.

Nesta etapa, gerou-se a matriz de confusão normalizada das previsões de cada um dos classificadores, de forma a analisar as previsões por classe dos classificadores. Adicionalmente, observaram-se as métricas Precisão, *F1-score* e *Recall* por classe.

5.4.4 CRIAÇÃO E AVALIAÇÃO DO DESEMPENHO DE UM CLASSIFICADOR *ENSEMBLE*

Esta seção detalhará a escolha dos classificadores os quais constituirão um classificador *ensemble* e a metodologia de análise de desempenho do mesmo.

5.4.4.1 CRIAÇÃO DE UM CLASSIFICADOR *ENSEMBLE*

De forma a avaliar um classificador *ensemble* aplicado ao problema proposto, observou-se o desempenho geral dos classificadores utilizados nas análises anteriores, selecionando classificadores que apresentaram métricas consideradas satisfatórias (conforme explicado na Seção 6.3.4) para constituírem um classificador *ensemble* baseado no voto da maioria, sabendo que deve-se escolher classificadores que não realizem exatamente o mesmo tipo de previsão para as mesmas instâncias, visto que isto criaria um viés para a previsão do classificador *ensemble*. Os classificadores selecionados utilizaram configuração otimizada (para ambas as métricas, respectivamente).

5.4.4.2 DESEMPENHO DE UM CLASSIFICADOR *ENSEMBLE*

Repetiram-se os procedimentos detalhados na Seção 5.4.1.1 para o classificador *ensemble*. Adicionalmente, repetiram-se também os procedimentos descritos na Seção 5.4.3, possibilitando a análise das previsões do classificador *ensemble* quando confrontado com um conjunto de dados novos.

5.5 ANÁLISE ESTATÍSTICA DOS RESULTADOS

De forma a possibilitar a análise da existência de diferenças significativas entre as distribuições das instâncias previstas por diferentes classificadores, empregou-se do teste de Mann-Whitney, utilizando-se as previsões realizadas por meio de validação cruzada *k-fold* (k = 10) repetida (N = 4), totalizando 40 amostras em cada distribuição utilizada. Para fins de determinação da significância, utilizou-se um intervalo de confiança de 95% ($\alpha = 0.05$).

5.6 DEFINIÇÃO DE *BASELINES* PARA AVALIAÇÃO

Para avaliar e comparar os resultados obtidos através dos classificadores utilizados foram estabelecidas *baselines*. Para a primeira *baseline*, observou-se a quantidade de partidas por classe (vitória do mandante, vitoria do visitante e empate), de forma a utilizar a razão do número de instâncias da classe majoritária com relação ao total de instâncias como uma das *baselines* de comparação dos algoritmos. Assim, definiu-se como *baseline* a proporção de vitórias do time mandante. A segunda *baseline* utilizada foi definida como a proporção de vitórias do time com maior ELO. Estas *baselines* assumem um preditor naïve que sempre retorna como resultado uma vitória do time mandante ou uma vitória do time com maior ELO, respectivamente.

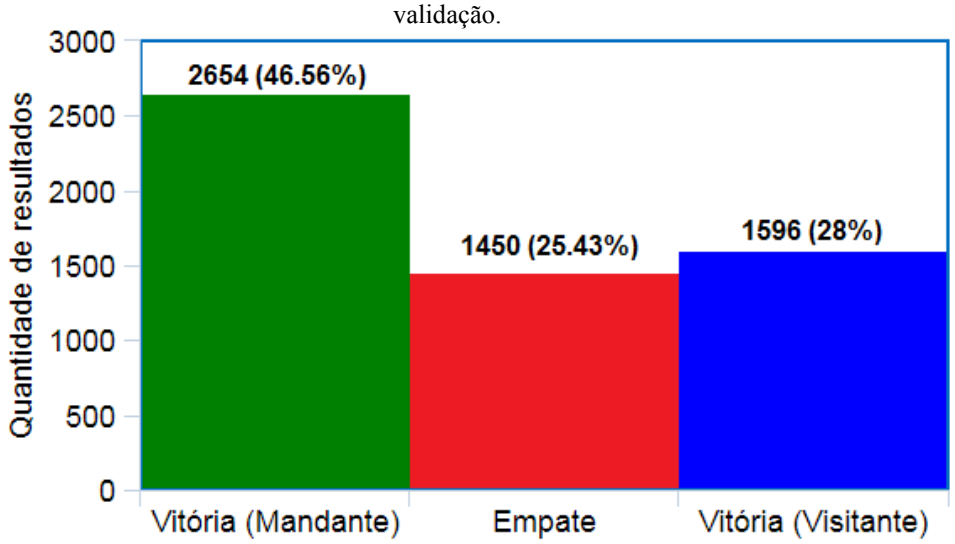
6 RESULTADOS

Esta seção detalhará os resultados encontrados seguindo a metodologia proposta.

6.1 *BASELINES* DE AVALIAÇÃO

A proporção de cada classe no conjunto de dados utilizado para treinamento e validação pode ser observada na Figura 10 - Proporção de resultados do conjunto de dados utilizado (Premier League) para treinamento e validação.

Figura 10 - Proporção de resultados do conjunto de dados utilizado (Premier League) para treinamento e validação.



Assim, a primeira *baseline* a ser considerada é a hipótese de sempre classificar o resultado de uma partida futura como sendo da classe Vitória (Mandante), visto que esta classe apresenta uma incidência maior quando comparada as demais no conjunto de dados utilizado (46.56% das instâncias). Esta *baseline* será denominada *baseline* Mandante.

A segunda *baseline* é baseada apenas no ELO. Observaram-se todos os eventos onde o ELO do time mandante era maior do que o elo do time visitante e, considerando todas as partidas de todas as temporadas utilizadas pelos classificadores (temporadas 2002/2003 – 2016/2017, incluindo ambas), observou-se que o time com maior ELO saiu vencedor de 50.24% das partidas realizadas. Assim, esta *baseline* será denominada *baseline* ELO.

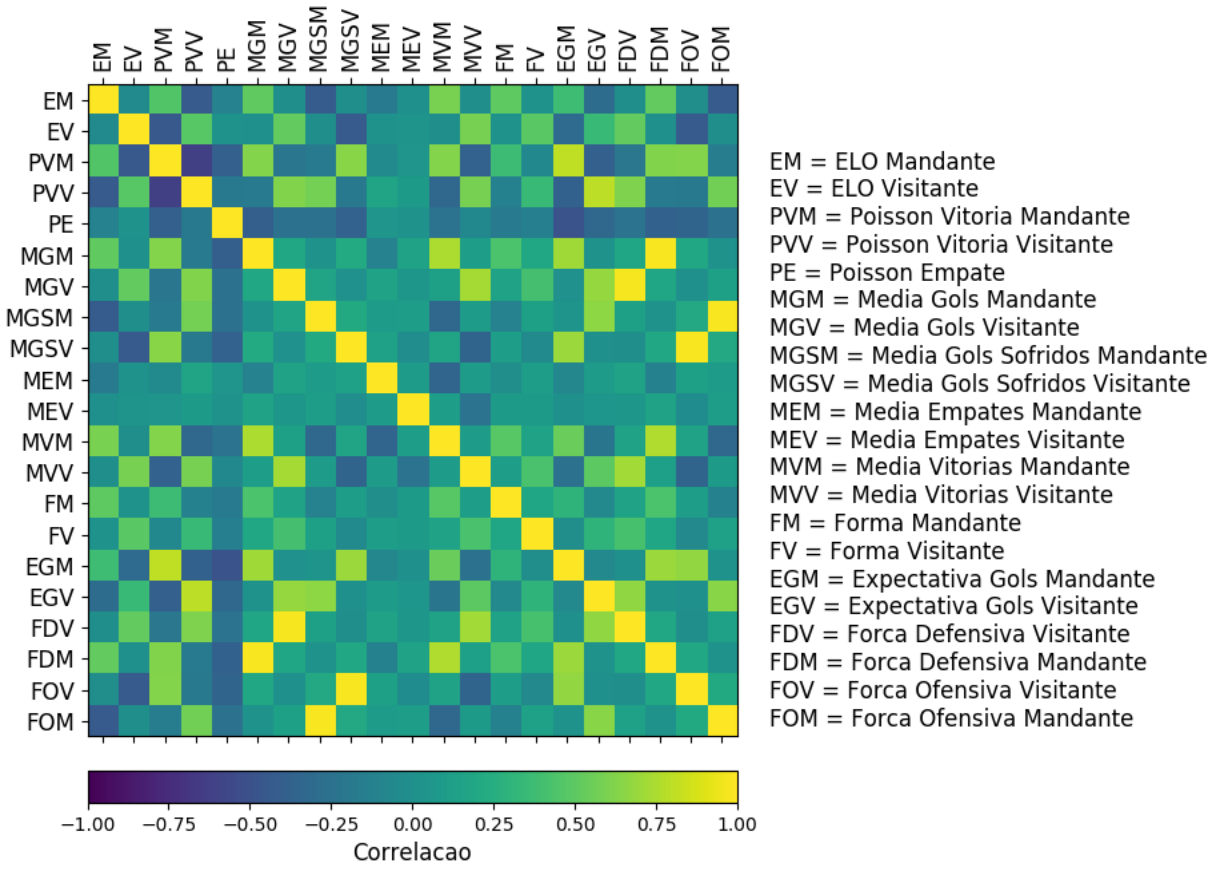
Estes valores reportados para ambas as *baselines* serão usados como referência para discussão da avaliação da acurácia dos classificadores empregados, a fim de averiguar se a taxa de acerto obtida com métodos de *machine learning* supera a taxa de acerto de classificadores *naïve*, cujas decisões são baseadas exclusivamente no time mandante ou no time com maior ELO da partida.

6.2 ANÁLISE DE CORRELAÇÃO DAS *FEATURES*

Nesta seção será detalhada a análise da correlação de Pearson entre as *features* utilizadas.

A Figura 11 apresenta a matriz de correlação (Pearson) entre as 21 *features* definidas.

Figura 11 - Matriz de correlação (Pearson) entre as *features*.



A Tabela 8 apresenta as correlações mais expressivas (acima de 0.7) entre as *features*.

Tabela 8 - Pares de *features* com as correlações de Pearson mais expressivas.

Par de *features*	Correlação (Pearson)
(FOM, MGM)	0.978
(FDV, MGSV)	0.977
(FDM, MGSM)	0.977
(FOV, MGV)	0.976
(EGM, PVM)	0.812
(EGV, PVV)	0.803
(FOM, MVM)	0.758
(MGM, MVM)	0.757
(MGV, MVV)	0.733
(FOV, MVV)	0.725
(EGM, MGM)	0.718
(EGM, FOM)	0.701

Observa-se que as médias de gols sofridos (MGSV e MGSM) e marcados (MGV e MGM) possuem uma correlação próxima de 1 com relação a força defensiva (FDV e FDM) e ofensiva (FOV e FOM). Assim, optou-se por retirar as forças defensivas e ofensivas do conjunto de *features* utilizado.

