UNIVERSIDADE FEDERAL DE UBERLÂNDIA

João Victor de Oliveira

Análise comparativa de algoritmos de aprendizado de máquina aplicados ao Campeonato Brasileiro de Futebol

Uberlândia, Brasil

2024

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

João Victor de Oliveira

Análise comparativa de algoritmos de aprendizado de máquina aplicados ao Campeonato Brasileiro de Futebol

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Rodrigo Sanches Miani

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2024

João Victor de Oliveira

Análise comparativa de algoritmos de aprendizado de máquina aplicados ao Campeonato Brasileiro de Futebol

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 22 de novembro de 2024:

Prof. Dr. Rodrigo Sanches Miani
Orientador

Leandro Nogueira Couto

Renato Aparecido Pimentel da Silva

Uberlândia, Brasil
2024

Agradecimentos

Agradeço aos meus pais, José Carlos e Roseli, por me proporcionarem desde cedo uma educação de qualidade, por sempre me apoiarem em uma cidade distante, independente da dificuldade, e serem minha inspiração de luta contínua e garra.

À minha irmã, Bruna, que me passou muitos aprendizados sobre a vida universitária e me ensinou muito, ajudando a continuar no meu caminho.

À República Taverna, minha casa na maior parte da minha graduação e segunda família, onde conheci amigos que levarei para sempre e vivi por completo a experiência universitária.

Aos meus amigos, Murilo, Matheus e Thiago, que estiveram presentes em diversos momentos desta caminhada, seja dando suporte nas matérias, tendo conversas sobre um punhado de coisas aleatórias ou apenas saindo para esquecer os problemas.

Aos meus professores da FACOM, por compartilharem seus conhecimentos e por me prepararem para a vida após a graduação, em especial ao meu orientador, Professor Doutor Rodrigo Miani, por permitir que eu realizasse um estudo sobre um tema que tanto me agrada, pela ajuda e pelo suporte constantes, pela paciência e pela leveza com que conduziu todo o processo.

Resumo

As predições relacionadas ao futebol despertam interesse tanto dos curiosos quanto dos profissionais desse esporte, e com o atual volume de dados disponíveis, o uso de aprendizado de máquina nesse contexto vai se tornando peça fundamental para alcançar muitos objetivos. O objetivo deste trabalho foi construir modelos de previsão para analisar os resultados do mandante em duas situações: vitória ou empate/derrota, e vitória ou derrota. Os algoritmos escolhidos para a construção dos modelos foram: Regressão Logística, *Random Forest*, *k-Nearest Neighbors*, *Naive Bayes*, Máquina de Vetores de Suporte, XGBoost e LGBM. Para aumentar a generalização e evitar o *overfitting*, os dados foram divididos em treino e teste e fez-se o uso da validação cruzada. O modelo considerado destaque no cenário sem empates foi o da Máquina de Vetores de Suporte, com acurácia de 72.65% e F1 de 82.04%. Já o modelo de destaque do cenário com empates foi o do XGBoost, com uma boa acurácia de 60.27% e um bom F1 de 64.86%. Para os modelos utilizados no cenário sem empates, os resultados encontrados indicam um desempenho acima do esperado e do que é encontrado nos modelos disponíveis publicamente, muito pelo fato dos empates terem sido removidos, o que deixou o problema de classificação mais simples e facilitou a predição dos modelos. Quando foi trocado o cenário para que tivesse empates nos dados, alguns algoritmos tiveram bons resultados como o XGB e o NB, enquanto outros tiveram desempenho abaixo do esperado, como o *k-Nearest Neighbors*.

Palavras-chave: Aprendizado de máquina, Futebol, Pré-processamento, Métricas, Desempenho da predição.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

IA	Inteligência Artificial
AM	Aprendizado de Máquina
LR	*Logistic Regression* (Regressão Logística)
RF	*Random Forest* (Floresta Aleatória)
XGB	*Extreme Gradient Boosting* (XGBoost)
NB	*Naive Bayes* (Bayesiano Ingênuo)
KNN	*k-Nearest Neighbors* (K-ésimo Vizinho mais Próximo)
SVM	*Support Vector Machine* (Máquina de Vetores de Suporte)
LGBM	*Light Gradient-Boosting Machine* (LightGBM)
MLP	*Multilayer perceptron* (Perceptron multicamadas)
SMO	*Sequential Minimal Optimization* (Otimização Sequencial Mínima)
SMOReg	*Sequential Minimal Optimization Regression* (Regressão de Otimização Sequencial Mínima)
FP	Falso positivo
FN	Falso negativo
VP	Verdadeiro positivo
VN	Verdadeiro negativo

Sumário

1 Introdução

O futebol é uma paixão nacional, se tratando de Brasil. Segundo o IBGE - Instituto Brasileiro de Geografia e Estatística (IBGE, 2017), nenhum outro esporte é mais praticado do que ele, sendo que dos 38.8 milhões de praticantes de esportes no país, 15.3 milhões o praticam, 39.3% do total. Passando para um espectro mundial, o cenário não muda, o futebol continua sendo o esporte mais popular, já que segundo a FIFA - Federação Internacional de Futebol (FIFA, 2007), considerando todos os países que são filiados à máxima entidade do futebol, o número de futebolistas, tanto homens quanto mulheres, com a inclusão de árbitros e oficiais, chega a 270 milhões.

Por ser um esporte dinâmico, onde há uma carga de imprevisibilidade nas partidas, a paixão pelo mesmo acaba sendo impulsionada, já que em muitas oportunidades, o considerado pior time ganha do melhor, ou há um inédito e improvável campeão de alguma liga, ou ainda, simplesmente, acontece a vitória de um time numa má fase em cima do maior rival. Para além do papel de torcedor, esse dinamismo faz com que muitos se arrisquem nas predições, os famosos palpites. As rivalidades históricas, a presença de um grande astro no time, a sequência positiva ou negativa de um clube, essas e diversas outras questões, podem ser consideradas pelos palpiteiros mais casuais.

Mas as predições não estão limitadas ao âmbito casual. As casas de apostas estão muito envolvidas nesse crescente uso de dados, tecnologia e matemática para a provisão das famosas *odds*, que nada mais são que as chances de que um determinado evento aconteça. Existe também o uso por parte dos clientes dessas casas, que desenvolvem modelos de previsão, que são ferramentas analíticas, e que utilizam algoritmos e técnicas estatísticas para prever eventos futuros para auxiliá-los durante a decisão de qual equipe apostar. A questão é, esse mercado vem se tornando extremamente popular no Brasil, e segundo (SIMON, 2023), o Brasil gerou o maior número de visitas aos sites no segmento de apostas esportivas no mundo, sendo que em Dezembro de 2022, houve um aumento de 75% em visitas, demonstrando o grande potencial deste mercado.

Confirmando essa veia profissional, jornalistas e programas esportivos se debruçam sobre o tema, ocupando um bom tempo nas televisões e sites, onde há até mesmo competições entre integrantes desses mesmos programas. Além disso, é foco de estudos há um tempo, como em (ARTUSO, 2008), que estudou a formulação de um modelo que estimou as pontuações necessárias para alcançar certas posições na tabela dos campeonatos brasileiros de futebol, séries A e B. Também podemos citar (HUCALJUK; RAKIPOVIć, 2011), onde foi utilizado algoritmos de aprendizagem para determinar os resultados dos jogos da competição Liga dos Campeões. Com a implementação feita com auxílio do software

Weka, o software desenvolvido neste segundo trabalho conseguiu ter com redes neurais o seu melhor desempenho, com a rede consistindo em 5 camadas ocultas, e sendo treinada pelo algoritmo de retropropagação. Um terceiro trabalho, (DEUS, 2019a), fez a predição de resultados das partidas de cinco ligas europeias, onde o objetivo dessa pesquisa era comparar algoritmos de aprendizado de máquina para o cenário esportivo, mais especificamente o futebol. Foram analisados algoritmos regressores e classificadores e concluiu-se que os destaques ficaram por conta dos algoritmos de Perceptron Multicamadas e as Máquina de Vetores de Suporte, ficando acima de 60% de Acurácia em alguns casos. Apesar dos resultados satisfatórios, todos esses estudos mencionados focam em apenas uma competição, como o primeiro e terceiro mencionados, ou não foca em uma competição brasileira, como o segundo. Além de cada um ter sua particularidade de como os modelos são construídos e suas bases de dados já existirem previamente.

Associado ao fato do trabalho ter como alvo o esporte mais popular do país, de um tema que pode gerar tamanha movimentação econômica e de dados, juntamente com a evolução de técnicas de armazenamento, processamento e distribuição de dados relacionados às partidas, as principais contribuições deste trabalho são duas. A primeira é a construção de uma base de dados com estatísticas e informações sobre as partidas do Campeonato Brasileiro de Futebol Série A, entre os anos 2018 e 2023. A segunda é a análise comparativa de algoritmos de aprendizado de máquina para predição, contendo alguns tradicionais e outros comumente utilizados em competições de predição de partidas de futebol na plataforma Kaggle.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo geral deste trabalho é avaliar os algoritmos de predição Regressão Logística, *Random Forest*, *k-Nearest Neighbors*, *Naive Bayes*, Máquina de Vetores de Suporte, XGBoost e LGBM aplicados à base de resultados de futebol, e comparar suas respectivas métricas dos resultados após utilização desses algoritmos para a predição dos resultados das partidas.

1.1.2 Objetivos específicos

- Analisar bases de dados com estatísticas sobre as partidas da Série A.
- Construir uma nova base de dados a partir das já existentes.
- Construir modelos preditivos a usando algoritmos disponíveis na biblioteca *scikit-learn* da linguagem *Python*.

- Utilizar as métricas Acurácia, Precisão, Recall e F1 para fins de comparação entre os algoritmos.

1.2 Organização do trabalho

Este trabalho é organizado da seguinte forma: o Capítulo 2 aborda conceitos básicos como Campeonato Brasileiro de Futebol, Aprendizado de Máquina, Pré-processamento de dados, todos os algoritmos utilizados nesta pesquisa e as métricas, também descreve trabalhos correlatos ao trabalho desenvolvido. O Capítulo 3 descreve os passos para o desenvolvimento da pesquisa, como a construção da base de dados, a escolha dos atributos e a construção e utilização dos modelos. O Capítulo 4 apresenta os resultados e análise dos mesmos. O Capítulo 5 traz as conclusões deste trabalho e sugestões de trabalhos futuros.

2 Revisão Bibliográfica

Neste capítulo, serão abordados conceitos básicos que estão presentes na construção deste trabalho e que são necessários para a compreensão do mesmo. Também estarão presentes trabalhos correlatos.

2.1 Campeonato Brasileiro de Futebol

Desde 2003 com o formato de pontos corridos com 2 turnos e desde 2006 com os 20 clubes que o disputam, o Campeonato Brasileiro de Futebol da Série A (ou primeira divisão) mantém esse formato até os dias atuais (SOUZA[1], 2015).

Campeonato de pontos corridos e com 2 turnos é aquele que todos os times jogam contra todos, garantindo que joguem tanto em casa quanto fora com qualquer time. A vitória conta 3 pontos para o vencedor e nenhum para o perdedor e o empate 1 ponto para cada. O time que conseguir somar mais pontos ao final de todas as rodadas será o campeão e os quatro times que somarem menos pontos, descem de divisão. É considerado um dos mais disputados, equilibrados e fortes do mundo (GE, 2023).

2.2 Aprendizado de Máquina

A expansão do uso de Inteligência Artificial (IA) em uso de soluções computacionais para problemas reais e do dia a dia acontece desde a década de 1970. E o que motivou o desenvolvimento de ferramentas que fossem mais autônomas e que não dependessem da intervenção humana para que pudessem adquirir conhecimento, foi justamente o crescimento da complexidade dos problemas a serem tratados pelas máquinas, e da velocidade e volume de dados gerados por diferentes áreas. Para a criação dessas ferramentas, na maioria dos casos, são usados os conceitos e técnicas de Aprendizado de Máquina (AM), ou seja, a maioria das ferramentas são baseadas em AM, que é uma subárea da IA que visa desenvolver o aprendizado autônomo de sistemas computacionais (FACELI et al., 2021).