6.3 ANÁLISE DO DESEMPENHO DOS CLASSIFICADORES

Nesta seção será detalhada a análise do desempenho dos classificadores, baseando-se nas métricas definidas (acurácia e *F1-score*) utilizando-se da metodologia de validação cruzada *k-fold* repetida.

6.3.1 DESEMPENHO EM FUNÇÃO DA QUANTIDADE DE PARTIDAS IGNORADAS

Primeiramente, analisou-se o desempenho dos classificadores variando-se a quantidade de partidas ignoradas no início de cada temporada. Foram testadas e avaliadas três variantes: ignorando-se as primeiras 3 partidas, 10 partidas ou 19 partidas, cujos resultados são mostrados na Figura 12, Figura 13 e na Figura 14, respectivamente.

Figura 12 - (a) Acurácia e (b) *F1*-score dos classificadores ao se ignorar as primeiras 3 partidas de cada temporada. A linha azul representa a *baseline* ELO e a linha verde representa a *baseline* Mandante.

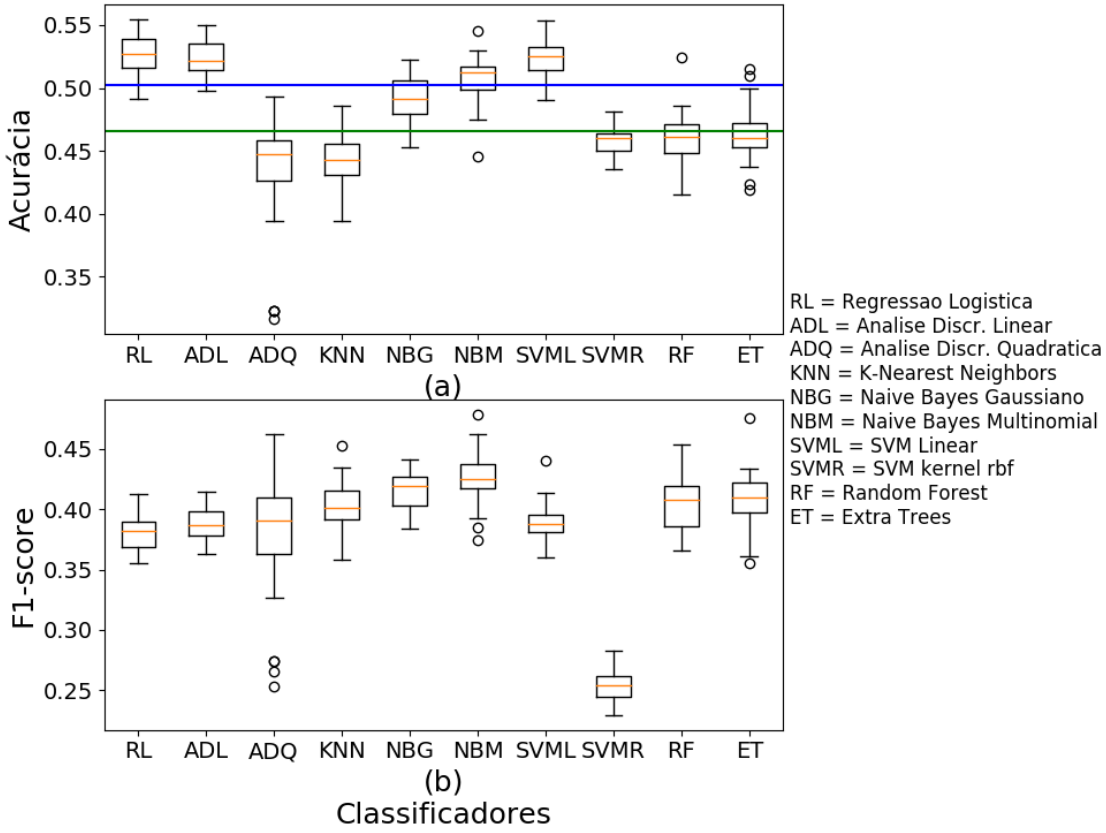


Figura 13 - (a) Acurácia e (b) *F1-score* dos classificadores ao se ignorar as primeiras 10 partidas de cada temporada. A linha azul representa a *baseline* ELO e a linha verde representa a *baseline* Mandante.

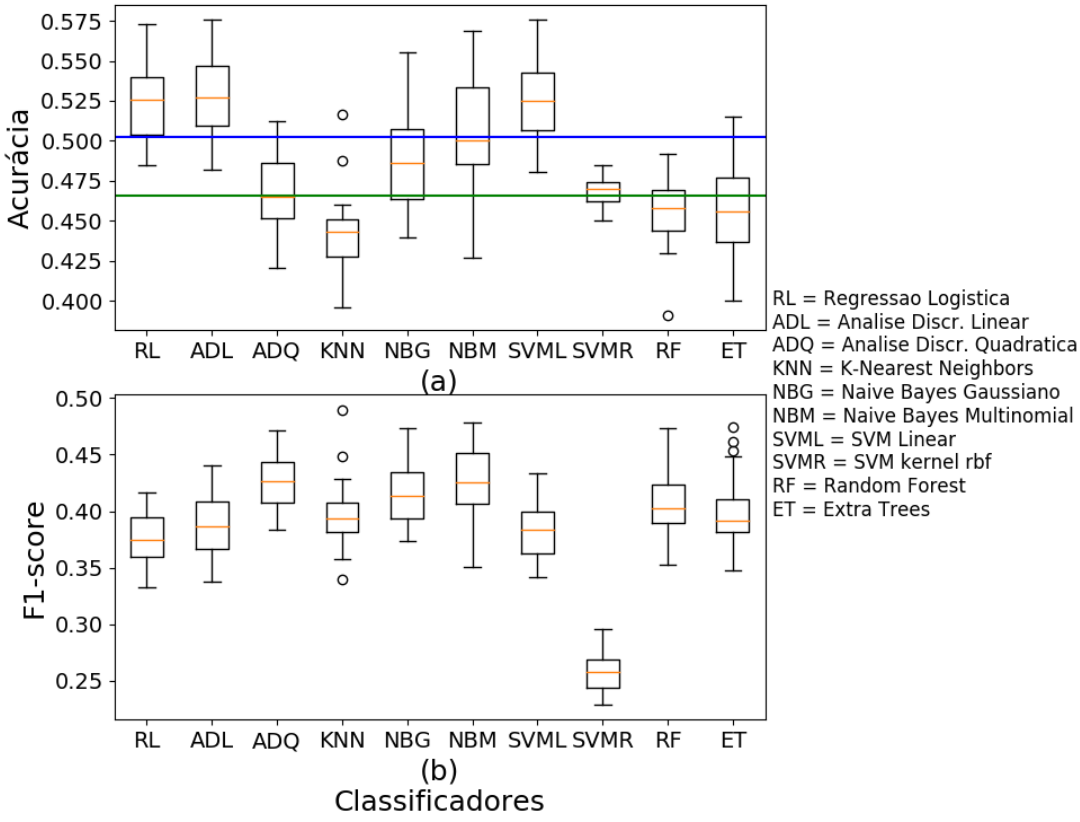
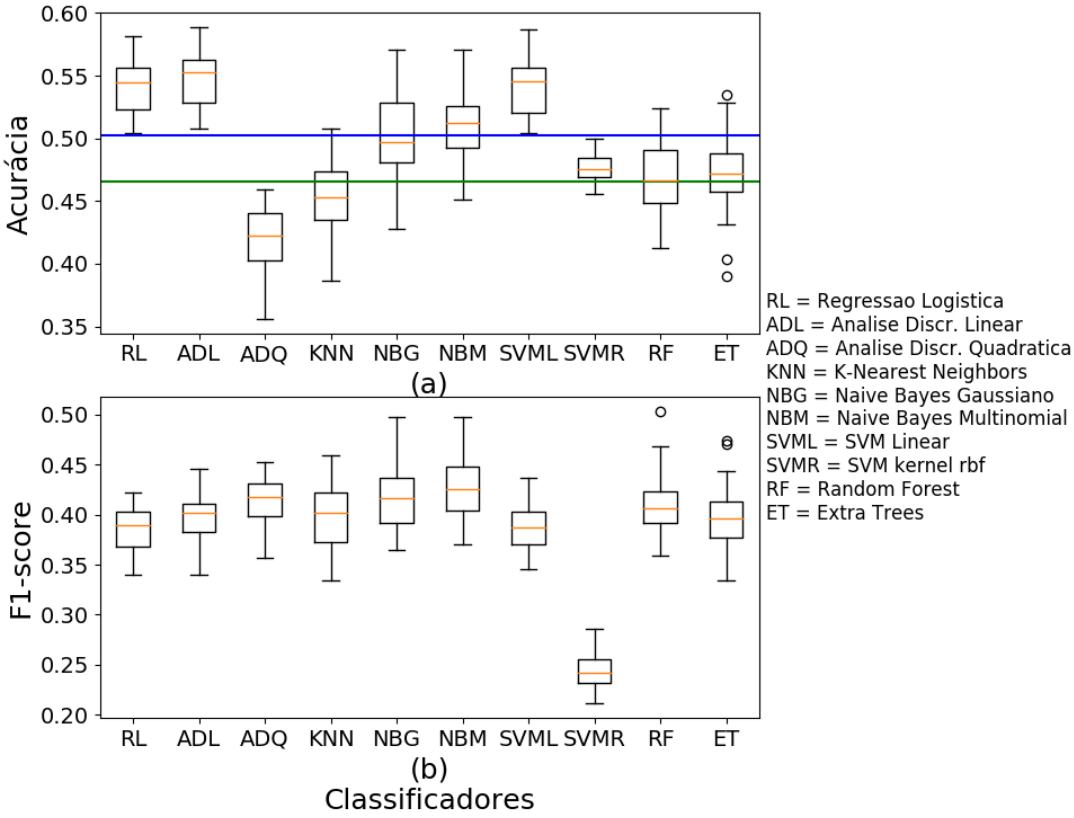


Figura 14 - (a) Acurácia e (b) *F1-score* dos classificadores ao se ignorar as primeiras 19 partidas de cada temporada. A linha azul representa a *baseline* ELO e a linha verde representa a *baseline* Mandante.



A Tabela 9 e a Tabela 10 apresentam os valores numéricos das métricas demonstradas na Figura 12, Figura 13 e na Figura 14.

Tabela 9 - Acurácia dos classificadores em função da quantidade de partidas utilizadas. O melhor desempenho médio é destacado em negrito.

Classificador	Partidas (3 ign.)	Acurácia (Média) (%)	Desvio Padrão (%)	Partidas (10 ign.)	Acurácia (Média) (%)	Desvio Padrão (%)	Partidas (19 ign.)	Acurácia (Média) (%)	Desvio Padrão (%)
RL	4550	52.74	1.48	3640	52.52	2.25	2470	**54.08**	**2.04**
ADL	4550	52.39	1.37	3640	52.79	2.50	2470	**54.82**	**2.01**
ADQ	4550	43.60	4.34	3640	**46.85**	**2.34**	2470	41.92	2.67
KNN	4550	44.33	1.99	3640	44.03	2.19	2470	**45.07**	**2.88**
NBG	4550	49.22	1.72	3640	48.74	3.05	2470	**50.17**	**3.16**
NBM	4550	50.66	1.80	3640	50.54	3.36	2470	**51.14**	**2.79**
SVML	4550	52.38	1.59	3640	52.59	2.39	2470	**54.07**	**2.19**
SVMR	4550	45.87	0.96	3640	46.85	0.82	2470	**47.58**	**0.94**
RF	4550	46.11	1.87	3640	45.69	1.89	2470	**46.83**	**2.69**
ET	4550	46.29	2.03	3640	45.53	2.72	2470	**47.29**	**3.10**

Observa-se que o desvio padrão da maioria dos classificadores tende a aumentar conforme a quantidade de partidas ignoradas aumenta, exceto para os classificador ADQ e

SVMR. Os classificadores que apresentaram maior acurácia foram a Análise Discriminante Linear (ADL), a Regressão Logística (RL) e o *Support Vector Machine* com *kernel* linear (SVML). É possível observar que a acurácia de todos os classificadores, exceto a Análise Discriminante Quadrática, tendem a aumentar com um aumento na quantidade de partidas ignoradas. Isto pode ser explicado através do fato de que ao se ignorar uma maior quantidade de partidas, diversas *features* apresentam valores que definem de maneira mais próxima da realidade a qualidade dos times, como por exemplo as médias de gols, o ELO e a forma.

Tabela 10 - *F1-score* dos classificadores em função da quantidade de partidas utilizadas. O melhor desempenho médio por classificador é destacado em negrito.

Classificador	Partidas (3 ign.)	*F1-score* (Média) (%)	Desvio Padrão (%)	Partidas (10 ign.)	*F1-score* (Média) (%)	Desvio Padrão (%)	Partidas (19 ign.)	*F1-score* (Média) (%)	Desvio Padrão (%)
RL	4550	38.10	1.31	3640	37.64	2.10	2470	**38.56**	**2.03**
ADL	4550	38.80	1.32	3640	38.80	2.53	2470	**39.68**	**2.14**
ADQ	4550	38.15	4.77	3640	**42.63**	**2.33**	2470	41.35	2.47
KNN	4550	**40.24**	**2.09**	3640	39.47	2.48	2470	39.79	3.13
NBG	4550	41.49	1.59	3640	41.46	2.69	2470	**41.71**	**3.05**
NBM	4550	**42.69**	**2.06**	3640	42.51	3.00	2470	42.60	2.86
SVML	4550	**38.92**	**1.48**	3640	38.26	2.29	2470	38.75	2.14
SVMR	4550	25.45	1.31	3640	**25.90**	**1.51**	2470	24.22	1.67
RF	4550	40.38	2.06	3640	40.61	2.45	2470	**41.03**	**2.92**
ET	4550	**40.85**	**2.17**	3640	39.70	2.82	2470	39.83	2.97

É possível observar que não há um aumento expressivo do *F1-score* para nenhum dos classificadores ao se ignorar uma quantidade maior de partidas no início de cada temporada. O classificador que apresentou o maior *F1-score* foi o classificador Naive Bayes Multinomial (NBM) ao se ignorar três partidas, porém o classificador Análise Discriminante Quadrática (ADQ) também apresentou aproximadamente o mesmo valor de *F1-score*.

De forma a validar estatisticamente a diferença entre a distribuição dos resultados obtidos ignorando-se 3 partidas e 19 partidas, empregou-se o teste Mann-Whitney, utilizando-se como amostras 40 instâncias de acurácia média e do *F1-score* médio de cada configuração, obtidos no processo de validação cruzada repetida. A Tabela 11 apresenta os resultados do teste considerando um intervalo de confiança de 5% ($\alpha = 0.05$).

Tabela 11 - Resultado do teste de Mann-Whitney com relação à significância da diferença entre as distribuições das métricas de desempenho. Diferenças significativas (valor-p < 0.05) são destacadas em negrito.

Classificador	valor-*p* (acurácia)	valor-*p* (*F1-score*)	Distribuição da acurácia significativamente diferente	Distribuição do *F1-score* significativamente diferente
RL	**0.0023**	0.1371	Sim	Não
ADL	**4.67×10^{-7}**	**0.0167**	Sim	Sim
ADQ	**0.0005**	**0.0004**	Sim	Sim
KNN	0.1090	0.6476	Não	Não
NBG	0.1543	0.9501	Não	Não
NBM	0.5832	0.7254	Não	Não
SVML	**0.0009**	0.6545	Sim	Não
SVMR	**6.22×10^{-10}**	**0.001**	Sim	Sim
RF	0.1488	0.6545	Não	Não
ET	**0.049**	**0.034**	Sim	Sim

Observa-se que o *F1-score* é significativamente alterado pela quantidade de partidas ignoradas apenas para os classificadores Análise Discriminante Quadrática (ADQ), Análise Discriminante Linear (ADL), *Support Vector Machine* com *kernel* RBF (SVMR) e para o classificador *Extra Trees* (ET), enquanto que a acurácia apresenta distribuição significativamente diferente em 6 dentre 10 classificadores. Uma hipótese para tal comportamento pode ser baseada no fato de que o ELO possui grande impacto (como será demonstrado posteriormente) na tomada de decisão dos classificadores. Assim, ignorando-se mais partidas, obtém-se um ELO mais atual e que corresponde mais fielmente à realidade dos times competidores daquela temporada, o que por sua vez "facilita" a tomada de decisão de qual time será o vencedor, aumentando a acurácia. Porém, este comportamento não necessariamente aumentará o *F1-score*, visto que não necessariamente a taxa de acerto por classe será aumentada. Ou seja, um classificador pode aumentar sua acurácia sem aumentar seu *F1-score* se o maior número de acertos se concentrar em uma classe específica, como a classe majoritária, tendo em vista o desbalanceamento de classes existente. Observa-se, em grande parte dos classificadores onde existe uma diferença significativa entre as distribuições de acurácia, que a mesma aumentou com um aumento na quantidade de partidas ignoradas.

6.3.2 OTIMIZAÇÃO DOS HIPERPARÂMETROS DOS CLASSIFICADORES

Nesta seção será detalhada a busca pelos melhores hiperparâmetros de cada classificador. Deve-se ter em mente que os classificadores Análise Discriminante Linear (ADL), Análise Discriminante Quadrática (ADQ) e o classificador Naive Bayes Gaussiano (NBG) não possuem hiperparâmetros que possam ser otimizados e, portanto, não serão incluídos nesta análise.

Tanto a acurácia quanto o *F1-score* foram otimizados ao se utilizar a penalização L1 com parâmetro C = 625 com o classificador Regressão Logística.

O classificador SVM linear com parâmetro C = 0.1 apresentou maior acurácia. Já o maior *F1-score* foi registrado ao se utilizar C = 10.

O classificador SVM com *kernel* RBF com parâmetro C = 1 e $\gamma = 0.0001$ apresentou maior acurácia. Já o maior *F1-score* foi encontrado utilizando-se C = 10 e $\gamma = 0.001$.

O classificador *K-Nearest Neighbors* apresentou a maior acurácia quando definiu-se a quantidade de vizinhos como 450. Já para o *F1-score*, o classificador foi otimizado para uma quantidade de vizinhos igual a 10.

O classificador Naive Bayes Multinomial apresentou a maior acurcácia com o parâmetro de suavização $\alpha = 1130$. O maior *F1-score* registrado foi alcançado utilizando-se $\alpha = 15$.

Os hiperparâmetros que otimizaram a acurácia ao se utilizar o classificador *Random Forest* foram a utilização da entropia como critério para avaliação da qualidade da divisão da árvore, a utilização de 1000 árvores e a limitação do número máximo de *features* consideradas ao se procurar a melhor divisão como cinco. Já a otimização do *F1-score* ocorreu ao se utilizar 1000 árvores, 17 *features* consideradas em cada divisão e critério de Índice Gini.

Os hiperparâmetros que otimizaram a acurácia ao se utilizar o classificador *Extra Trees* foram a utilização do critério de Índice Gini como critério para avaliação da qualidade da divisão de nós da árvore, a utilização de 1000 árvores e a limitação do número máximo de *features* consideradas ao se procurar a melhor divisão como cinco. Já a otimização do *F1-score* ocorreu ao se utilizar 1000 árvores, 17 *features* consideradas em cada divisão e critério da entropia.

6.3.3 DESEMPENHO DOS CLASSIFICADORES APÓS OTIMIZAÇÃO

A Tabela 12 e a Tabela 13 apresentam acurácia e o *F1-score* após otimização. Os dados apresentados na Tabela 12 foram obtidos ao se utilizar os hiperparâmetros otimizados para a acurácia, enquanto os dados apresentados na Tabela 13 apresentam os resultados obtidos ao se otimizar os hiperparâmetros com relação ao *F1-score*.

Tabela 12 - Acurácia dos classificadores em função da quantidade de partidas utilizadas após otimização dos classificadores. O melhor desempenho médio por classificador é destacado em negrito.

Classificador	Partidas (3 ign.)	Acurácia (Média) (%)	Desvio Padrão (%)	Partidas (10 ign.)	Acurácia (Média) (%)	Desvio Padrão (%)	Partidas (19 ign.)	Acurácia (Média) (%)	Desvio Padrão (%)
RL	4550	52.40	1.44	3640	52.87	2.49	2470	**54.73**	**2.06**
ADL	4550	52.39	1.37	3640	52.79	2.50	2470	**54.82**	**2.01**
ADQ	4550	43.60	4.34	3640	**46.85**	**2.34**	2470	41.92	2.67
KNN	4550	53.20	1.41	3640	53.13	2.05	2470	**54.34**	**1.93**
NBG	4550	49.22	1.72	3640	48.74	3.05	2470	**50.17**	**3.16**
NBM	4550	**52.95**	**1.41**	3640	52.56	2.18	2470	52.87	1.28
SVML	4550	52.85	1.31	3640	53.06	2.21	2470	**54.35**	**2.12**
SVMR	4550	53.09	1.39	3640	53.00	2.07	2470	**54.36**	**2.17**
RF	4550	51.42	1.58	3640	50.65	2.24	2470	**52.14**	**2.24**
ET	4550	51.69	1.53	3640	50.98	1.98	2470	**52.07**	**2.25**

Novamente, observa-se que a maioria dos classificadores, exceto a Análise Discriminante Quadrática (ADQ) e o Naive Bayes Multinomial (NBM) apresentaram maior acurácia média quando as primeiras 19 partidas do campeonato foram ignoradas.