Os algoritmos de AM são amplamente utilizados em diversas tarefas, e ainda segundo (FACELI et al., 2021), estas podem ser classificadas como preditivas ou descritivas. Em uma tarefa preditiva, é oferecido ao algoritmo de AM um conjunto de dados de treinamento previamente rotulados e ele irá induzir um modelo preditivo a predizer - para um novo objeto com seus atributos (característica ou aspecto de um objeto) - o valor do seu atributo alvo. Podemos ter como exemplo um modelo preditivo que, a partir de sintomas,

pode prever o estado de saúde de um paciente. Já em uma tarefa descritiva, ao invés de atribuir um valor, ela identifica, define e extrai padrões a partir de um conjunto de dados, podendo ter como principal exemplo, o agrupamento de dados, que procura grupos de objetos similares entre si.

Hierarquicamente, as categorias do Aprendizado de Máquina indutivo (aprende a realizar generalizações a partir de um conjunto de dados) e suas tarefas associadas se dividem entre algoritmos supervisionados (tarefas preditivas) e algoritmos não supervisionados (tarefas descritivas). No primeiro, as tarefas preditivas se distinguem pelo valor de rótulo a ser predito: discreto, no caso de tarefas de classificação; e contínuo, no caso de tarefas de regressão. As descritivas são divididas em: agrupamento, que dividem os dados em grupo de acordo com sua similaridade; sumarização, que buscam uma descrição simples e compacta para um conjunto de dados; e associação, que procuram padrões frequentes de associação entre os atributos de um conjunto de dados (FACELI et al., 2021).

2.3 Pré-processamento de dados

Um processo que é essencial para que se possa ter uma melhora na qualidade dos dados obtidos, melhorando, assim, a acurácia e eficácia de futuras análises que utilizam algoritmos de machine *learning* é o chamado pré-processamento de dados. Nada mais é do que um conjunto de técnicas que pode ser dividido em: limpeza de dados, integração de dados, transformação de dados e redução de dados. Esse pré-processamento é tão importante pelo fato das bases, em muitas ocasiões, conterem registros inconsistentes, faltantes, duplicados, com valores discrepantes, e mais alguns outros possíveis problemas.(SILVA, 2021).

2.3.1 Limpeza de dados

A presença de dados faltantes em conjuntos de dados é um problema comum que pode comprometer a qualidade das análises. Esses valores ausentes podem surgir em diversas etapas do processo de coleta e preparação dos dados. Para garantir a confiabilidade dos resultados, é fundamental tratar esses valores faltantes de forma adequada. As principais abordagens para lidar com essa situação incluem (SIVAKUMAR; GUNASUNDARI, 2017):

- Remoção das linhas que contenham valores faltantes.
- Substituição dos valores faltantes por estimativas, como a média ou a mediana.

2.3.2 Integração de dados

É a etapa que combina dados de múltiplas fontes e os armazenam de forma coerente, sendo que as fontes desses dedos podem incluir múltiplos bancos e até banco de arquivos simples. Alguns desafios são encontrados no processo de integração, como a redundância de dados e a inconsistência na nomenclatura de atributos (SIVAKUMAR; GUNASUNDARI, 2017).

2.3.3 Transformação de dados

É nesta etapa que os dados são transformados em formas apropriadas para o uso pelos modelos. Ela envolve o seguinte (SIVAKUMAR; GUNASUNDARI, 2017):

- Normalização: Ajusta os valores dos atributos para uma escala comum, facilitando a comparação entre diferentes variáveis.
- Suavização: Elimina o ruído presente nos dados, que pode distorcer os resultados da análise.
- Agregação: Combina dados detalhados em informações mais resumidas. Por exemplo, dados diários podem ser agrupados para obter totais mensais ou anuais.
- Generalização: Simplifica os dados, substituindo valores específicos por categorias mais amplas.

2.3.4 Redução de dados

Quando há grandes volumes de dados a serem analisados, pode ocorrer uma demora muito longa na análise dos mesmos, tornando a análise inviável. E são as técnicas de redução de dados que tornam a tarefa mais eficiente, sem comprometer a integridade dos dados originais.

Algumas abordagens para efetuar a redução de um conjunto de dados são (SIVAKUMAR; GUNASUNDARI, 2017):

- Utilizar operações de agregação de dados.
- Redução da dimensão do *dataset*, eliminando atributos irrelevantes ou redundantes.
- Compressão dos dados através de mecanismos de *encondig*.
- Redução de numerosidade de dados que substitui os dados originais por representações mais compactas, como modelos paramétricos ou não paramétricos.

2.4 Algoritmos de aprendizagem supervisionada

Nesta seção, serão apresentados os algoritmos utilizados na execução de tarefas e treinamento dos modelos utilizados, de forma supervisionada, para a predição dos resultados.

2.4.1 Regressão Logística (LR)

Quando temos uma situação em que a resposta que desejamos prever é binária, ou seja, podendo assumir apenas 2 valores, o método Regressão Logística é o mais comumente usado para estes casos (HILBE, 2011).

A LR é considerada uma técnica estatística, onde a variável dependente Y, que é o atributo que se quer prever, sendo binária, segue a distribuição de Bernoulli e tem uma probabilidade desconhecida p (FÁVERO; BELFIORE, 2017).

$$Y = \begin{cases} 1, & \text{se ocorrer sucesso} \\ 0, & \text{se ocorrer fracasso} \end{cases}$$

A probabilidade de sucesso é $0 \leq p \leq 1$ e a probabilidade de fracasso é $q = 1 - p$.

Nesse modelo, dado uma combinação linear de variáveis independentes, é feita a estimação da probabilidade desconhecida p (GONZALEZ, 2018).

2.4.2 Random Forest (RF)

Random Forest é um algoritmo de aprendizado de máquina que pertence à categoria de comitê de classificação, o que significa que ele combina múltiplos modelos de árvores de decisão para realizar previsões mais robustas e precisas (BABOOTA; KAUR, 2019).

Para reduzir a variância e aumentar a estabilidade do modelo, ele utiliza a técnica de *Bootstraping*, na qual várias amostras aleatórias (com reposição) dos dados de treinamento são usadas para treinar cada árvore (ABDULKAREEM; ABDULAZEEZ, 2021). Ao final da execução da tarefa, haverá a classificação de todas as árvores de decisão do conjunto, onde a árvore que for mais votada será a fornecida no final (ABDULKAREEM; ABDULAZEEZ, 2021).

2.4.3 XGBoost (XGB)

O XGBoost, acrônimo em inglês para *Extreme Gradient Boosting*, é uma técnica de aprendizado de máquina que faz parte de uma classe de algoritmos conhecida como comitê de classificação, mais especificamente do tipo algoritmo de *boosting*, onde a ideia

é empregar algoritmos que sozinhos não são bons para predições e transformá-los em modelos-base para a criação de um algoritmo de *boosting*, obtendo um modelo final muito mais preciso e confiável (WEERADDANA; PREMARATNE, 2021).

Em comparação aos algoritmos de *boosting*, o XGB foi projetado para melhorar a velocidade de execução e o desempenho do modelo (WEERADDANA; PREMARATNE, 2021). Algumas mudanças que proporcionaram isso foi o tratamento interno de dados desbalanceados através de técnicas de amostragem e ponderação de classes, e a utilização de solução aditiva, facilitando a otimização e interpretação do modelo (CHEN; GUESTRIN, 2016).

2.4.4 Naive Bayes (NB)

O algoritmo *Naive Bayes* é simples. Seu fluxo pode ser representado por um diagrama do tipo Grafo Acíclico Direcionado, com apenas um nó pai, que representa o nó não observado, ou seja, a variável que queremos prever, e tem vários filhos, que correspondem aos nós não observados representantes das características que influenciam o resultado do nó pai (MAGLOGIANNIS, 2007).

Baseando-se na probabilidade condicional, há nesta abordagem uma tabela de probabilidades associada a cada filho, que mostra a chance da cada nó pai acontecer de acordo com uma característica específica e que é atualizada através dos dados de treino. Além disso, quando queremos prever um novo resultado, a tabela de probabilidade de todas as características é consultada e é calculado a probabilidade mais provável para o pai. (RAY, 2019).

2.4.5 k-Nearest Neighbors (KNN)

Segundo (RAY, 2019), KNN, acrônimo em inglês para *k-Nearest Neighbors*, é um algoritmo de classificação que usa um banco de dados que possui pontos de dados agrupados em várias classes e que tenta classificar um novo ponto de dado. Para (MAGLOGIANNIS, 2007) a ideia por trás da técnica é a de que instâncias de um conjunto de dados existem geralmente na proximidade de outras instâncias que têm propriedades semelhantes, e dessa maneira quando uma instância não estiver classificada, havendo outras que estão com classes rotuladas, o valor de rótulo dessa poderá ser determinado observando a classe de seus vizinhos mais próximos.

O KNN consegue localizar as k instâncias mais próximas da instância que está como objetivo, e através da identificação da classe mais frequente, determina a classe da instância de consulta (MAGLOGIANNIS, 2007).

2.4.6 Máquina de Vetores de Suporte (SVM)

SVM tem a capacidade de executar tarefas tanto de classificação quanto de regressão (RAY, 2019). Esse algoritmo tem como ideia chave encontrar uma fronteira de decisão que maximize a margem entre as classes, ou seja, o objetivo é encontrar uma superfície que não só separa as duas classes de dados, mas também que aumente a distância entre os pontos mais próximos de cada classe, pois quanto maior a margem, menor a chance de um dado ser classificado incorretamente. Com a fronteira encontrada, fica possível a classificação dos dados (MAGLOGIANNIS, 2007).

2.4.7 LightGBM (LGBM)

Desenvolvido pela Microsoft, o LightGBM, acrônimo em inglês para *Light Gradient Boosting Machine*, é uma implementação eficiente do algoritmo de *gradient boosting*. No seu funcionamento, é criado uma série de árvores de decisão, que são como fluxogramas que ajudam tomar decisões, e posteriormente ele ajusta essas árvores para encontrar a melhor forma de separar os dados e fazer as previsões (RUFO et al., 2021). Comparado a outros algoritmos de predição, os algoritmos de *boosting* de árvores, que é o caso do LGBM, os superam (BASHA; RAJPUT; VANDHAN, 2018).

2.5 Avaliação de modelos

2.5.1 Matriz de confusão

A matriz de confusão é uma métrica em forma de tabela que tem como objetivo mostrar o desempenho de um algoritmo de classificação. Ela mostra as classificações corretas *versus* as classificações preditas para cada classe (atributo que queremos descobrir) dentro de um conjunto de exemplos dado (MATOS et al., 2009).

Considerando uma matriz de confusão de dimensão 2 x 2, com duas classes rotuladas como positivo e negativo, cada linha corresponde a uma instância da classe que representa. Já para cada coluna, é representada a previsão de cada classe. Cada elemento da matriz tem seu significado:

- Falso positivo (FP): algoritmo previu que é da classe positivo, mas é negativo.
- Falso negativo (FN): algoritmo previu que é da classe negativo, mas é positivo.
- Verdadeiro positivo (VP): algoritmo previu que é da classe positivo, e realmente é negativo.
- Verdadeiro negativo (VN): algoritmo previu que é da classe negativo, e realmente é negativo.

2.5.2 Métricas

Nesta seção serão introduzidas as métricas que usaremos para poder medir o desempenho dos modelos criados.

2.5.2.1 Acurácia

A Acurácia representa a proporção de predições corretas realizadas pelo modelo em relação ao total de predições, servindo como um indicador geral da qualidade do modelo (MARCHI; FONSECA; BODÊ, 2023).