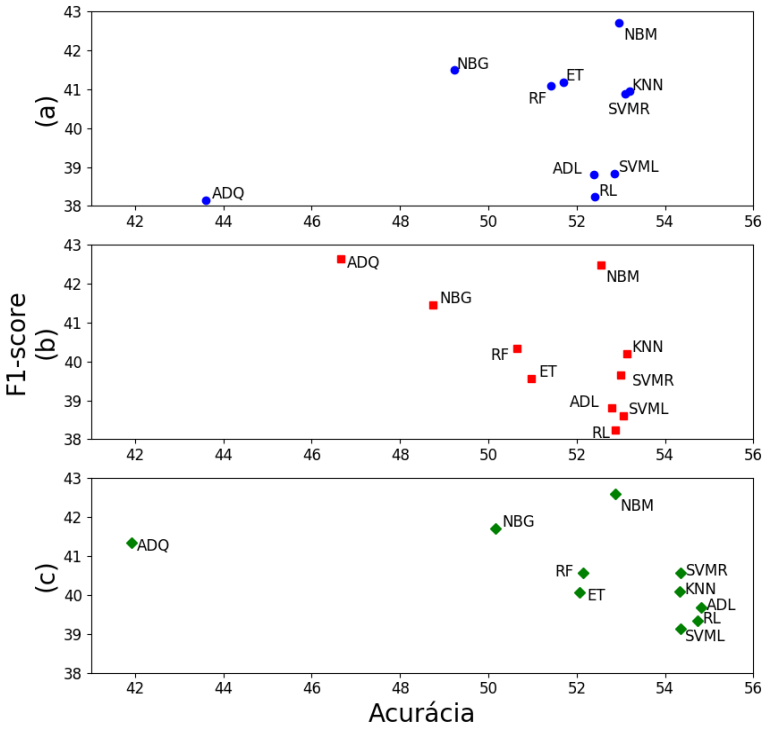
Tabela 13 - *F1-score* dos classificadores em função da quantidade de partidas utilizadas após otimização dos classificadores. O melhor desempenho médio por classificador é destacado em negrito.

Classificador	Partidas (3 ign.)	*F1-score* (Média) (%)	Desvio Padrão (%)	Partidas (10 ign.)	*F1-score* (Média) (%)	Desvio Padrão (%)	Partidas (19 ign.)	*F1-score* (Média) (%)	Desvio Padrão (%)
RL	4550	38.23	1.30	3640	38.23	2.35	2470	**39.34**	**2.13**
ADL	4550	38.80	1.32	3640	38.80	2.53	2470	**39.68**	**2.14**
ADQ	4550	38.15	4.77	3640	**42.63**	**2.33**	2470	41.35	2.47
KNN	4550	**40.94**	**2.07**	3640	40.20	2.39	2470	40.09	3.02
NBG	4550	41.49	1.59	3640	41.46	2.69	2470	**41.71**	**3.05**
NBM	4550	**42.70**	**2.06**	3640	42.48	3.14	2470	42.59	2.85
SVML	4550	38.83	1.67	3640	38.60	2.54	2470	**39.12**	**2.39**
SVMR	4550	**40.88**	**2.00**	3640	39.66	2.80	2470	40.57	3.13
RF	4550	**41.09**	**1.86**	3640	40.33	2.23	2470	40.56	2.82
ET	4550	**41.19**	**1.38**	3640	39.56	2.30	2470	40.07	2.68

Novamente observa-se que classificadores não apresentaram variações expressivas no *F1-score* através da otimização dos hiperparâmetros, apresentando apenas aumento no desvio padrão conforme diminui-se o total de partidas analisadas.

É possível analisar o gráfico de dispersão do *F1-score versus* a acurácia e os efeitos da quantidade de partidas ignoradas sobre as métricas analisadas na Figura 15.

Figura 15 - Gráfico de dispersão do *F1-score versus* acurácia para (a) 3 partidas ignoradas, (b) 10 partidas ignoradas e (c) 19 partidas ignoradas.



Observa-se que o classificador Naibe Bayes Multinomial (NBM) apresenta uma estabilização com relação a acurácia e ao *F1-score*. Além disso, observa-se que a Análise Discriminante Quadrática (ADQ) apresenta a maior variação dentre os classificadores analisados, tanto para a acurácia quanto para o *F1-score*. Nota-se que os classificadores SVMR, KNN, ADL, RL, SVML, ET, RF e NBG apresentam um ganho de acurácia com o aumento da quantidade de partidas ignoradas de 10 partidas para 19 partidas, enquanto que esse aumento na acurácia não foi observado ao se aumentar a quantidade de partidas ignoradas de 3 para 10 partidas.

De forma a validar estatisticamente a diferença entre a distribuição dos resultados obtidos antes e depois da otimização dos hiperparâmetros, empregou-se o teste Mann-Whitney, utilizando-se como amostras 40 instâncias de acurácia média e do *F1-score* médio de cada configuração. Deve-se notar que foram comparadas apenas as métricas na configuração na qual as primeiras 19 partidas são ignoradas, visto que é nesta configuração que a acurácia média é maior. Deve-se ter em mente que os classificadores ADL, ADQ e NBG não possuem hiperparâmetros e, portanto, não sofreram qualquer otimização.

Tabela 14 - Resultado do teste de Mann-Whitney com relação à significância da diferença entre as distribuições das métricas de desempenho. Diferenças significativas (valor-o < 0.05) são destacadas em negrito.

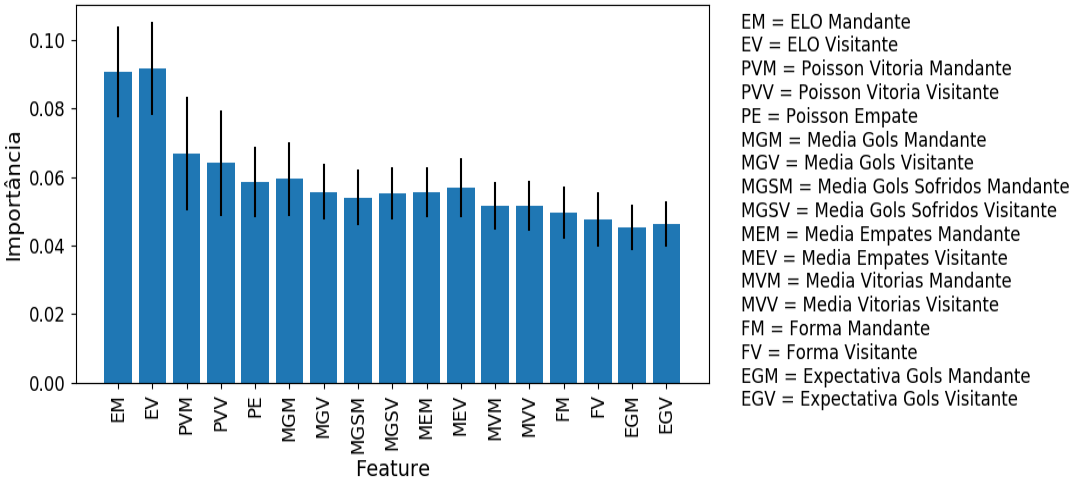
Classificador	valor-*p* (acurácia)	valor-*p* (*F1-score*)	Distribuição da acurácia significativamente diferente	Distribuição do *F1-score* significativamente diferente
RL	0.1514	0.1178	Não	Não
ADL	-	-	-	-
ADQ	-	-	-	-
KNN	**1.65×10^{-14}**	0.7399	Sim	Não
NBG	-	-	-	-
NBM	**0.0007**	0.9577	Sim	Não
SVML	0.6099	0.4329	Não	Não
SVMR	**1.41×10^{-14}**	**1.43×10^{-14}**	Sim	Sim
RF	**2.17×10^{-11}**	0.7326	Sim	Não
ET	**1.08×10^{-9}**	**1.93×10^{-12}**	Sim	Sim

Novamente observa-se que apenas dois classificadores apresentaram variações significativas no *F1-score* através da otimização dos hiperparâmetros. Já a acurácia apresentou distribuição significativamente diferente na maioria dos classificadores, exceto na Regressão Logística (RL) e no *Support Vector Machine* com *kernel* linear (SVML).

6.3.3.1 DESEMPENHO DOS CLASSIFICADORES APÓS SELEÇÃO DE FEATURES

Na Figura 16 é possível observar a importância de cada *feature*, estimada com um classificador *Random Forest*, ao se ignorar as 3 primeiras partidas de cada temporada, sendo a importância definida pelo Índice Gini.

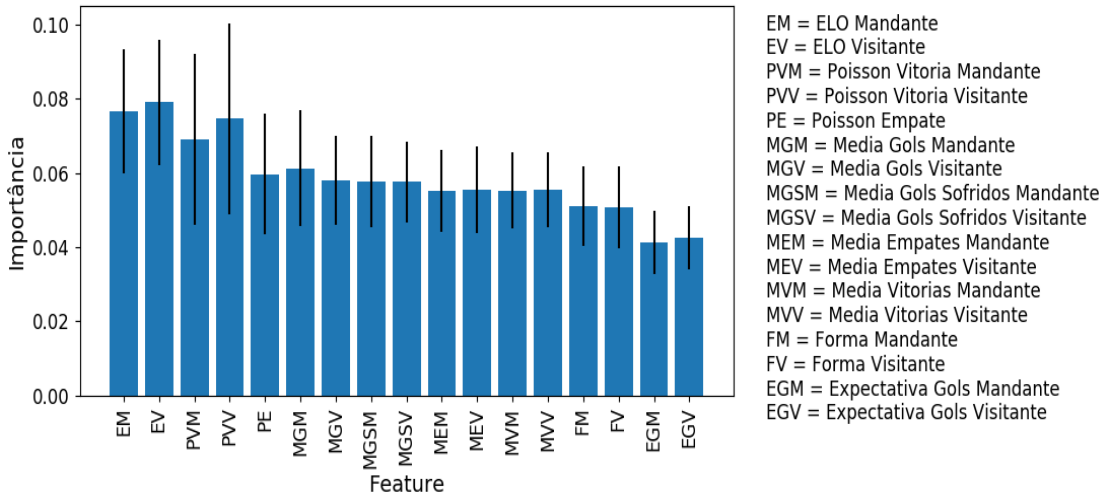
Figura 16 - Importância das *features* ao se ignorar as primeiras 3 partidas de cada temporada.



Na Figura 17 é possível observar a importância de cada *feature* ao se ignorar as 19 primeiras partidas de cada temporada.

Observa-se que a mudança mais expressiva na importância das *features* ao se ignorar um maior número de partidas é o aumento da importância das probabilidades de Poisson de vitória do mandante e do visitante, que se tornam tão importantes quanto os ELOs. Porém, em termos gerais, a importância das *features* não sofre mudanças expressivas com relação a quantidade de partidas analisadas.

Figura 17 - Importância das features ao se ignorar as primeiras 19 partidas de cada temporada.



Observa-se que a mudança mais expressiva na importância das *features* ao se ignorar um maior número de partidas é o aumento da importância das probabilidades de Poisson de vitória do mandante e do visitante, que se tornam tão importantes quanto os ELOs. Porém, em termos gerais, a importância das *features* não sofre mudanças expressivas com relação a quantidade de partidas analisadas.

A Figura 18 apresenta as curvas de acurácia em função da quantidade de *features* utilizadas, sendo que as *features* estão ordenadas da mais importante para a menos importante, de forma decrescente, seguindo a ordem de importância apresentada na Figura 17.

É possível observar que a maior variação registrada é relacionada ao classificador Análise Discriminante Quadrática (ADQ), que apresenta uma queda na acurácia média após a adição da quinta *feature* (Poisson Empate). Observa-se também que os dois classificadores baseados em árvores de decisão (*Random Forest* (RF) e *Extra Trees* (ET)) apresentam um aumento na acurácia média conforme mais *features* são adicionadas, enquanto que os classificadores lineares (Análise Discriminante Linear (ADL) e Regressão Logística (RL)) apresentam uma acurácia média aproximadamente constante com relação à adição de *features* de menor importância.

Figura 18 - Acurácia em função da quantidade de *features*.

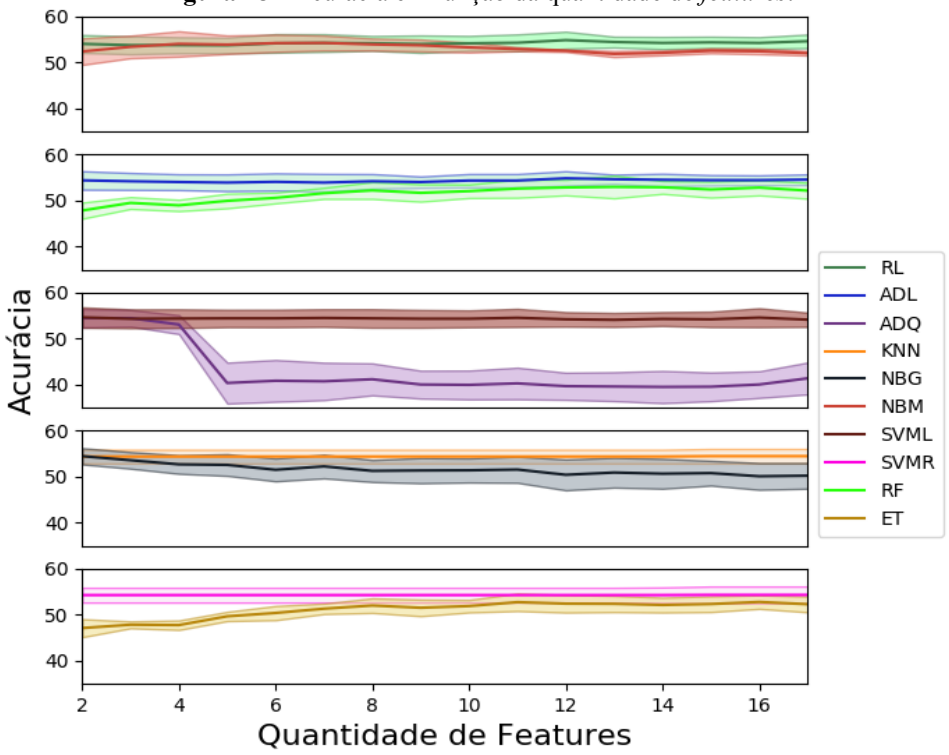
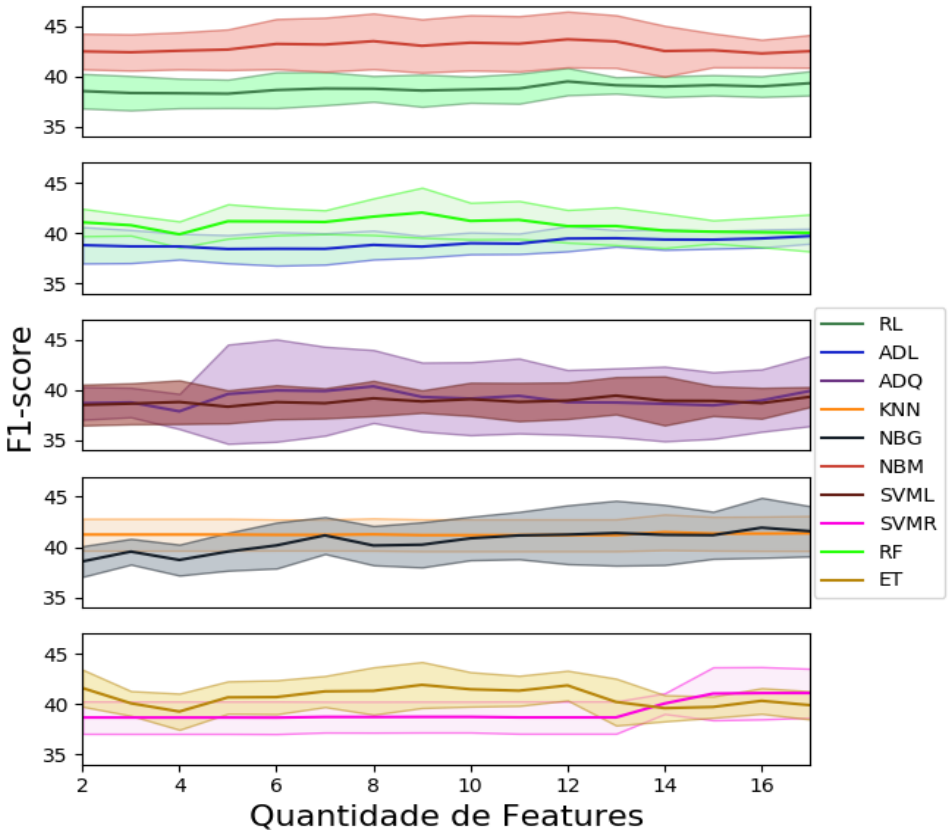


Figura 19 - *F1-score* em função da quantidade de *features*.

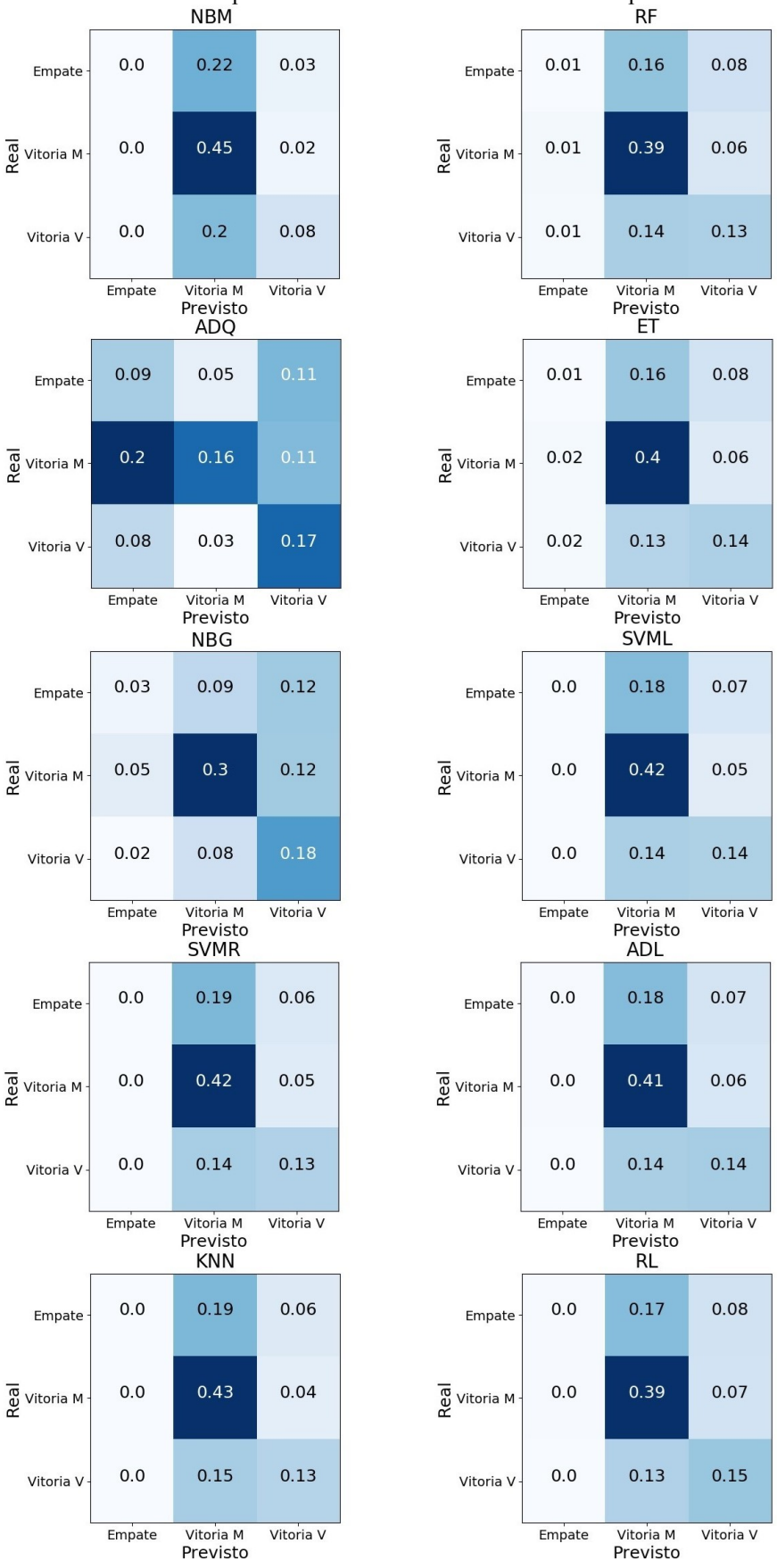


Com relação à variação do *F1*-score em função da quantidade de features usadas no treinamento, é possível observar que após a inclusão da quinta *feature* (Empate Poisson), a variância do classificador Análise Discriminante Quadrática (ADQ) aumenta consideravelmente, sendo que esta característica pode explicar a redução da acurácia observada na curva da Figura 18. Observa-se também que ambos os classificadores baseados em árvores (*Random Forest* (RF) e *Extra Trees* (ET)) apresentam o maior *F1-score* ao se utilizar 9 *features*.