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.1}$$

2.5.2.2 Precisão

A Precisão calcula a razão entre o número de exemplos classificados como verdadeiros positivos e a soma dos verdadeiros positivos e falsos positivos, o que evidencia a habilidade do modelo em evitar falsos positivos (MARCHI; FONSECA; BODÊ, 2023).

$$Precisão = \frac{VP}{VP + FP} \tag{2.2}$$

2.5.2.3 Recall

O Recall mede a capacidade do modelo de encontrar todos os casos positivos, reduzindo ao mínimo a ocorrência de falsos negativos (MARCHI; FONSECA; BODÊ, 2023).

$$Recall = \frac{VP}{VP + FN} \tag{2.3}$$

2.5.2.4 F1-score

A métrica *F1-score* obtém uma avaliação com melhor equilíbrio, já que ela combina Precisão e Recall em uma só pontuação, e é calculada como a média harmônica entre essas duas métricas. Ela enfatiza a capacidade do modelo ter uma predição precisa e conseguir minimizar os falsos positivos (MARCHI; FONSECA; BODÊ, 2023).

$$F1 = 2 * \frac{Precisão * Recall}{Precisão + Recall} \tag{2.4}$$

2.6 Trabalhos correlatos

Nesta seção serão apresentados alguns trabalhos correlatos à esta pesquisa, apresentando semelhanças e diferenças presentes.

2.6.1 Utilização de Aprendizado de Máquina para Previsão de Resultados de Jogos de Futebol

O objetivo da pesquisa conduzida em (DEUS, 2019b) foi comparar algoritmos de aprendizado de máquina para o cenário esportivo, mais especificamente o futebol. Foram analisados algoritmos regressores e classificadores e depois foi executado um teste de hipótese, especificamente o Teste T para comparar esses algoritmos. Sendo que os algoritmos escolhidos foram, por parte dos classificadores, o Bayesiano Ingênuo (*Naive Bayes*), por parte dos regressores, a Regressão Linear e já para ambos (classificação e regressão), foram escolhidos Redes MLP, acrônimo em inglês para *Multilayer Perceptron*, e SVM (SMO e SMOReg). Tanto Bayesiano Ingênuo quanto SVM estarão presentes neste trabalho.

Em relação aos dados, foram usadas duas bases de dados, ambas disponíveis no site Kaggle: a primeira é a da Copa do Mundo, que estão em planilhas e tem dados desde a primeira até a que aconteceu no Brasil em 2014; A segunda é a *European Soccer Database*, que contém dados de mais de 25 mil partidas de futebol das 11 principais ligas europeias, mas que nesse trabalho foram escolhidas 5 ligas para serem usadas nos testes, Liga Inglesa, Alemã, Francesa, Italiana e Espanhola com jogos de 2008 a 2016. E é aí que vem a primeira diferença,já que nesse trabalho, a única liga analisada será a Séria A do Campeonato Brasileiro.

A partir das duas bases de dados, foi feito um pré-processamento de dados, retirando dados nulos ou que não seriam relevantes para a pesquisa. Após isso, foram criadas várias versões de data sets, onde umas tinham atributos diferentes de outras, como por exemplo, atributos que priorizavam mais resultados a longo prazo, outras que priorizavam a curto prazo. Também foram adicionados alguns atributos relacionados à casa de apostas, referentes às taxas de aposta para vitória, empate e derrota, indo de encontro ao que essa pesquisa também pretende fazer.

Para auxiliar na modelagem e análise de dados, além do treinamento e execução de testes com os algoritmos mencionados, o software Weka foi o escolhido. Após todo o processo e execução dos testes, sendo obtidos os resultados, foi feita, também através do Weka, uma comparação de algoritmos, dois a dois, com o teste de hipótese bicaudal denominado Teste T. E mesmo não sendo possível comparar todos de uma vez, foi possível verificar que SMO tende a ser melhor que Regressão Linear e que MLP e SMO tendem a ter melhor desempenho comparados ao Bayesiano Ingênuo.

2.6.2 Avaliação dos Modelos de Machine Learning: Verificando os Resultados no Futebol

O objetivo do trabalho proposto por Conceição (2022) é fazer a predição de forma correta dos resultados dos jogos da Série A do Campeonato Brasileiro, o que se assemelha muito a um dos objetivos dessa pesquisa.

Com relação ao conjunto de dados, os autores utilizaram os dados dos campeonatos brasileiros Série A dos anos 2015 a 2021, advindos do site Football Data UK [1], que foram subdivididos em subconjuntos de treinamento e teste. Os dados dos anos de 2015 a 2018 do campeonato ficaram exclusivos para treinamento dos modelos e os três anos restantes 2019 a 2021 foram os dados utilizados para testar a eficácia das previsões dos algoritmos selecionados por esta pesquisa. E para definir quais eram os melhores algoritmos para aturarem como classificadores desses modelos, foi utilizado os coeficientes Gini Ratio e Gini.

Sobre os algoritmos escolhidos, estão presentes *Random Forest*, *k-Nearest Neighbors*, *Naive Bayes*, Regressão Logística, *Gradient Boost* e *Decision Tree*, onde os quatro primeiros também estarão presentes nesse trabalho.

Para fazer todo o processo, foi utilizado o Orange Data Mining, que é um software de código aberto que possui um conjunto de ferramentas que permite a visualização de dados, o aprendizado de máquina e a mineração de dados. Algo diferente do que foi proposto para esse trabalho se encontra no modo em que os dados serão tratados e como será construído os modelos para predição, já que no trabalho proposto aqui, será utilizado a linguagem *Python* juntamente com algumas bibliotecas como *scikit-learn* e pandas para que isso seja feito.

Em relação aos resultados, foram obtidos em média 46% de acertos dos resultados de previsão dos modelos analisados e as principais variáveis envolvidas foram AvgA (probabilidade média de vitória em casa) e MaxH (probabilidade de vitória time da casa).

2.6.3 Prediction of football match results with Machine Learning

Diferentemente do presente trabalho, a que se encontra em Rodrigues e Pinto (2022) analisa as cinco temporadas completas do campeonato inglês de futebol, a Premiere League, das temporadas 2013/2014 até 2018/2019. Além dos atributos básicos advindos da base de dados, também foram incorporados ao conjunto de dados os atributos relacionados à nota de características individuais dos jogadores e gerais dos times, advindas do jogo eletrônico de futebol FIFA, através do site Sofifa [2], e atributos relacionados a apostas, já

[1] https://www.football-data.co.uk/
[2] https://www.sofifa.com

que um outro objetivo da pesquisa citada é que o modelo de previsão seja incluído em um sistema de suporte à decisão de apostas.

Durante a etapa de análise e processamento de dados, foi usada descrição de dados, tratamento de dados e por último, a exploração de dados. Antes de testar os diferentes algoritmos, os dados foram normalizados usando o método *z-score* para eliminar o efeito de grandes variações nos valores, e só então as variáveis mais importantes para prever os resultados foram identificadas através do algoritmo Boruta.

Em relação aos algoritmos, os testados foram: *Naive Bayes*, *k-Nearest Neighbors*, *Random Forest*, Máquina de Vetores de Suportes (SVM), XGBoost, Redes Neurais Artificiais (RNA), C5.0 e Regressão Logística Multinomial, onde os cinco primeiros são usados nessa pesquisa. Os algoritmos que permitiram alcançar os melhores resultados nos diferentes casos foram SVM, *Random Forest*, Xgboost e RNA. O melhor modelo utilizou o algoritmo *Random Forest*, que obteve uma taxa de acerto de 65,26% e uma margem de lucro de 26,74% (relacionado aos acertos das apostas).

3 Desenvolvimento

Neste capítulo serão apresentadas as etapas que permitem a classificação e avaliação dos modelos de previsão para dados coletados da Série A do Campeonato Brasileiro.

3.1 Descrição geral do trabalho

As etapas do trabalho foram divididas da seguinte forma:

1. Seleção do conjunto de dados a ser usado e coleta: busca das bases de dados, análise das mesmas, seleção das mais apropriadas para o objetivo do estudo e escolha de quais atributos são mais importantes, para posteriormente coletar os dados e construir a base a ser utilizada.
2. Análise exploratória: compreensão dos dados, permitindo tomar decisões mais assertivas de como tratar os dados.
3. Pré-processamento e tratamento dos dados: tratamento dos dados para retirar valores nulos, remover possíveis inconsistências e outras modificações que facilitem a manipulação e uso pelos modelos.
4. Modelagem de dados: divisão do conjunto de dados entre treino e teste, criação dos modelos preditivos com os algoritmos escolhidos e treinamento dos modelos com o *dataset* escolhido.
5. Avaliação dos modelos: teste dos modelos e uso de métricas para poder avaliar os resultados obtidos após o teste.

Nas Seções 3.2, 3.3, 3.4, 3.5 e 3.6, serão explicadas com mais detalhes cada uma das etapas mencionadas acima.

3.2 Seleção do conjunto de dados

Ao longo da primeira etapa, foram analisadas diversas possibilidades de bases de dados que poderiam servir de base para a construção do conjunto de dados final que usaríamos neste estudo. Considerou-se o conteúdo e a quantidade das informações pré-jogo, ou seja, apenas bases que possuíam informações e estatísticas relevantes relacionadas a um contexto anterior ao início do jogo foram consideras e posteriormente selecionadas.

A primeira base de dados investigada foi a *Brasileirão Série A* do site Base dos Dados [1], que traz informações como número de gols, estádio, árbitro, técnicos, e mais outras interessantes, mas o que mais chamou atenção foram três informações que não são tão facilmente encontradas nas bases buscadas: as colocações dos times logo antes do confronto acontecer, os valores dos times que entraram em campo e a idade média dos jogadores titulares, sendo que todas essas informações foram aplicadas tanto para o time mandante quanto para o visitante. Quase todas as informações datam do ano de 2003 até o ano de 2024, podendo a base ser aumentada ano a ano segundo a descrição da tabela encontrada no site.

A segunda escolhida foi a *Campeonato Brasileiro de Futebol*. Disponibilizada no site Kaggle [2], ela contém algumas informações abrangentes sobre o dia do jogo, como técnicos, formações utilizadas pelos times e estados aos que os times envolvidos na disputa pertencem. Sobre a extensão das datas, a base é composta por dados advindos desde a primeira edição do campeonato brasileiro Série A, em 2003, até o ano de 2023.

A terceira base de dados escolhida foi a *Brazil Serie A Matches*, também datada de 2003 até 2023, advinda do site FootyStats [3], um grande site de estatísticas de futebol do mundo. Neste *dataset* existem informações relevantes para o critério previamente estabelecido da escolha das bases, que são os dados de pré-jogo, dentre eles: média de pontos ganho por jogo pelo time, o cálculo da expectativa de gols do time, média de gols por jogo entre os times, e outras mais tão importantes quanto as citadas. Além disso, por haver uma forte participação de atributos relacionados a apostas esportivas, como, por exemplo, a chance calculada por casas de apostas para haver mais de dois gols na partida, ou então a chance do time visitante vencer o jogo, esse conjunto de dados foi o que mais forneceu atributos para a base de dados final. Pelo futebol possuir vários fatores que podem contribuir para o resultado de uma partida, como o momento do time e a fase dos jogadores, fica difícil quantificá-los numericamente, e recorrendo a essas estatísticas das casas de apostas, que conseguem condensar muitas informações, podemos ter dados possivelmente mais relevantes no momento dos modelos os utilizarem.

A base de dados construída neste trabalho, e que pode ser usada para o treinamento e teste dos algoritmos, foi chamada de *Informações e estatísticas - Brasileirão Série A*, contendo a primeira versão da tabela, nomeada de "brasileirao_serie_a_v1". Nesta tabela, há a junção de dados das 3 bases citadas acima, e foi construída no formato mostrado pela Tabela 1.

Uma observação a ser destacada é que, antes de realizar a análise exploratória, identificou-se que os atributos relacionados à idade e ao valor dos times apresentavam

[1] https://basedosdados.org/
[2] https://www.kaggle.com/
[3] https://footystats.org/

dados inconsistentes e incompletos antes de 2018. Assim, optou-se por utilizar o intervalo entre 2018 e 2023 para a construção da análise.