6.3.3.2 DESEMPENHO DOS CLASSIFICADORES EM UM CONJUNTO DE DADOS NOVOS

Na Figura 20 é possível observar a matriz de confusão normalizada dos classificadores ao se utilizar as temporadas compreendidas entre 2002/2003 – 2014/2015 (incluindo ambas) como dados de treinamento e as temporadas 2015/2016 -2016/2017 como dados de teste, ignorando as primeiras 19 partidas de cada temporada.

Figura 20 - Matriz de confusão normalizada em função do total de instâncias analisadas no conjunto de dados de teste. As cores mais escuras representam maior incidência de instâncias previstas corretamente.



A Tabela 15 sumariza e compara a acurácia obtida por todos os classificadores no conjunto de dados de teste, enquanto que as tabelas compreendidas entre a Tabela 16 e a Tabela 25 detalham as métricas (*F1-score*, precisão e *recall*) por classe, para cada um dos classificadores treinados.

Tabela 15 - Acurácia dos classificadores no conjunto de dados de teste.

Classificador	Acurácia (%)
RL	54%
ADL	55%
ADQ	42%
KNN	56%
NBG	51%
NBM	53%
SVML	56%
SVMR	55%
RF	53%
ET	55%

Tabela 16 - Métricas por classe do classificador Regressão Logística (RL).

Classe	*F1-score*	Precisão	*Recall*	Instâncias
Empate	0.02	0.50	0.01	95
Vitória M	0.68	0.57	0.84	179
Vitória V	0.52	0.50	0.55	106

Tabela 17 - Métricas por classe do classificador Análise Discriminante Linear (ADL).

Classe	*F1-score*	Precisão	*Recall*	Instâncias
Empate	0.02	0.50	0.01	95
Vitória M	0.68	0.56	0.86	179
Vitória V	0.51	0.51	0.51	106

Tabela 18 - Métricas por classe do classificador Análise Discriminante Quadrática (ADQ).

Classe	*F1-score*	Precisão	*Recall*	Instâncias
Empate	0.30	0.25	0.37	95
Vitória M	0.46	0.67	0.35	179
Vitória V	0.50	0.43	0.59	106

Tabela 19 - Métricas por classe do classificador *K-Nearest Neighbors* (KNN).

Classe	*F1-score*	Precisão	*Recall*	Instâncias
Empate	0	0	0	95
Vitória M	0.69	0.55	0.91	179
Vitória V	0.51	0.57	0.46	106

Tabela 20 - Métricas por classe do classificador Naive Bayes Gaussiano (NBG).

Classe	*F1-score*	Precisão	*Recall*	Instâncias
Empate	0.18	0.32	0.13	95
Vitória M	0.64	0.64	0.64	179
Vitória V	0.52	0.43	0.66	106

Tabela 21 - Métricas por classe do classificador Naive Bayes Multinomial (NBM).

Classe	*F1-score*	Precisão	*Recall*	Instâncias
Empate	0	0	0	95
Vitória M	0.67	0.52	0.96	179
Vitória V	0.40	0.62	0.29	106

Tabela 22 - Métricas por classe do classificador *Support Vector Machine* com *kernel* linear (SVML).

Classe	*F1-score*	Precisão	*Recall*	Instâncias
Empate	0	0	0	95
Vitória M	0.69	0.56	0.90	179
Vitória V	0.52	0.55	0.49	106

Tabela 23 - Métricas por classe do classificador *Support Vector Machine* com *kernel* RBF (SVMR).

Classe	*F1-score*	Precisão	*Recall*	Instâncias
Empate	0	0	0	95
Vitória M	0.69	0.56	0.89	179
Vitória V	0.52	0.55	0.48	106

Tabela 24 - Métricas por classe do classificador *Random Forest* (RF).

Classe	*F1-score*	Precisão	*Recall*	Instâncias
Empate	0.04	0.18	0.02	95
Vitória M	0.68	0.57	0.84	179
Vitória V	0.47	0.47	0.47	106

Tabela 25 - Métricas por classe do classificador *Extra Trees* (ET).

Classe	*F1-score*	Precisão	*Recall*	Instâncias
Empate	0.07	0.25	0.04	95
Vitória M	0.69	0.58	0.84	179
Vitória V	0.49	0.50	0.49	106

Sabendo que o *Recall* informa a taxa de verdadeiros positivos, isto é, quanto dos resultados relevantes foram corretamente preditos pelos classificadores, é possível constatar que o classificador Análise Discriminante Quadrática (ADQ), por apresentar o maior *Recall* para a classe Empate, foi o classificador que melhor recuperou instâncias pertencentes a esta classe. Por outro lado, por ter uma precisão baixa, é possível que muitas instâncias da classe Vitória M ou Vitória V tenham sido incorretamente classificadas por este classificado como sendo da classe Empate. O classificador Naive Bayes Gaussiano (NBG), embora não apresente *Recall* comparável ao da Análise Discriminante Quadrática (ADQ), também apresenta um bom número de partidas de Empate classificadas como pertencentes esta classe Empate, o que faz com que o *Recall* deste classificador seja superior aos dos demais

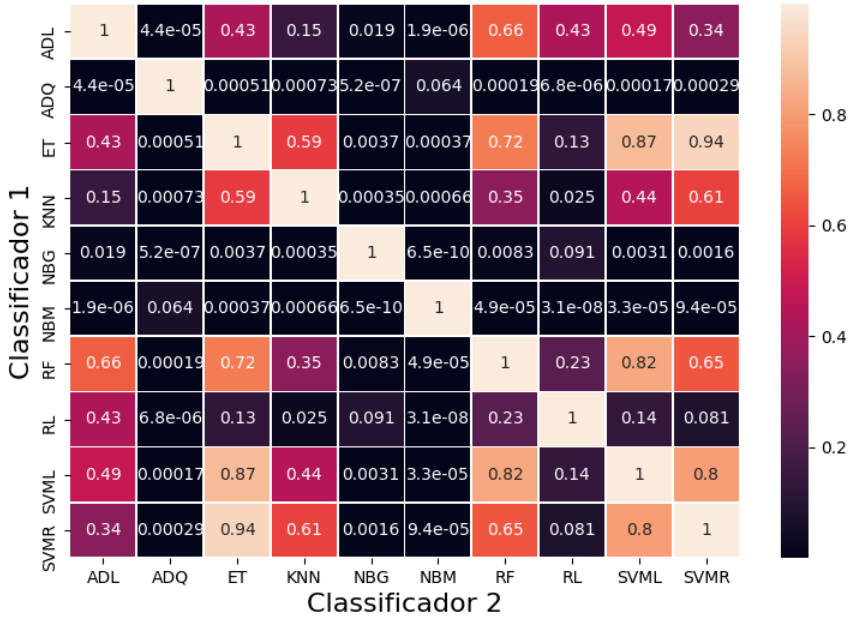
classificadores, os quais não conseguem recuperar de forma satisfatória instâncias referentes a empates.

É possível observar que a grande maioria dos classificadores não prevê empates ou então prevê uma quantidade baixa desta instância. O classificador Análise Discriminante Quadrática é o classificador que mais prevê empates, porém o mesmo é o que apresenta a menor acurácia, visto também que este classificador é o que apresenta a menor quantidade de previsões com relação à classe Vitória Mandante (classe majoritária neste contexto). A partir da análise deste conjunto de dados novos, observou-se que os classificadores que apresentaram maior acurácia foram o *Support Vector Machine* com *kernel* linear (SVML), apresentando 56% de acurácia, *K-Nearest Neighbors* (KNN), apresentando também 56% de acurácia. Nota-se que ambos os classificadores que apresentaram a maior acurácia não possuem previsões para a classe Empate. O classificador que prevê instâncias como sendo pertencentes à classe Empate o qual apresentou maior acurácia foi o classificador *Extra Trees* (RF), apresentando 55% de acurácia.

6.3.4 DESEMPENHO DE UM CLASSIFICADOR *ENSEMBLE*

De forma a selecionar classificadores para criar um *ensemble*, observou-se se as distribuições das instâncias previstas pelos classificadores são significativamente diferentes através do teste Mann-Whitney, de forma a se possibilitar a criação de um *ensemble* com classificadores que não apresentam a mesma metodologia de previsão, ou seja, classificadores que não realizem exatamente a mesma previsão para as mesmas instâncias, visto que este comportamento pode criar um viés para o classificador *ensemble*.

Figura 21 - Valor-p do teste estatístico Mann-Whitney aplicado à distribuição das instâncias previstas por cada classificador ao utilizar o conjunto de dados de teste.



A análise da existência de uma diferença significativa entre as distribuições das instâncias previstas pelos classificadores é realizada de forma a evitar a inclusão de uma quantidade elevada de classificadores que não possuem uma real diferença em suas previsões no *ensemble*, visto que desta forma não há um ganho real de generalização e robustez, criando um viés na previsão do classificador *ensemble*. Na análise da Figura 21, utilizou-se novamente o valor-p de 0.05 como limite, sendo que valores menores do que 0.05 possibilitam a rejeição da hipótese nula de que não existe diferença significativa entre as distribuições de instâncias previstas.

Assim, optou-se pela utilização dos classificadores Regressão Logística (RL), Naive Bayes Multinomial (NBM), Naive Bayes Gaussiano (NBG), *K-Nearest Neighbors* (KNN), *Extra Trees* (ET) e Análise Discriminante Quadrática (ADQ) no *ensemble* (ENS), utilizando como *ensemble* um classificador que decide através da votação da maioria a qual classe uma instância pertence. Todos os classificadores os quais possuem hiperparâmetros foram utilizados de acordo com a configuração otimizada dos mesmos, tanto para acurácia quanto para o *F1-score*.

Primeiramente, repetiu-se a análise realizada na Seção 6.3.3 (apenas para o classificador *ensemble*). Os resultados podem ser observados na Tabela 26 e na Tabela 27, podendo-se também comparar o desempenho do classificador *ensemble* com as métricas já obtidas dos demais classificadores, as quais foram repetidas de forma a facilitar esta comparação.

Tabela 26 - Acurácia dos classificadores em função da quantidade de partidas utilizadas após otimização dos classificadores. O melhor desempenho médio por classificador é destacado em negrito.