Nº	**Nome do atributo**	**Descrição**
1	ano_campeonato	ano corrente do Campeonato Brasileiro de Futebol Série A
2	mes_campeonato	mês corrente da partida
3	data	data da partida
4	rodada	número da rodada da partida
5	time_mandante	nome do time mandante
6	time_visitante	nome do time visitante
7	estadio	nome do estádio da partida
8	PPJ_pre_jogo_mandante	média de pontos por jogo do time mandante antes da partida começar
9	PPJ_pre_jogo_visitante	média de pontos por jogo do time visitante antes da partida começar
10	xG_pre_jogo_mandante	gols esperados da equipe mandante
11	xG_pre_jogo_visitante	gols esperados da equipe visitante
12	GPJ_pre_jogo	média de gols por jogo antes da partida começar
13	AM_porcentagem_pre_jogo	média, antes do jogo começar, da probabilidade de ambas as equipes marcarem
14	A15_porcentagem_pre_jogo	média, antes do jogo começar, da probabilidade entre ambas equipes marcarem mais de 1.5 gols na partida
15	A25_porcentagem_pre_jogo	média, antes do jogo começar, da probabilidade entre ambas equipes marcarem mais de 2.5 gols na partida
16	A45_porcentagem_pre_jogo	média, antes do jogo começar, da probabilidade entre ambas equipes marcarem mais de 4.5 gols na partida
17	EPJ_pre_jogo	média do número de escanteios das duas equipes antes do jogo começar
18	odds_mandante_vence	média das odds, no pré-jogo, entre casas de apostas para que o time mandante vença a partida

Nº	**Nome do atributo**	**Descrição**
19	odds_empate	média das odds, no pré-jogo, entre casas de apostas para que acabe com empate a partida
20	odds_visitante_vence	média das odds, no pré-jogo, entre casas de apostas para que o time visitante vença a partida
21	odds_A15	média das odds, no pré-jogo, entre casas de apostas para que haja mais de 1.5 gols no jogo
22	odds_A25	média das odds, no pré-jogo, entre casas de apostas para que haja mais de 2.5 gols no jogo
23	odds_A35	média das odds, no pré-jogo, entre casas de apostas para que haja mais de 3.5 gols no jogo
24	odds_A45	média das odds, no pré-jogo, entre casas de apostas para que haja mais de 4.5 gols no jogo
25	odds_AM_sim	média das odds, no pré-jogo, entre casas de apostas para que ambas as equipes marquem pelo menos um gol cada na partida
26	odds_AM_nao	média das odds, no pré-jogo, entre casas de apostas para que, no máximo, apenas uma equipe marque gol
27	formacao_mandante	formação tática do time mandante que começa a partida
28	formacao_visitante	formação tática do time visitante que começa a partida
29	estado_mandante	estado brasileiro de origem do time mandante
30	estado_visitante	estado brasileiro de origem do time visitante
31	colocacao_mandante	colocação na tabela do campeonato do time mandante antes do jogo começar
32	colocacao_visitante	colocação na tabela do campeonato do time visitante antes do jogo começar
33	valor_equipe_titular_mandante	valor financeiro da equipe titular do time mandande para a partida

Nº	Nome do atributo	Descrição
34	valor_equipe_titular_visitante	valor financeiro da equipe titular do time visitante para a partida
35	idade_media_titular_mandante	média da idade da equipe titular do time mandante para a partida
36	idade_media_titular_visitante	média da idade da equipe titular do time visitante para a partida
37	vencedor	mostra o resultado da partida, que pode ser mandante venceu, ou visitante venceu ou empate.

Tabela 1 – Atributos do *dataset* e suas descrições

3.3 Análise exploratória

A análise exploratória pode ser entendida como um conjunto de métodos adequados para a exploração e interpretação de um conjunto de dados numéricos, tendo como maior objetivo o aumento do conhecimento do pesquisador sobre os dados para poder identificar padrões de interesse e criar representações que destaquem esses padrões (LOPES et al., 2019).

O primeiro passo desta etapa foi identificar as características gerais do conjunto de dados que foi construído. Analisando a tabela "brasileirao_serie_a_v1", o número de registros é de 2046 e compreende 37 atributos referentes a cada partida de futebol do Brasileirão Série A. Também foi identificado a distribuição dos atributos alvos, presentes na Figura 1

Posteriormente foi descoberto se existiam instâncias com atributos vazios, e quantos e quais eram esses atributos, como evidenciado na Tabela 2 a seguir.

Atributo	Número de valores vazios
estádio	53
valor_equipe_mandante	23
valor_equipe_visitante	23
idade_media_titular_mandante	23
idade_media_titular_visitante	23

Tabela 2 – Atributos com valores vazios

Por último, foram identificadas algumas inconsistências e erros nos nomes dos estádios, nas idades médias dos times e nos valores dos times. Existiam alguns estádios que,

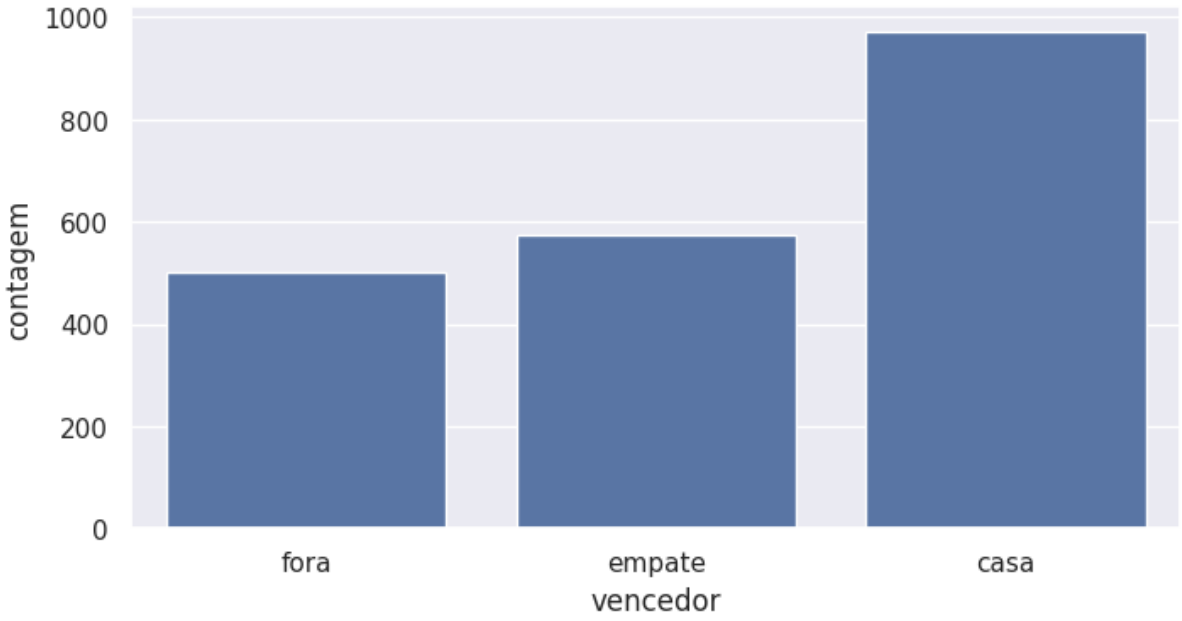

Figura 1 – Distribuição de valores do atributo alvo.

apesar de nomes diferentes, significavam, na prática, o mesmo estádio. Em relação as idades, haviam valores na casa da centena, o que é, no mínimo, estranho. Foi descoberto que estes valores estavam dessa forma pelo motivo da falta do ponto flutuante para delimitar a idade como dezena. Já nos valores dos times, haviam algumas instâncias com valores muito baixos que não eram compatíveis com o valor real de uma equipe profissional de futebol da Série A do campeonato brasileiro.

3.4 Pré-processamento

A fim de obter o melhor desempenho e o funcionamento correto dos algoritmos e aplicar as possíveis melhorias encontradas na etapa da análise exploratória, é nesta etapa que foram feitas as mudanças necessárias na base de dados utilizada.

As seguintes melhorias e ajustes foram aplicados:

1. Dados nulos: todas as instâncias que continham valores desse tipo foram removidas.
2. Idades: os atributos idade_media_titular_mandante e idade_media_titular_visitante que estavam com valores na casa das centenas em algumas instâncias foram ajustados para serem mostrados corretamente, colocando o ponto flutuante no local correto e transformando novamente a idade em dezena.
3. Valor do time: todas as instâncias que continham times avaliados abaixo de um valor delimitado, que para este estudo foi R$1.000.000,00 (um milhão de reais),

foram removidas. Esse foi um valor estipulado após pesquisa de valores de mercado dos times de cada edição do campeonato, de 2018 até 2023.

4. Estádios: padronização do nome dos estádios, removendo a discrepância antes existida, onde o mesmo estádio tinha mais de um nome correspondente no atributo estadio.

5. Divisão de *datasets*: A partir do *dataset* contendo todas as mudanças e melhorias citadas acima, foram criados dois a mais para serem usados. A abordagem adotada para o primeiro foi a de retirar as instâncias em que o vencedor, atributo alvo, era o empate. Já a abordagem adotada para o segundo foi a de manter os empates. Dessa forma, foi possível ter dois cenários diferentes para realizar os treinos e testes dos algoritmos.

6. Transformação dos dados: Como o estudo proposto faz o uso de um atributo alvo com valores categóricos, para se adequar aos algoritmos classificadores utilizados e ter um melhor desempenho dos mesmos, ocorreu a substituição dos valores do atributo vencedor que eram categóricos por numéricos. Dessa maneira, utilizando como exemplo o conjunto de dados sem empates, a previsão se moldaria ao fato do time da casa ganhar, representado pelo 1, ou perder, representado pelo 0. Já utilizando como exemplo o conjunto com empates, a previsão se moldaria ao fato do time da casa vencer, representado pelo 1, ou de não vencer, podendo ser empate ou derrota, representado pelo 0. Ainda na questão de adaptar os *datasets* aos algoritmos classificadores, também foi utilizada a técnica de codificação *one-hot*, que nada mais é que transformar atributos categóricos em atributos numéricos binários, mas diferentemente do que foi explicado anteriormente, não apenas do atributo alvo, mas sim de todos os categóricos presentes nos conjuntos de dados, facilitando a interpretação dos dados e tornando os conjuntos de dados utilizados compatíveis com todos os algoritmos usados no estudo. As representações das transformação realizadas nos conjuntos de dados são mostradas nas Tabelas 3 e 4. Na Figura 2 é ilustrado um exemplo de como funciona a codificação *one-hot*.

	Atributo *vencedor*	**Atributo *vencedor***
Valor antes da transformação	casa	fora
Valor depois da transformação	1	0

Tabela 3 – Atributo *vencedor* transformado - conjunto de dados sem empate

3.5 Modelagem de dados

Nesta quarta etapa é feita a separação do atributo *vencedor* do resto dos atributos do conjunto de dados, a divisão dos dados entre treino e teste, a criação dos modelos

	Atributo *vencedor*	**Atributo *vencedor***	**Atributo *vencedor***
Valor antes da transformação	casa	fora	empate
Valor depois da transformação	1	0	0

Tabela 4 – Atributo *vencedor* transformado - conjunto de dados com empate

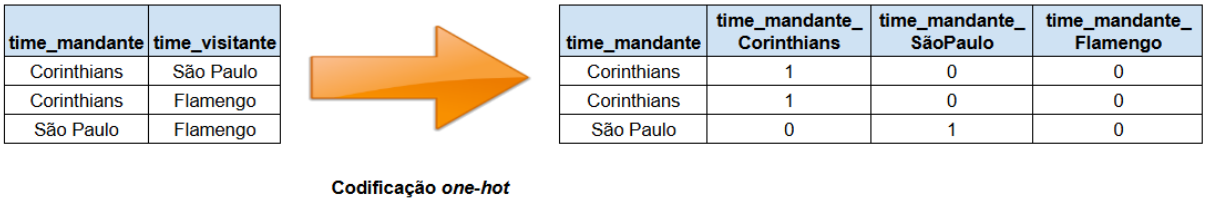

Figura 2 – Funcionamento da codificação *one-hot*.

preditores de acordo com cada algoritmo e avaliação dos modelos criados com os dados de treino.

Para que seja possível simular um cenário real, onde o resultado a ser previsto não é conhecido, acontece a separação do atributo alvo do resto do conjunto de dados, evitando que o modelo que vai predizer tenha conhecimento do resultado esperado e consiga mapear os atributos para o alvo de forma generalizada, sendo capaz de fazer novas previsões onde o alvo é desconhecido.

Já a divisão do conjunto de dados é feita justamente para que, primeiro, seja possível ensinar os modelos a identificar padrões e relação entre as variáveis através do uso dos dados de treino, e segundo, utilize os dados de teste para poder avaliar a performance do modelo em dados que ele nunca viu antes. Neste estudo, a divisão seguiu a proporção de 70% do conjunto de dados utilizado para treino e 30% utilizado para teste dos modelos, uma estratégia comum utilizada no meio de predições utilizando aprendizado de máquina. E para que fosse possível a divisão, foi utilizada uma função chamada *train_test_split* da biblioteca *scikit-learn* da linguagem *Python*, que é uma função que faz esta divisão de forma aleatória, garantindo que ambos os conjuntos sejam representativos do conjunto original.

Sobre a criação dos modelos, foram utilizados os algoritmos Regressão Logística, *Random Forest*, *KNeighborsClassifier*, que é um classificador que implementa o *k-Nearest Neighbors*, *Naive Bayes* com distribuição gaussiana, Máquina de Vetores de Suporte, XGBoost e LightGBM para esta tarefa. Tendo como foco a comparação entre seus respectivos modelos, todos foram usados na forma padrão, sem ajustes de hiperparâmetros. Mais uma vez o *scikit-learn* foi usado, desta vez para a criação dos modelos utilizando os algoritmos presentes em seu acervo de aprendizado de máquina.

A avaliação dos modelos com dados de treino foi feita para se ter uma ideia inicial

do desempenho dos mesmos. Utilizando uma técnica de validação cruzada, mais precisamente o *K-fold*, foi possível avaliar os modelos, analisar se estaria ocorrendo sobreajuste (modelo se ajusta demais aos dados de treinamento, perdendo a capacidade de generalizar para novos dados) e evitar o mesmo, e ter uma avaliação mais robusta graças a forma que essa técnica funciona. Especificamente no *K-fold*, os dados são divididos em k partes iguais, sendo que em cada iteração, uma parte é utilizada como conjunto de testes e as partes restantes são utilizadas para treino.

3.6 Avaliação e comparação dos modelos

A etapa final, que é de fato a predição do atributo alvo *vencedor* e a avaliação de todos os modelos criados utilizando os dados de teste, acontece para que possamos visualizar o desempenho de cada modelo simulando uma situação real em que eles não têm conhecimento dos dados a serem preditos. Após os modelos atuarem, são utilizadas matrizes de confusão, além das métricas Acurácia, Precisão, Recall e *F1-score* para medir o desempenho de cada modelo e possibilitar que haja comparações entre os modelos utilizados, o que será devidamente exposto no próximo capítulo.

4 Resultados

Neste capítulo, são apresentados os resultados obtidos por meio da execução de cada um dos algoritmos em ambos os conjuntos de dados, considerando o que foi criado sem empates e o com empates.

4.1 Cenário sem empates

Nesta seção, a partir das métricas, é possível ter insumos para poder discutir e comparar os resultados de cada modelo. Visando reduzir a complexidade do problema e ter um foco maior na vantagem do mandante da partida e o impacto do fator casa, essa abordagem nos traz os resultados apontados nas subseções 4.1.1 e 4.1.2.

4.1.1 Treinamento

Como dito anteriormente, a abordagem utilizada para o treinamento dos modelos foi o uso da validação cruzada, com 10 *folds*. Dessa forma, na Tabela 5, é mostrado como se distribui os resultados das métricas, incluindo o desvio padrão, onde é possível obter uma visão inicial do desempenho dos modelos. Os valores em negrito são os melhores resultados.

Algoritmo	**Acurácia**	**Precisão**	**Recall**	**F1**
LR	0.68 ± 0.04	0.70 ± 0.05	0.89 ± 0.04	$\mathbf{0.78 \pm 0.03}$
RF	0.66 ± 0.06	0.70 ± 0.01	0.87 ± 0.04	0.77 ± 0.05
KNN	0.64 ± 0.02	0.70 ± 0.03	0.78 ± 0.08	0.73 ± 0.03
NB	0.67 ± 0.06	0.69 ± 0.05	$\mathbf{0.91 \pm 0.04}$	$\mathbf{0.78 \pm 0.03}$
SVM	0.67 ± 0.04	0.68 ± 0.04	0.90 ± 0.04	0.77 ± 0.04
XGB	$\mathbf{0.70 \pm 0.04}$	$\mathbf{0.72 \pm 0.05}$	0.86 ± 0.04	0.77 ± 0.04
LGBM	0.66 ± 0.04	0.70 ± 0.05	0.81 ± 0.05	0.74 ± 0.03

Tabela 5 – Desempenho dos modelos preditivos utilizando dados de treinamento - cenário sem empates

Conforme pode ser visto na Tabela 5, os melhores resultados de Acurácia foram alcançados utilizando o modelo construído a partir do XGBoost, chegando a 70%, e que também ficou com o melhor resultado considerando a Precisão, atingindo um valor de 72%. Outro modelo que se destacou foi o baseado no algoritmo *Naive Bayes*, atingindo um Recall de 91% e um *F1-score* de 78%. Foi possível observar também que o modelo advindo da Regressão Logística igualou o índice de F1 do *Naive Bayes*.

4.1.2 Teste

A fase do teste é o momento que acontece a utilização dos modelos previamente criados e treinados, mas agora com um conjunto de dados para teste. Dessa maneira, é possível que haja uma avaliação mais assertiva, já que, como falado anteriormente, são dados desconhecidos pelo modelo.

Assim como as métricas foram calculadas após o treinamento, o mesmo será feito aqui, com a diferença de que será adicionada a matriz de confusão. É através dela que os falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos são identificados. Na Figura 3, é possível ver o conjunto de matrizes de confusão que compreende todos os modelos utilizados para predição. Já na Figura 4 temos um gráfico de barras, que mostra todas as métricas utilizadas divididas em grupos de algoritmos, objetivando uma maior facilidade de compreensão e comparação dos mesmos.

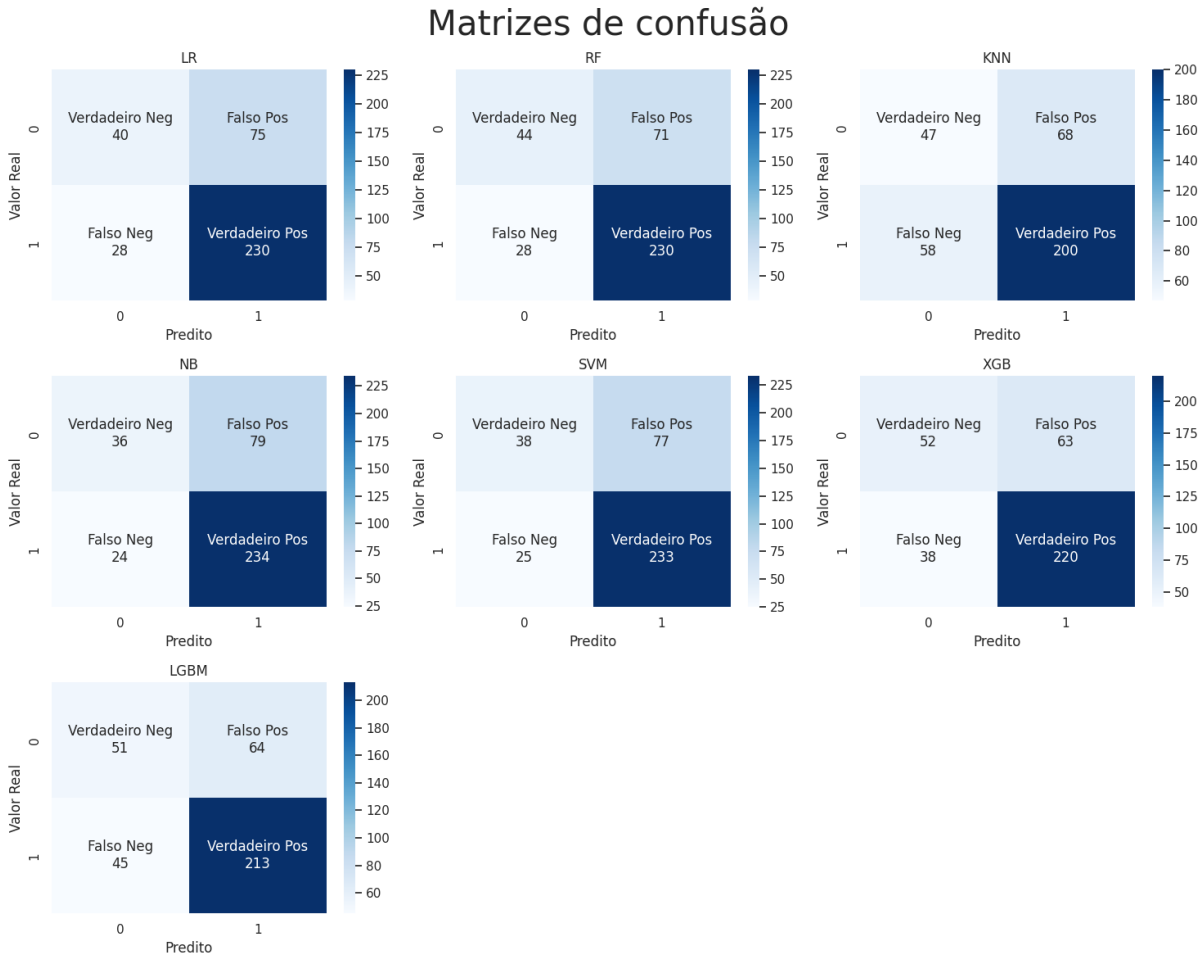

Figura 3 – Conjunto de matrizes de confusão dos modelos - cenário sem empates.

Observando as matrizes da Figura 3, é possível notar que o modelo advindo do algoritmo *Naive Bayes* foi o que mais apresentou casos de verdadeiros positivos, com valor de 234. Esses 44.91% dos casos totais fazem dele o modelo que mais acerta ao prever que

o time da casa ganha. Já o modelo do XGBoost obteve 52 casos de verdadeiros negativos, 9.98%, sendo o mais eficiente em prever resultados corretos quando o time de fora ganha ou há empate. O modelo baseado no *Naive Bayes* registrou o maior número de falsos positivos: 79, correspondendo a 15,16%. Isso o torna o modelo que mais frequentemente classifica como vitória do time da casa quando deveria ser fora ou empate. Por fim, com 58 falsos positivos, e uma porcentagem de 11.13% do total de casos, o modelo do KNN foi o que obteve o maior valor e, consequentemente, o que mais deixou de prever corretamente quando o time de casa venceu o jogo.