Classificador	Partidas (3 ign.)	Acurácia (Média) (%)	Desvio Padrão (%)	Partidas (10 ign.)	Acurácia (Média) (%)	Desvio Padrão (%)	Partidas (19 ign.)	Acurácia (Média) (%)	Desvio Padrão (%)
RL	4550	52.40	1.44	3640	52.87	2.49	2470	**54.73**	**2.06**
ADL	4550	52.39	1.37	3640	52.79	2.50	2470	**54.82**	**2.01**
ADQ	4550	43.60	4.34	3640	**46.85**	**2.34**	2470	41.92	2.67
KNN	4550	53.20	1.41	3640	53.13	2.05	2470	**54.34**	**1.93**
NBG	4550	49.22	1.72	3640	48.74	3.05	2470	**50.17**	**3.16**
NBM	4550	**52.95**	**1.41**	3640	52.56	2.18	2470	52.87	1.28
SVML	4550	52.85	1.31	3640	53.06	2.21	2470	**54.35**	**2.12**
SVMR	4550	53.09	1.39	3640	53.00	2.07	2470	**54.36**	**2.17**
RF	4550	51.42	1.58	3640	50.65	2.24	2470	**52.14**	**2.24**
ET	4550	51.69	1.53	3640	50.98	1.98	2470	**52.07**	**2.25**
ENS	4550	52.51	1.32	3640	52.94	2.20	2470	**54.51**	**2.04**

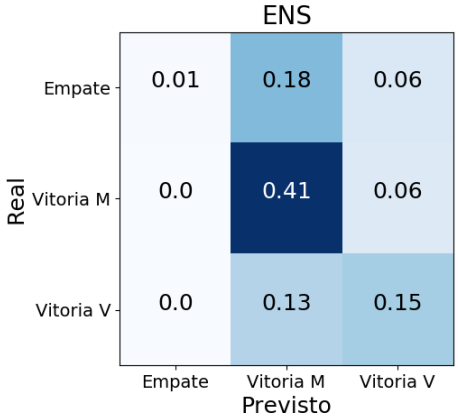
Tabela 27 - *F1-score* dos classificadores em função da quantidade de partidas utilizadas após otimização dos classificadores. O melhor desempenho médio por classificador é destacado em negrito.

Classificador	Partidas (3 ign.)	*F1-score* (Média) (%)	Desvio Padrão (%)	Partidas (10 ign.)	*F1-score* (Média) (%)	Desvio Padrão (%)	Partidas (19 ign.)	*F1-score* (Média) (%)	Desvio Padrão (%)
RL	4550	38.23	1.30	3640	38.23	2.35	2470	**39.34**	**2.13**
ADL	4550	38.80	1.32	3640	38.80	2.53	2470	**39.68**	**2.14**
ADQ	4550	38.15	4.77	3640	**42.63**	**2.33**	2470	41.35	2.47
KNN	4550	**40.94**	**2.07**	3640	40.20	2.39	2470	40.09	3.02
NBG	4550	41.49	1.59	3640	41.46	2.69	2470	**41.71**	**3.05**
NBM	4550	**42.70**	**2.06**	3640	42.48	3.14	2470	42.59	2.85
SVML	4550	38.83	1.67	3640	38.60	2.54	2470	**39.12**	**2.39**
SVMR	4550	**40.88**	**2.00**	3640	39.66	2.80	2470	40.57	3.13
RF	4550	**41.09**	**1.86**	3640	40.33	2.23	2470	40.56	2.82
ET	4550	**41.19**	**1.38**	3640	39.56	2.30	2470	40.07	2.68
ENS	4550	**41.48**	**1.85**	3640	40.09	3.00	2470	41.41	3.07

Observa-se que o classificador *ensemble* apresenta desempenho comparável ao dos classificadores com melhores métricas, tanto com relação à acurácia quanto ao *F1-score*.

A Figura 22 apresenta a matriz de confusão do classificador *ensemble* ao analisar o conjunto de dados de teste, ou seja, ao analisar um conjunto de dados novos, seguindo a análise realizada na Seção 6.3.3.2.

Figura 22 - Matriz de confusão do classificador *ensemble* ao analisar o conjunto de dados de teste.



As demais métricas obtidas para classificação das instâncias no conjunto de teste são apresentadas na Tabela 28.

Tabela 28 - Métricas por classe do classificador *Ensemble* (ENS).

Classe	*F1-score*	Precisão	Recall	Instâncias
Empate	0.04	0.5	0.02	95
Vitória M	0.69	0.57	0.88	179
Vitória V	0.54	0.56	0.53	106

Observa-se que o classificador *ensemble* apresenta acurácia de 57%, superior a acurácia dos classificadores *K-Nearest Neighbors* (KNN) e *Support Vector Machine* com *kernel* linear (SVML), sendo que ambos apresentaram acurácia de 56% ao analisar o mesmo conjunto de dados Ainda que o classificador *ensemble* não tenha uma boa capacidade de recuperar corretamente uma proporção satisfatória de exemplos da classe Empate, ele tem uma precisão mais equilibrada entre as três classes, errando menos a previsão de exemplos classificados como Empate.

6.3.4.1 COMPARAÇÃO COM A LITERATURA

A Tabela 29 apresenta uma comparação entre os resultados obtidos neste trabalho com os resultados utilizados na literatura referenciada.

Tabela 29 - Resumo dos métodos e resultados encontrados pelos autores estudados e resumo dos resultados encontrados neste trabalho.

Autor	Método	Partidas	Acurácia (%)
(ULMER; FERNANDEZ, 2013)	SVM Linear	760	51
	One-vs-all SGD	760	52
	One-vs-all SGD	342	51
	One-vs-all SGD	266	49
	One-vs-all SGD	190	48
	One-vs-all SGD	114	49
	SVM RBF	760	48
	Random Forest	760	50
(JOSEPH; FENTON; NEIL, 2006)	Rede Bayesiana Expert	114	59
(CHENG *et al.*, 2003)	Rede LVQ + BP	153	52
	ELO clássico	153	47
	Razão entre gols	153	49
(TRINDADE, 2013)	MLE	190	56
	MP	190	54
(SCHNEIDER, 2018)	RL (validação cruzada)	2470	54.73
	ADL (validação cruzada)	2470	54.82
	KNN (validação cruzada)	2470	54.34
	SVML (validação cruzada)	2470	54.35
	SVMR (validação cruzada)	2470	54.36
	ENS (validação cruzada)	2470	54.51
	RL	760	54
	ADL	760	55
	KNN	760	56
	SVML	760	56
	SVMR	760	55
	ENS	760	57

É possível observar que a análise realizada por Joseph, Fenton e Neil (2006) ainda apresenta a maior acurácia registrada na literatura estudada. Porém, deve-se ter em mente de que o trabalho realizado pelos autores analisa apenas um time, permitindo uma análise mais particular das *features* de interesse quando comparada a análise de um campeonato com dezenas de times ao longo dos anos. Observa-se que o desempenho do classificador *ensemble* na etapa de testes apresentou acurácia maior do que a observada nos demais trabalhos da

literatura. Além disso, observa-se que diversos classificadores apresentaram acurácia de 56%, comparável a maior acurácia registrada na literatura para análises de campeonatos.

7 CONCLUSÃO

Através da análise de desempenho de distintos classificadores utilizando *validação cruzada*, foi possível observar que os classificadores que apresentam a maior acurácia média foram a Regressão Logística (54.73% ± 2.06%), a Análise Discriminante Linear (54.82% ± 2.01), *Support Vector Machine*, tanto com o *kernel* linear (54.35% ± 2.12%) quanto com o *kernel* RBF (54.36% ± 2.24), e o classificador *K-Nearest Neighbors* (54.34% ± 1.93%), sendo que todos estes valores de acurácia média foram obtidos ao se ignorar as primeiras 19 partidas de cada temporada.

Entretanto, com base na análise da matriz de confusão gerada a partir da previsão realizada em um conjunto de dados de teste, observou-se que estes classificadores não apresentaram previsões de instâncias como pertencentes à classe Empate, o que está de acordo com resultados encontrados na literatura relacionada. Além disso, ao se observar as distribuições de previsão das instâncias do conjunto de dados de teste, foi possível constatar que as mesmas, em sua grande maioria, não são significativamente diferentes, o que embasa a conclusão de que a acurácia elevada destes classificadores está baseada nas mesmas premissas. Sabendo que estes classificadores possuem um viés para a previsão de vitórias do mandante ou do visitante é possível concluir que o aumento da acurácia a partir do aumento da quantidade de partidas ignoradas está ligado ao fato de que ao se ignorar uma quantidade maior de partidas, os classificadores possuem informações mais refinadas para determinar qual o time mais qualificado e, portanto, defini-lo como o vencedor da partida em questão.

O classificador o qual apresentou maior *F1-score* na análise com *cross-validation* foi a Análise Discriminante Quadrática (42.63% ± 2.33%), devido em parte ao fato de que este classificador é o que mais prevê instâncias como sendo pertencentes a classe Empate, o que o coloca a frente dos demais classificadores ao se analisar o *F1-score*. Porém, o mesmo

apresenta a menor acurácia entre todos os classificadores observados (42.63% ± 2.33%), demonstrando o comportamento detalhado pelo Teorema *No-Free-Lunch*.

Através da criação de um classificador *ensemble* com classificadores que possuem distribuições de previsões de instâncias significativamente diferentes observou-se uma acurácia média comparável aos maiores valores de acurácia observados pelos classificadores individuais. A mesma conclusão pode ser feita com relação ao *F1-score*. Ao se observar a matriz de confusão gerada através da análise de um conjunto de dados de teste constatou-se que a acurácia do classificador *ensemble* foi 1% maior (57%) do que a constatada nos classificadores individuais (56% - SVML e KNN). Assim, a utilização de um *ensemble* de classificadores pode apresentar-se como uma alternativa viável para aumentar a robustez e capacidade de generalização de um modelo de previsões, visto que o classificador *ensemble* apresentou resultados comparáveis aos encontrados na literatura.

Para trabalhos futuros, indica-se o estudo do aumento da incidência de previsões para a classe empate, sabendo que este problema está conectado ao desbalanço de classes proveniente da própria natureza do esporte analisado, visto que empates são os resultados com menor incidência na *Premier League*. Sugere-se o estudo da otimização de um classificador específico para aumento das instâncias previstas como Empate, adicionando-o ao *ensemble* de classificadores de forma a aumentar a capacidade de generalização do modelo. Além disso, sugere-se também o estudo de *features* que melhor descrevam partidas que possivelmente terminem em empate, visto que o ELO, sendo a *feature* de maior importância, descreve de forma mais adequada vitórias e derrotas. O estudo da incidência de empates de acordo com a posição da tabela da temporada atual é um exemplo de *feature* que poderia descrever times com tendência a empatar.

REFERÊNCIAS

ARLOT, S.; CELISSE, A. A survey of cross-validation procedures for model selection *. **Statistics Surveys**, 2010. v. 4, p. 40–79. Disponível em: <https://projecteuclid.org/download/pdfview_1/euclid.ssu/1268143839>. Acesso em: 21 out. 2017.