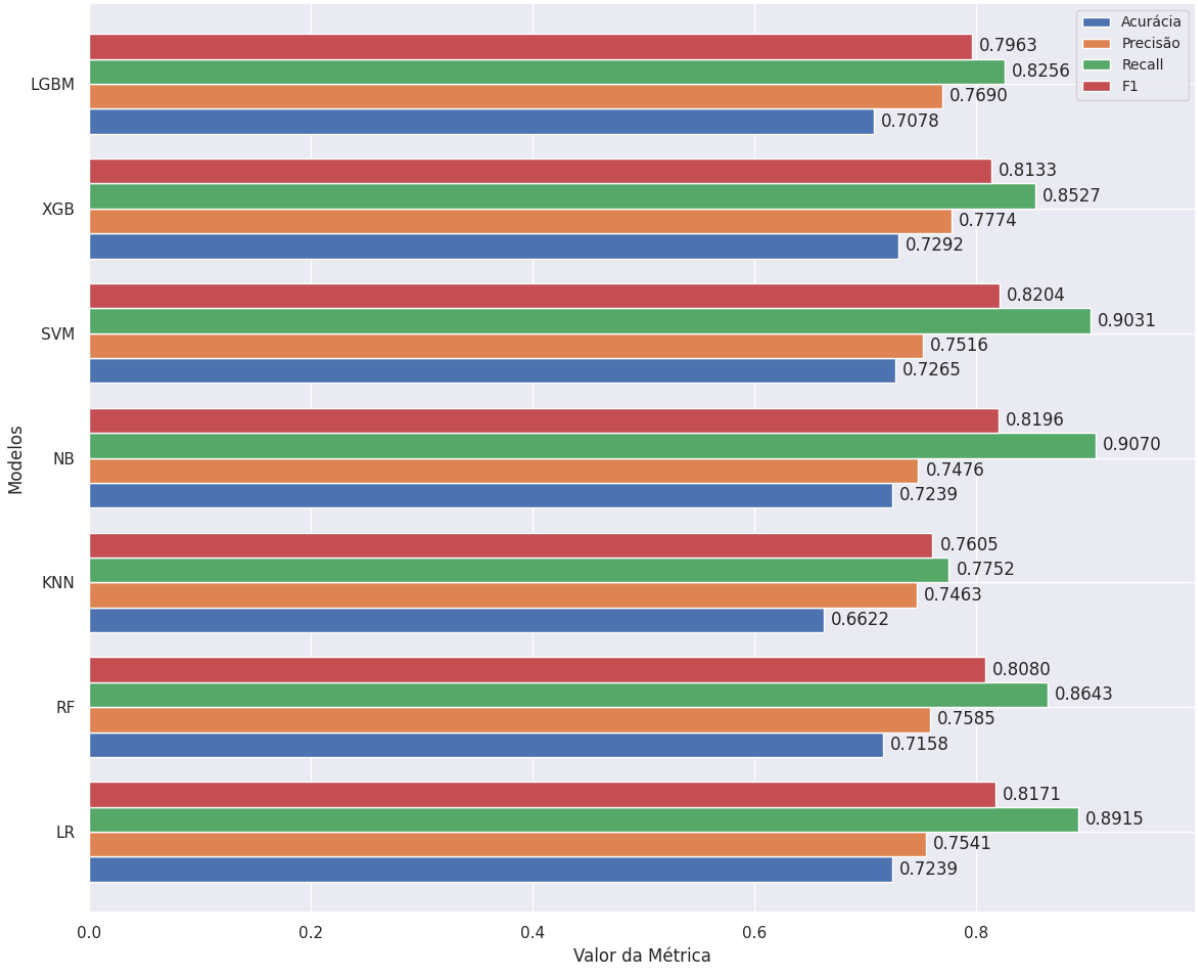

Figura 4 – Gráfico de barra das métricas dos modelos - cenário sem empates.

Em relação a Acurácia e Precisão, analisando a Figura 4, o modelo que mais se destacou foi o baseado no XGBoost, alcançando, respectivamente, os valores de 72.92% e 77.74%. Já para a Recall, o modelo do *Naive Bayes* foi o que mais se destacou, atingindo o valor de 90.70%. Por último, analisando a métrica F1, o modelo do SVM garantiu o valor de 82.04%, sendo o maior dentre os modelos.

4.2 Cenário com empates

Dentro desta seção, as subseções 4.2.2 e 4.2.3 contém os resultados obtidos tanto para treinamento quanto para validação dos testes. Nesta abordagem, o objetivo maior foi manter a capacidade do modelo de refletir a realidade quase completa do campeonato, pois, mesmo que ainda continue um problema binário, ele não deixa de aprender sobre o que caracteriza um jogo que termina empatado, diferentemente do cenário sem empates.

4.2.1 Treinamento

Assim como no treinamento do cenário sem empates, o com empates também utiliza a validação cruzada para poder calcular as métricas que darão uma visão inicial do desempenho dos modelos. Na Tabela 6, temos os algoritmos e seus respectivos resultados das métricas utilizadas, também contando com o desvio padrão de cada uma.

Algoritmo	**Acurácia**	**Precisão**	**Recall**	**F1**
LR	$\mathbf{0.63 \pm 0.03}$	0.62 ± 0.07	0.50 ± 0.05	0.56 ± 0.04
RF	0.60 ± 0.05	0.61 ± 0.05	0.49 ± 0.05	0.54 ± 0.03
KNN	0.58 ± 0.03	0.56 ± 0.07	0.55 ± 0.06	0.55 ± 0.03
NB	0.61 ± 0.04	$\mathbf{0.62 \pm 0.03}$	0.52 ± 0.05	0.56 ± 0.06
SVM	0.61 ± 0.04	0.61 ± 0.06	0.49 ± 0.04	0.54 ± 0.08
XGB	0.62 ± 0.03	0.57 ± 0.06	$\mathbf{0.58 \pm 0.21}$	$\mathbf{0.58 \pm 0.10}$
LGBM	0.59 ± 0.04	0.56 ± 0.04	0.56 ± 0.07	0.55 ± 0.04

Tabela 6 – Desempenho dos modelos preditivos utilizando dados de treinamento - cenário com empates

Conforme pode ser visto na Tabela 6, os melhores resultados de Acurácia foram alcançados utilizando o modelo construído a partir da Regressão Logística, com o valor de 63%. Na Precisão, a mais alta taxa foi de 62%, pertencendo ao modelo do *Naive Bayes*. Já para a Recall, o valor mais alto foi registrado pelo modelo do XGBoost, com 58%. Por fim, o F1 também teve sua taxa mais alta como 58% e também do modelo advindo do XGBoost. Uma observação que pode ser feita sobre esses dois últimos valores é que, apesar de serem os maiores em porcentagem, o desvio padrão também foi muito mais alto comparado aos outros valores de Recall e F1 calculados.

4.2.2 Teste

Toda a lógica aplicada ao teste do cenário sem empates, também foi aplicada aqui. A validação dos modelos segue o mesmo padrão, mas os resultados das métricas são diferentes, afinal, foi adicionado o empate a esse cenário. É possível observar estes resultados através das Figura 5 e Figura 6, que representam, respectivamente, as matrizes de confu-

são dos modelos envolvidos no estudo e o gráfico de barras que contém os resultados das métricas calculadas.

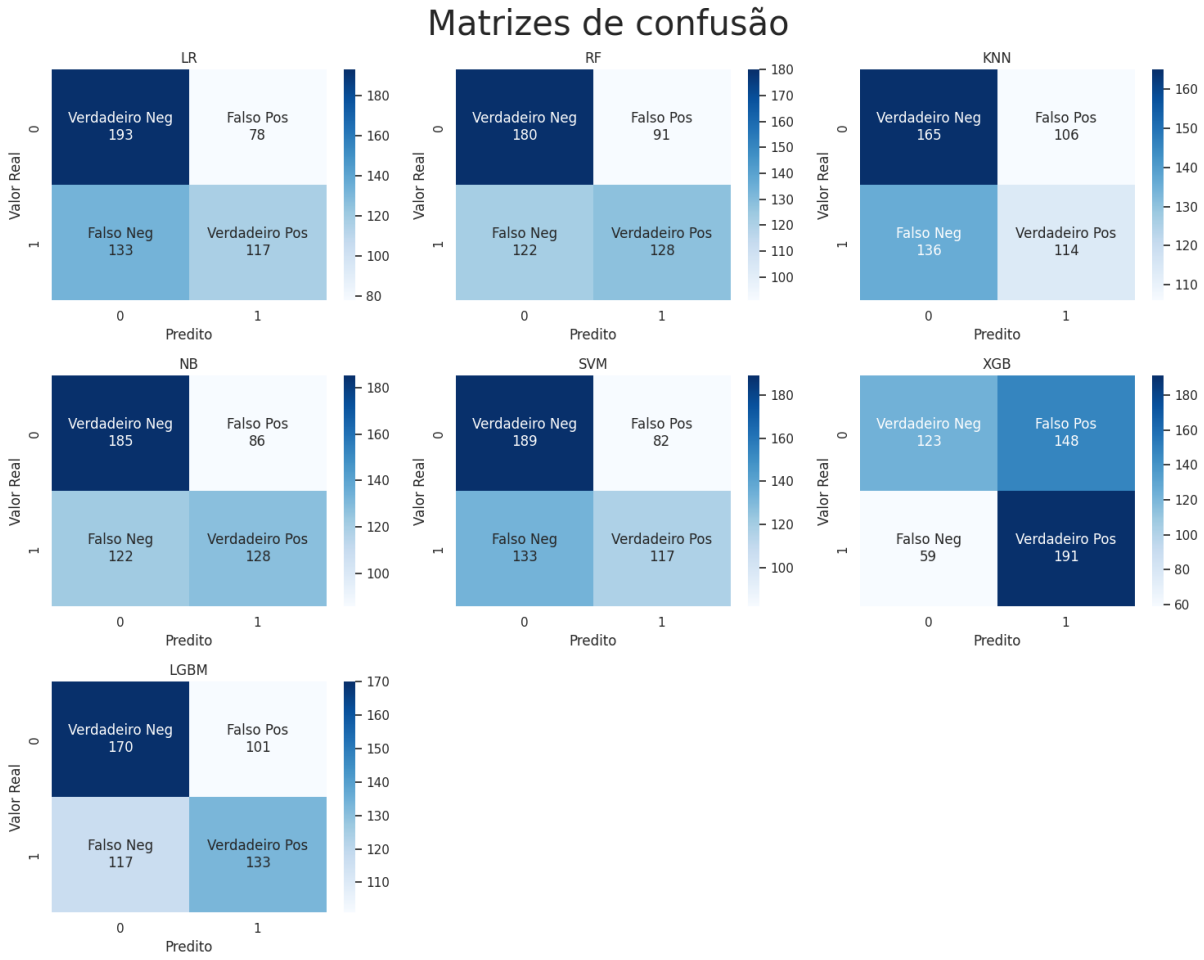

Figura 5 – Conjunto de matrizes de confusão dos modelos - cenário com empates.

A partir da Figura 5, é possível constatar que o modelo criado a partir do XGBoost foi o que mais apresentou casos de verdadeiros positivos, com 191, o que equivale 36.6% do total de casos. Ele é o que previu corretamente a maior quantidade de vitórias do time da casa. E com 194 casos, 37.23%, o modelo da Regressão Logística foi o que mais ocorreu verdadeiros negativos, sendo o mais eficiente em prever quando o time da casa não ganha. Já o modelo com mais falsos positivos foi o do XGBoost, com 148, equivalente a 28.41% do total, e foi o modelo que mais errou ao prever vitória do time da casa, onde o resultado real foi o time de fora ganhar ou empate. E o modelo com mais falsos negativos foi o do KNN, com 136 incidências, 26.10%, significando que é o que mais classifica incorretamente o vencedor como o time de fora ou empate quando, na verdade, deveria ser vitória do time da casa.