ASLAN, B. G.; INCEOGLU, M. M. A Comparative Study on Neural Network Based Soccer Result Prediction. **Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)**, 2007. p. 545–550. Disponível em: <http://ieeexplore.ieee.org/document/4389664/>.

BERWICK, R. C. An Idiot's guide to Support vector machines (SVMs). 2003. Disponível em: <http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf>. Acesso em: 8 dez. 2017.

BISGIN, H. *et al.* Diagnosis of long QT syndrome via support vector machines classification. **Journal of Biomedical Science and Engineering**, 2011. v. 4, n. 4, p. 264–271. Disponível em: <http://www.scirp.org/journal/doi.aspx?DOI=10.4236/jbise.2011.44036>. Acesso em: 22 out. 2017.

BOLLINGER, J. G.; DUFFIE, N. A. **Computer control of machines and processes**. [S.l.]: Addison-Wesley, 1988.

BREIMAN, L. Bagging Predictors. **Machine Learning**, 1996. v. 24, n. 2, p. 123–140. Disponível em: <http://link.springer.com/10.1023/A:1018054314350>. Acesso em: 8 dez. 2017.

CHENG, T. *et al.* A new model to forecast the results of matches based on hybrid neural networks in the soccer rating system. **Proceedings - 5th International Conference on Computational Intelligence and Multimedia Applications, ICCIMA 2003**, 2003. p. 308–313.

CORTES, C.; VAPNIK, V. **SUPPORT-VECTOR NETWORKS**. [S.l.]: [s.n.], 1995.

CRONIN, B. Poisson Distribution betting | How to predict soccer results using Poisson Distribution. [S.l.], 2017. Disponível em: <https://www.pinnacle.com/en/betting-articles/Soccer/how-to-calculate-poisson-distribution/MD62MLXUMKMXZ6A8>. Acesso em: 13 out. 2017.

DOBSON, S.; GODDARD, J. The Economics of Football. 2001. Disponível em: <http://www.cambridge.org>. Acesso em: 3 dez. 2017.

FOOTBALLDATABASE.COM. Methodology for Calculating FootballDatabase's World Football Clubs Ranking. [S.l.], 2017. Disponível em: <http://footballdatabase.com/methodology.php>. Acesso em: 14 out. 2017.

FREUND, Y.; SCHAPIRE, R. E. A Short Introduction to Boosting. **Journal of Japanese Society for Artificial Intelligence**, 1999. v. 14, n. 5, p. 771–780. Disponível em: <www.research.att.com/>. Acesso em: 10 dez. 2017.

GAMA, J. A. Functional Trees. **Machine Learning**, 2004. v. 55, p. 219–250. Disponível em: <https://link.springer.com/content/pdf/10.1023%2FB%3AMACH.0000027782.67192.13.pdf>. Acesso em: 10 dez. 2017.

GANGANWAR, V. An overview of classification algorithms for imbalanced datasets. **International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com**, 2012. v. 2, n. 4. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.3344&rep=rep1&type=pdf>. Acesso em: 22 out. 2017.

GÉRON, A. **Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems**. [S.l.]: [s.n.], 2017.

GYULA KRISZTIÁN, F. Predictive analysis of financial time series. 2014. Disponível em: <http://web.cs.elte.hu/blobs/diplomamunkak/bsc_matelem/2014/fora_gyula_krisztian.pdf>. Acesso em: 22 out. 2017.

HOPKINS, W. G. Impact Factors of Journals in Sport and Exercise Science and Medicine for 2010. **Sportscience**, 2010. v. 14, p. 60–62. Disponível em: <http://www.sportsci.org/2010/wghif.htm>. Acesso em: 3 dez. 2017.

JOSEPH, A.; FENTON, N. E.; NEIL, M. Predicting football results using Bayesian nets and other machine learning techniques. **Knowledge-Based Systems**, 2006. v. 19, n. 7, p. 544–553.

KHATTREE, R.; NAIK, D. N. **Applied Multivariate Statistics with SAS software**. Second Edition ed. Cary: [s.n.], 2003.

KOÇO, S. *et al.* On multi-class classification through the minimization of the confusion matrix norm. 2013. v. 29, p. 277–292. Disponível em: <http://proceedings.mlr.press/v29/Koco13.pdf>. Acesso em: 31 out. 2017.

KOLODNER, J. L. **Case-based reasoning**. [S.l.]: Morgan Kaufmann Publishers, 1993.

KOTSIANTIS, S. B. Supervised Machine Learning: A Review of Classification Techniques. **Informatica**, 2007. v. 31, p. 249–268. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.9683&rep=rep1&type=pdf>. Acesso em: 22 out. 2017.

LACY, S. Implementing an Elo rating system for European football | Stuart Lacy. [S.l.], 2017. Disponível em: <http://stuartlacy.co.uk/2017/08/31/implementing-an-elo-rating-system-for-european-football/>. Acesso em: 14 out. 2017.

LASEK, J.; SZLÁVIK, Z.; BHULAI, S. The predictive power of ranking systems in association football. **Int. J. Applied Pattern Recognition**, 2013. v. 1, n. 1, p. 27–46.

LI, L.; WU, Y.; YE, M. Experimental Comparisons of Multi-class Classifiers. **Informatica**, 2015. v. 39, p. 71–85. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.948.8353&rep=rep1&type=pdf>. Acesso em: 31 out. 2017.

MAHER, M. J. Modelling association football scores. [s.d.]. Disponível em: <http://www.90minut.pl/misc/maher.pdf>. Acesso em: 13 out. 2017.

MARKOVITCH, S.; ROSENSTEIN, D. Feature Generation Using General Constructor Functions. **Machine Learning**, 2002. v. 49, n. 1, p. 59–98. Disponível em: <http://link.springer.com/10.1023/A:1014046307775>. Acesso em: 22 out. 2017.

MICHALSKI, R. S.; CARBONELL, J. G.; MITCHELL, T. M. **Machine Learning : an Artificial Intelligence Approach**. [S.l.]: Springer Berlin Heidelberg, 1983.

NG, A. CS229 Lecture notes. 2008. Disponível em: <https://see.stanford.edu/materials/aimlcs229/cs229-notes1.pdf>. Acesso em: 22 out. 2017.

NILSSON, N. J. **Introduction to Machine Learning**. Stanford: Stanford University, 1998.

OMEZ, D.; ROJAS, A. An Empirical Overview of the No Free Lunch Theorem and Its Effect on Real-World Machine Learning Classification. 2016. Disponível em: <http://upcommons.upc.edu/bitstream/handle/2117/81906/An+empirical+overview+of+the+No+Free+Lunch+Theorem+and+its+effect+on+Real-World+Machine+Learning+Classification(1).pdf;jsessionid=081BBC89CAD1C5625FC55A3A1E0C8BEB?sequence=6>. Acesso em: 22 out. 2017.

PAULINO, D. C. *et al.* Sociedade Portuguesa de Estatística Autores. 2011. Disponível em: <http://www.ufpa.br/heliton/arquivos/aplicada/glossario_SPEABE.pdf>. Acesso em: 10 dez. 2017.

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python Gaël Varoquaux. **Journal of Machine Learning Research**, 2011. v. 12, p. 2825–2830. Disponível em: <http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>. Acesso em: 21 out. 2017.

REFAEILZADEH, P.; TANG, L.; LIU, H. Scientific Fundamentals. 2008. Disponível em: <http://leitang.net/papers/ency-cross-validation.pdf>. Acesso em: 21 out. 2017.

SAITO, T.; REHMSMEIER, M. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. **PLOS ONE**, 4 mar. 2015. v. 10, n. 3, p. e0118432. Disponível em: <http://dx.plos.org/10.1371/journal.pone.0118432>. Acesso em: 22 out. 2017.

SHALEV-SCHWARTZ, S.; BEN-DAVID, S. **Understanding Machine Learning: From Theory to Algorithms**. [S.l.]: [s.n.], 2014.

SILVA, L. M. O. Da. **Uma Aplicação de Árvores de Decisão, Redes Neurais e KNN para a Identificação de Modelos ARMA Não-Sazonais e Sazonais**. [S.l.]: PUC-RJ, 2005a. Disponível em: <http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0024879_05_cap_03.pdf>. Acesso em: 10 dez. 2017.

SIULY, S.; LI, Y.; ZHANG, Y. **EEG Signal Analysis and Classification : Techniques and Applications.** [S.l.]: Springer, 2017.

SOKOLOVA, M.; JAPKOWICZ, N.; SZPAKOWICZ, S. Beyond Accuracy, F-score, and ROC: A Family of Discriminant Measures for Performance Evaluation. 2006. Disponível em: <https://vvvvw.aaai.org/Papers/Workshops/2006/WS-06-06/WS06-06-006.pdf>. Acesso em: 22 out. 2017.

TIMMARAJU, A. S.; PALNITKAR, A.; KHANNA, V. Game ON! Predicting English Premier League Match Outcomes. 2013. Disponível em: <http://cs229.stanford.edu/proj2013/TimmarajuPalnitkarKhanna-GameON!PredictionOfEPLMatchOutcomes.pdf>. Acesso em: 4 nov. 2017.

TRINDADE, A. Predicting Results of Brazilian Soccer League Matches. 2013. p. 13.

ULMER, B.; FERNANDEZ, M. Predicting Soccer Match Results in the English Premier League. 2013. p. 5. Disponível em: <http://cs229.stanford.edu/proj2014/Ben Ulmer, Matt Fernandez, Predicting Soccer Results in the English Premier League.pdf>.

VARELLA, C. A. A. ANÁLISE MULTIVARIADA APLICADA AS CIÊNCIAS

AGRÁRIAS PÓS-GRADUAÇÃO EM AGRONOMIA CIÊNCIA DO SOLO: CPGA-CS ANÁLISE DISCRIMINANTE. 2004a. Disponível em: <http://www.ufrrj.br/institutos/it/deng/varella/Downloads/multivariada aplicada as ciencias agrarias/Aulas/ANALISE DISCRIMINANTE.pdf>. Acesso em: 11 dez. 2017.

YU, L.; LIU, H. Efficient Feature Selection via Analysis of Relevance and Redundancy. **Journal of Machine Learning Research**, 2004. v. 5, p. 1205–1224. Disponível em: <http://www.jmlr.org/papers/volume5/yu04a/yu04a.pdf>. Acesso em: 22 out. 2017.

ZHANG, H. The Optimality of Naive Bayes Naive Bayes and Augmented Naive Bayes. 2004a. Disponível em: <http://www.cs.unb.ca/~hzhang/publications/FLAIRS04ZhangH.pdf>. Acesso em: 11 dez. 2017.

ZHANG, T. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. **AI Magazine**, 2001. v. 22, n. 2, p. 103–104. Disponível em: <https://pdfs.semanticscholar.org/3e82/5b4d4ca9f0405d7759c15fc10c702f26a2ec.pdf>. Acesso em: 8 dez. 2017.