Através de uma análise feita da Figura 6, é tido que o modelo que mais se destacou, considerando a métrica Acurácia, foi o baseado no XGBoost, alcançando o valor de 60.27%. Considerando agora a Precisão, a maior taxa obtida foi de 60% cravados, pertencendo ao

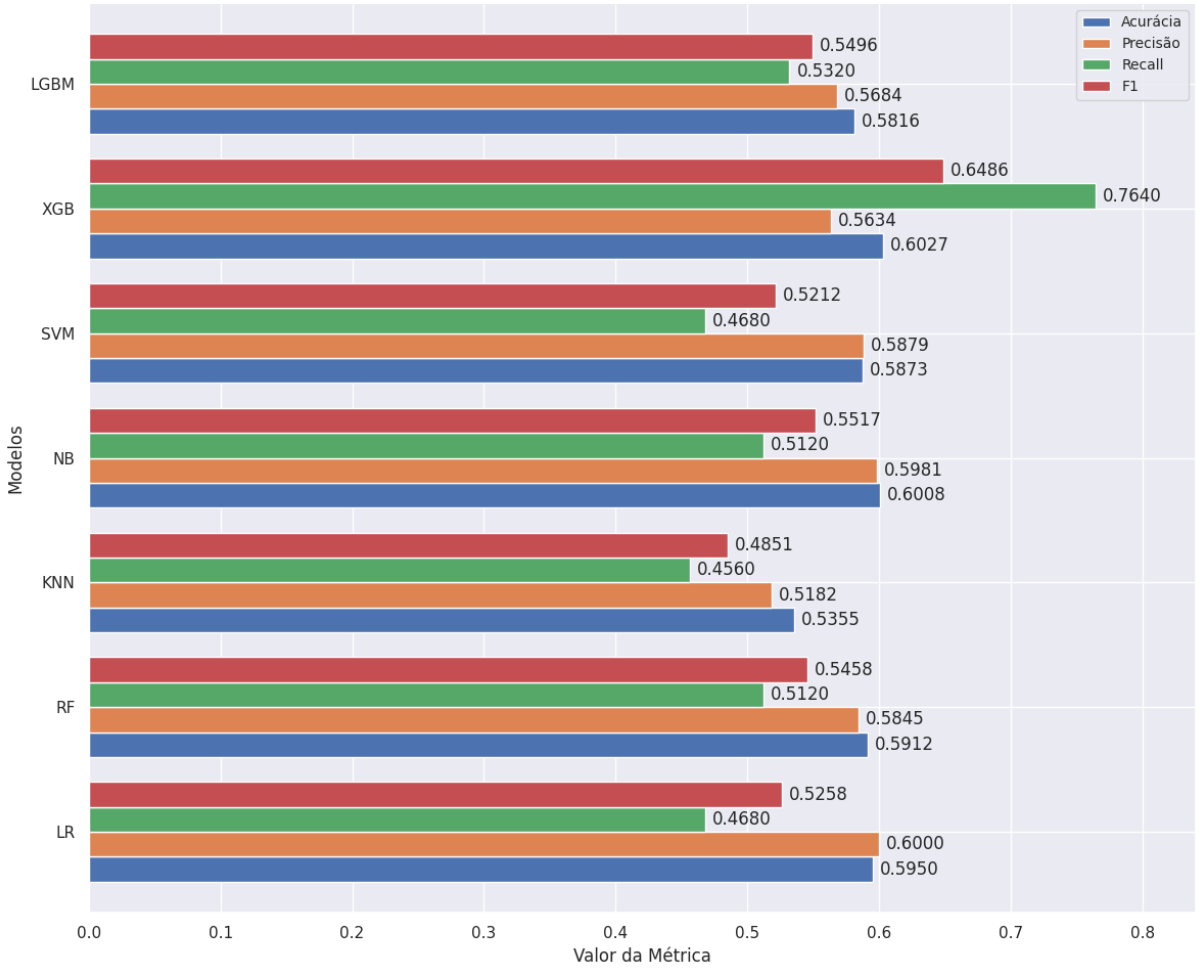

Figura 6 – Gráfico de barra das métricas dos modelos - cenário com empates.

modelo da Regressão Logística. Já para a Recall e F1, o modelo de destaque foi novamente o XGBoost, atingindo o valor de 76.40% para a primeira e 64.86% na segunda.

4.3 Discussão dos resultados

Considerando o cenário sem empates, exposto na Tabela 5, apesar de não ter havido uma disparidade grande entre os valores de cada modelo, há uma leve vantagem para o XGBoost, acompanhado, logo em seguida, do *Naive Bayes*, onde ambos obtiveram duas métricas de destaque cada um, com os maiores valores dentre as comparadas. O primeiro possui a maior porcentagem de acertos das previsões feitas e tem o modelo mais confiável nas predições positivas, já o segundo é o modelo mais completo para encontrar os exemplos positivos e o que tem o melhor equilíbrio entre classificar corretamente um caso positivo e encontrar os exemplos positivos. Mas como escrito anteriormente, não há uma grande diferença no desempenho dos modelos utilizando os dados de treino para que se possa apontar um grande destaque nessa fase.

Já no cenário com empates, exposto na Tabela 6, há uma mudança, já que a

disparidade entre o desempenho dos modelos aumenta. Temos nesse cenário o modelo advindo do algoritmo Regressão Logística como um modelo consistente em manter a maioria de suas métricas entre as melhores, com exceção da Recall, que está entre as piores, mostrando que ele não é um modelo tão bom quanto, por exemplo, o do LGBM em encontrar os exemplos positivos neste caso. Outro destaque neste cenário, mas por motivo diferente, fica por conta do modelo do XGBoost, que possui tanto a métrica de Recall quanto a F1 como as mais altas comparadas com as de outros métodos, só que o desvio padrão de ambas também é muito grande comparado às métricas dos demais, o que sugere que o modelo é menos consistente em seus resultados.

Considerando somente os dados de teste, as matrizes têm dados e insumos interessantes que podem ajudar no entendimento dos modelos e suas características. Uma análise possível das matrizes da Figura 3, representando o cenário sem empates, é de que há uma facilidade maior de todos os modelos em encontrar o verdadeiro positivo, que nesse caso seria o vencedor ser *casa* e o predito também ser *casa*. Esse comportamento pode ter acontecido pelo fato de que quando houve o corte das instâncias com empates do *dataset*, implicou em um desbalanceamento no conjunto de dados relacionado aos atributos alvo, significando que houve uma sobrerrepresentação da classe *casa* em desfavor da classe *fora*. Com o desbalanceamento, evidente na Figura 7, os modelos tendem a prever a classe majoritária e podem ter dificuldade de aprender as características da classe minoritária.

No cenário com empate, as matrizes de confusão expostas na Figura 5 apresentam um cenário diferente ao apresentado anteriormente, já que, a princípio, a distribuição do conjunto de classes está mais balanceado, como mostrado na Figura 8. Considerando o exemplo do modelo do XGBoost, onde há um número alto de verdadeiros positivos, 191, e também um número alto de verdadeiros negativos, 123, é possível imaginar que ele pode ser um ótimo modelo de predição, mas ao observar os casos de falsos positivos, com o valor de 148, indica que este modelo tem uma taxa de acerto alta na predição de exemplos positivos e exemplos negativos, mas ao custo de ter uma alta taxa de casos falsos positivos, ou seja, para poder ter um bom desempenho nos dois primeiros aspectos citados, o modelo tende a prever os casos como positivos (*casa* sendo o vencedor e o modelo predizendo *casa*), aumentando muito o número de previsões onde *fora* é o vencedor ou há o *empate*, mas o modelo prediz como *casa*. Analisando dessa forma é possível encontrar diferentes aspectos de cada modelo.

Com o auxílio das matrizes de confusão, é possível ter uma visão geral e parcial do desempenho dos modelos, mas são com as métricas restantes, mostradas na Figura 4 para o cenário sem empates e na Figura 6 para os cenários com empates, que são fornecidos os elementos necessários para poder avaliar e comparar os modelos.

- SVM: possui a maior F1 (0.8204), algo importante para garantir que as previsões

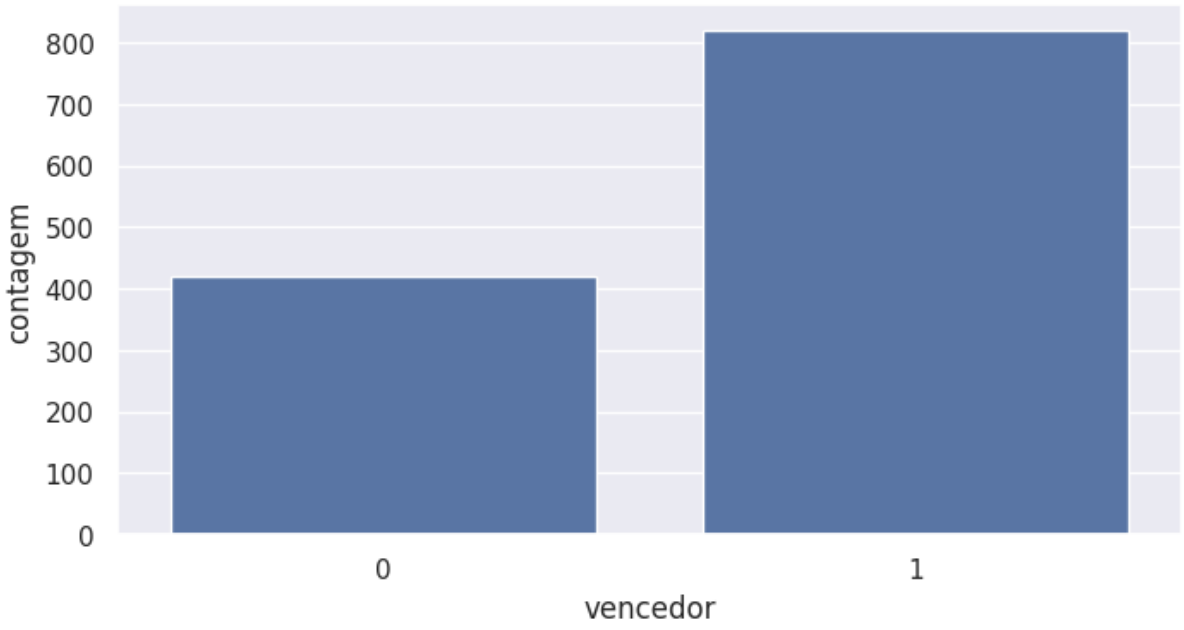

Figura 7 – Distribuição dos valores do atributo alvo - cenário sem empates

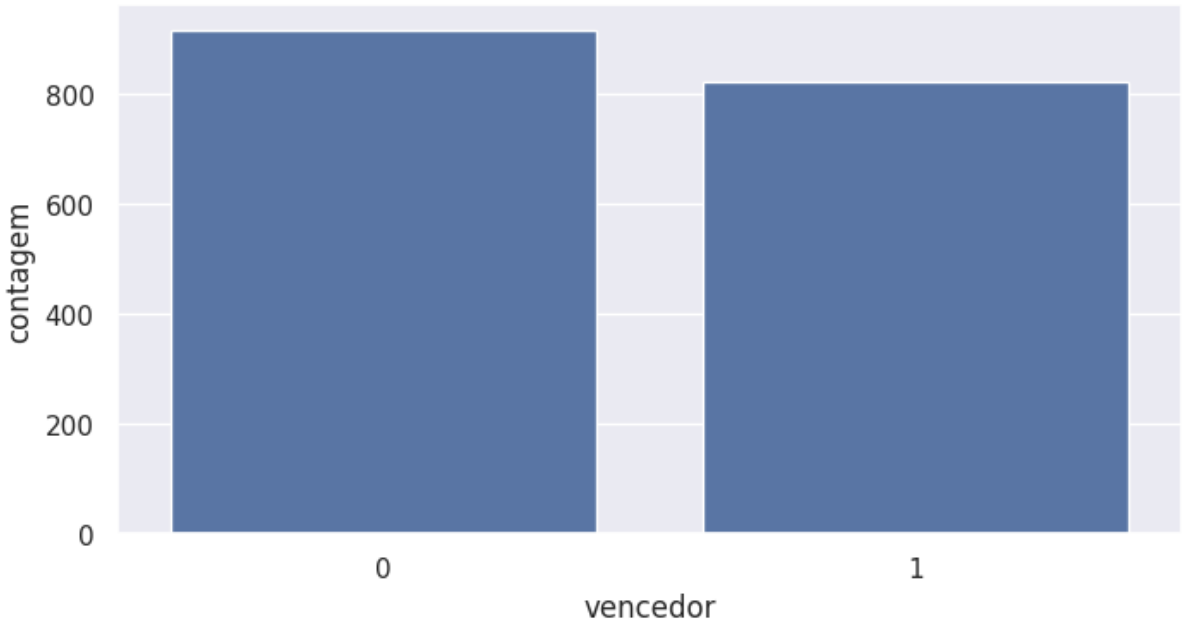

Figura 8 – Distribuição dos valores do atributo alvo - cenário com empates

sejam confiáveis e consistentes. Além de uma alta Precisão (0.7516) e Recall (0.9031), que mostram que o modelo conseguiu entender bem a separação entre as classes.

- *Naive Bayes*: apresenta um excelente desempenho em Recall (0.9069), sendo o mais alto entre os modelos, indicando que ele consegue indicar a maioria das vitórias da casa. Apesar da Precisão (0.7476) ser ligeiramente menor que a de alguns outros modelos, apresenta um ótimo equilíbrio de Precisão e Recall, com bom F1 (0.8196).
- Regressão Logística: foi considerado sua primeira posição na Acurácia (0.7292) para estar presente entre os destaques, indicando que o modelo fez a melhor previsão geral dos vencedores.

Os modelos considerados destaques no cenário com empates são:

- XGBoost: possuindo a primeira posição de 3 métricas, Acurácia(0.6027), F1(0.6486) e Recall(0.764), esse modelo se destaca principalmente no Recall, o que indica que ele é eficaz na identificação correta das partidas onde o time vencedor é o da casa. Por sua F1 também ser a mais alta, indica que há bom equilíbrio entre Precisão e Recall e sugere que há um bom índice de acertos para exemplos positivos (*casa*) e negativos (*fora* ou *empate*). Embora a Precisão não seja a mais alta, a combinação entre F1 e Acurácia projeta este modelo com um dos de melhores desempenho neste estudo.
- *Naive Bayes*: similar ao XGB, ele apresenta uma boa combinação entre Acurácia (0.6008) e F1 (0.5517), o que sugere sua eficácia em identificar o vencedor de maneira correta e também de garantir o equilíbrio entre Precisão e Recall, podendo fazer previsões mais equilibradas entre as classes.
- Regressão Logística: por possuir uma Acurácia (0.5950) parecida aos dos modelos citados anteriormente e um F1 (0.5258) razoável, seu desempenho pode ser indicado como razoável na classificação de partidas. Por outro lado, sua Precisão(0.6) é a mais alta da tabela, mostrando que em boa parte das predições de vitória do time da casa, ele está correto.

O destaque negativo para ambos os cenários citados acima é do modelo do KNN, que apresentou as menores pontuações em todas as métricas, sugerindo que ele apresenta dificuldade em generalizar bem os padrões dos dados.

Uma última observação que pode ser feita é que, ao remover os empates para o cenário sem empates, o conjunto de dados se tornou mais simples para os modelos, facilitando a tarefa de classificação para todos eles, o que foi refletido no aumento dos valores das métricas de todos os modelos deste cenário.

5 Conclusão

O futebol, um dos esportes mais populares do mundo, juntamente com as crescentes e ricas bases de dados relacionadas ao esporte, podem produzir insumos e informações valiosas, em diversas áreas, quando essas bases forem analisadas com técnicas de aprendizado de máquina e ciência de dados. O presente estudo utilizou essas técnicas para comparar algoritmos de aprendizado de máquina aplicados em um base construída para esta finalidade, com dados do Campeonato Brasileiro de Futebol Série A.

Os testes foram realizados utilizando duas abordagens, conjunto de dados sem empates, e conjunto de dados contendo empates. Comparado à primeira abordagem, os valores das métricas resultantes dos modelos da segunda abordagem foram significantemente maiores devido ao fato da abordagem sem empates simplificar bastante o problema de predição para os modelos preverem.

O problema consistia em prever o atributo vencedor, que era necessariamente binário e assumia os valores 0 ou 1. No cenário sem empate, o vencedor sendo casa foi representado por 1, e caso o vencedor fosse fora ou acontecesse o empate, seriam representados por 0. Os algoritmos com melhor desempenho foram o XGBoost e o *Naive Bayes*, ambos aparecendo com bons valores de métricas e boas combinações das mesmas nos dois cenários, mostrando que se adequaram bem ao problema de prever o resultado das partidas.

Os resultados do trabalho indicam que: i) mudanças que alteram o sentido do atributo alvo são impactantes para as classificações feitas pelos modelos e ii) alguns modelos criados a partir dos algoritmos escolhidos podem ser bem sucedidos nas predições das partidas do Campeonato Brasileiro de Futebol e outros não são adequados para este tipo de problema.

Para trabalhos futuros, sugere-se a utilização da técnica de ajuste de hiperparâmetros para que se possam encontrar possíveis melhores versões dos modelos criados, de um conjunto de dados contendo uma classe alvo multinomial e de algoritmos classificadores capazes de predizer esse tipo de atributo, fazendo com que o problema se aproxime mais da realidade, onde pode haver vencedor de casa, de fora e empate, todos representando suas próprias classes.

Referências

ABDULKAREEM, N. M.; ABDULAZEEZ, A. M. Machine learning classification based on radom forest algorithm: A review. **International journal of science and business**, IJSAB International, v. 5, n. 2, p. 128–142, 2021. Citado na página 16.

ARTUSO, A. R. Distribuição gaussiana dos resultados do campeonato brasileiro de futebol: um modelo para estimar classificações em campeonatos de modalidades coletivas. **Revista Brasileira de Ciências do Esporte**, v. 30, n. 1, 2008. Citado na página 10.

BABOOTA, R.; KAUR, H. Predictive analysis and modelling football results using machine learning approach for english premier league. **International Journal of Forecasting**, Elsevier, v. 35, n. 2, p. 741–755, 2019. Citado na página 16.

BASHA, S. M.; RAJPUT, D. S.; VANDHAN, V. Impact of gradient ascent and boosting algorithm in classification. **International Journal of Intelligent Engineering & Systems**, v. 11, n. 1, 2018. Citado na página 18.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining**. [S.l.: s.n.], 2016. p. 785–794. Citado na página 17.

CONCEIÇÃO, M. P. d. **Avaliação dos modelos de machine learning: verificando os resultados no futebol**. Dissertação (B.S. thesis) — Universidade Federal Fluminense, 2022. Citado na página 21.

DEUS, G. A. d. **Utilização de aprendizado de máquina para previsão de resultados de jogos de futebol**. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2019. Citado na página 11.

DEUS, G. A. d. **Utilização de aprendizado de máquina para previsão de resultados de jogos de futebol**. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2019. Citado na página 20.

FACELI, K.; LORENA, A. C.; GAMA, J.; ALMEIDA, T. A. d.; CARVALHO, A. C. P. d. L. F. d. Inteligência artificial: uma abordagem de aprendizado de máquina. 2021. Citado 2 vezes nas páginas 13 e 14.

FÁVERO, L. P.; BELFIORE, P. **Manual de análise de dados: estatística e modelagem multivariada com Excel®, SPSS® e Stata®**. [S.l.]: Elsevier Brasil, 2017. Citado na página 16.

FIFA. **FIFA Big Count 2006: 270 million people active in football**. 2007. <https://digitalhub.fifa.com/m/55621f9fdc8ea7b4/original/mzid0qmguixkcmruvema-pdf.pdf>. [Online; accessed 09-Dezembro-2023]. Citado na página 10.

GE. **Brasileirão supera Premier League e é eleito a liga mais forte do mundo pela segunda vez seguida**. 2023. <https://ge.globo.com/futebol/futebol-internacional/noticia/2023/01/23/brasileirao-supera-premier-league-e-e-eleito-a-liga-mais-forte-do-mundo-pela-segunda-vez-seguida.ghtml>. [Online; accessed 30-Outubro-2023]. Citado na página 13.

GONZALEZ, L. d. A. Regressão logística e suas aplicações. Universidade Federal do Maranhão, 2018. Citado na página 16.

HILBE, J. M. Logistic regression. **International encyclopedia of statistical science**, v. 1, p. 15–32, 2011. Citado na página 16.

HUCALJUK, J.; RAKIPOVIć, A. Predicting football scores using machine learning techniques. In: **2011 Proceedings of the 34th International Convention MIPRO**. [S.l.: s.n.], 2011. p. 1623–1627. Citado na página 10.

IBGE. **Falta de tempo e de interesse são os principais motivos para não se praticar esportes no Brasil**. 2017. <https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013-agencia-de-noticias/releases/15128-falta-de-tempo-e-de-interesse-sao-os-principais-motivos-para-nao-se-praticar-esportes-no-bras [Online; accessed 09-Dezembro-2023]. Citado na página 10.

LOPES, G. R.; ALMEIDA, A. W. S.; DELBEM, A.; TOLEDO, C. F. M. Introdução à análise exploratória de dados com python. **Minicursos ERCAS ENUCMPI**, v. 2019, p. 160–176, 2019. Citado na página 27.

MAGLOGIANNIS, I. G. **Emerging artificial intelligence applications in computer engineering: real word ai systems with applications in ehealth, hci, information retrieval and pervasive technologies**. [S.l.]: Ios Press, 2007. v. 160. Citado 2 vezes nas páginas 17 e 18.

MARCHI, A. J.; FONSECA, M. Z.; BODÊ, J. Machine learning: aplicabilidade em monitoramento de redes. In: **FatecSeg-Congresso de Segurança da Informação**. [S.l.: s.n.], 2023. Citado na página 19.

MATOS, P. F.; LOMBARDI, L. d. O.; CIFERRI, R. R.; PARDO, T. A.; CIFERRI, C. D.; VIEIRA, M. T. Relatório técnico "métricas de avaliaçao". **Universidade Federal de Sao Carlos**, 2009. Citado na página 18.

RAY, S. A quick review of machine learning algorithms. In: IEEE. **2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)**. [S.l.], 2019. p. 35–39. Citado 2 vezes nas páginas 17 e 18.

RODRIGUES, F.; PINTO, Â. Prediction of football match results with machine learning. **Procedia Computer Science**, Elsevier, v. 204, p. 463–470, 2022. Citado na página 21.

RUFO, D. D.; DEBELEE, T. G.; IBENTHAL, A.; NEGERA, W. G. Diagnosis of diabetes mellitus using gradient boosting machine (lightgbm). **Diagnostics**, MDPI, v. 11, n. 9, p. 1714, 2021. Citado na página 18.

SILVA, D. F. B. F. d. **Pré-processamento de Dados e Comparação entre Algoritmos de Machine Learning para a Análise Preditiva de Falhas em Linhas de Produção para o Controlo**. Tese (Doutorado) — Instituto Superior de Engenharia do Porto, 2021. Citado na página 14.

SIMON, V. A. C. **Brasil lidera crescimento de visitas a sites de apostas esportivas**. 2023. <https://www.similarweb.com/blog/pt/insights/brasil-lidera-crescimento-de-visitas-a-sites-de-apostas-esportivas/>. [Online; accessed 09-Dezembro-2023]. Citado na página 10.

SIVAKUMAR, A.; GUNASUNDARI, R. A survey on data preprocessing techniques for bioinformatics and web usage mining. **International Journal of Pure and Applied Mathematics**, v. 117, n. 20, p. 785–794, 2017. Citado 2 vezes nas páginas 14 e 15.

SOUZA[1], W. F. de. A geografia do futebol, o campeonato brasileiro de pontos corridos e o modelo mais adequado ao brasil. 2015. Citado na página 13.

WEERADDANA, N.; PREMARATNE, S. Unique approach for cricket match outcome prediction using xgboost algorithms. **Journal of Theoretical and Applied Information Technology**, v. 99, n. 9, p. 2162–2173, 2021. Citado na página 17.