CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Análise de concordância sobre previsões de resultado de partidas de futebol

Juan Lima de Albuquerque

João Pessoa, PB

2023

Juan Lima de Albuquerque

Análise de concordância sobre previsões de resultado de partidas de futebol

Monografia apresentada ao curso Engenharia de Computação
do Centro de Informática, da Universidade Federal da Paraíba,
como requisito para a obtenção do grau de Bacharel em Análise de concordância sobre previsões de resultado de partidas de futebol

Orientadora: Thaís Gaudencio do Rêgo

Junho de 2023

Catalogação na publicação
Seção de Catalogação e Classificação

A345a Albuquerque, Juan Lima de.
Análise de concordância sobre previsões de resultado de partidas de futebol / Juan Lima de Albuquerque. - João Pessoa, 2023.
46 f. : il.

Orientação: Thaís Gaudencio do Rêgo.
TCC (Graduação) - UFPB/CI.

1. Aprendizagem de máquina. 2. Futebol. 3. Previsão de partidas. 4. Análise de concordância. 5. Agrupamento de modelos. I. Rêgo, Thaís Gaudencio do. II. Título.

UFPB/CI CDU 004.832

Elaborado por Michelle de Kássia Fonseca Barbosa - CRB-738

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Engenharia de Computaçãointitulado ***Análise de concordância sobre previsões de resultado de partidas de futebol*** de autoria de Juan Lima de Albuquerque, aprovada pela banca examinadora constituída pelos seguintes professores:

Thaís Gaudencio do Rêgo

Prof. Dr. Thaís Gaudencio do Rêgo
Universidade Federal da Paraíba

Telmo de Menezes e Silva Filho

Prof. Dr. Telmo De Menezes E Silva Filho
Universidade Federal da Paraíba

Yuri de Almeida Malheiros Barbosa

Prof. Dr. Yuri de Almeida Malheiros Barbosa
Universidade Federal da Paraíba

João Pessoa, 27 de junho de 2023

Centro de Informática, Universidade Federal da Paraíba
Rua dos Escoteiros, Mangabeira VII, João Pessoa, Paraíba, Brasil CEP: 58058-600
Fone: +55 (83) 3216 7093 / Fax: +55 (83) 3216 7117

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a Deus pela oportunidade de realizar um sonho meu e da minha família: a conclusão de um curso superior em engenharia. Também gostaria de expressar minha gratidão aos meus pais, Regina e Rubenilson, que, mesmo sendo jovens, aprenderam com a vida a criar um filho, longe de casa, para proporcionar uma educação de melhor qualidade. Eles sempre me apoiaram e se esforçaram ao máximo para me fornecer toda a estrutura necessária.

Além disso, gostaria de agradecer a uma pessoa muito especial que encontrei durante minha jornada universitária: minha melhor amiga, namorada, noiva e futura esposa, Suanny. Agradeço por todos os momentos, disciplinas, risadas e tristezas que compartilhamos ao longo desse percurso. Obrigado por me apoiar e incentivar durante toda essa jornada, te amo.

Também quero expressar minha gratidão aos amigos para toda a vida que fiz durante o curso: Adriano, Alexandre, Arthur, Ícaro, José Eugênio e José Olívio. Nossa companheirismo e união tornaram tudo mais fácil e leve nessa caminhada desafiadora que foi o nosso curso.

E não posso deixar de agradecer aos professores Telmo, Thaís e Yuri pelo tempo, disponibilidade e atenção que dedicaram durante o desenvolvimento do projeto, mesmo com todos as dificuldades, vocês sempre se mostraram compreensíveis e disponíveis para ajudar. Em especial, gostaria de agradecer à professora Thaís por cuidar de todos nós do curso de Engenharia de Computação. Você é uma referência para todos nós, e nunca desistiu desses alunos que lhe deram tanto trabalho. Obrigado por tudo.

RESUMO

Vivemos em uma época em que se movimenta um alto volume de dinheiro no mercado de apostas esportivas, e com isso o interesse recente na utilização de técnicas de aprendizagem de máquina para predição de resultados esportivos tem crescido. Este trabalho, através da utilização de uma base de jogos da *Premier League*, avaliou o uso de um conjunto de algoritmos de aprendizagem de máquina para prever resultados esportivos e avaliar a eficácia de diferentes níveis da concordância das previsões dos diferente modelos usados. O resultado das previsões após análise de concordância entre as previsões dos modelos e a dificuldade em prever empates foram resultados destacados no trabalho. Como resultado mais significativo, tivemos uma taxa de acurácia de 83% após exclusão dos jogos que não tiveram 100% de concordância nas respostas entre os modelos treinados. Além disso, foram sugeridos trabalhos futuros, como a inclusão de outras variáveis e características dos jogos, a generalização dos testes para outras ligas e uma nova abordagem para o problema de previsões em casos de empate.

Palavras-chave: <Aprendizagem de máquina>, <Futebol>, <Previsão de partidas>, <Análise de Concordância>, <Agrupamento de modelos>.

ABSTRACT

We live in a time of high turnover of money in the sports betting market, and with this the recent interest in using machine learning techniques for predicting sports outcomes has grown. This paper, using a database of Premier League games, evaluated the use of a set of machine learning algorithms to predict sports results and assess the effectiveness of different levels of agreement of the predictions of the different models used. The results of the predictions after analyzing the agreement between the predictions of the models and the difficulty in predicting draws were highlighted in the paper. As a more significant result, we had an accuracy rate of 83% after excluding the games that did not have 100% agreement in the answers between the trained models. Besides this, future works were suggested, such as the inclusion of other variables and characteristics of the games, the generalization of the tests to other leagues, and a new approach to the problem of predictions in tie cases.

Translated with www.DeepL.com/Translator (free version)

Key-words: <Machine learning>, <Soccer>, <Match prediction>, <Agreement Analysis>, <Model joining>.

LISTA DE FIGURAS

1 Tipos da aprendizagem supervisionada. 18

2 Representação gráfica do hiperplano, margem e vetores de suporte. 19

3 Perceptron Multicamadas. 21

4 Funcionamento do *Random Forest* 22

5 Modelo de matriz de confusão. 23

6 Matriz de correlação. 33

7 Diagrama de fluxo do processo de previsão. 37

8 Variação conforme mudança do *threshold*. 39

9 Matrizes de confusão, com valores normalizados, geradas a partir da variação do *threshold*. 42

10 Matriz de confusão do *XGBoost* com valores normalizados. 43

LISTA DE TABELAS

1 Resumo dos trabalhos relacionados. 28

2 Atributo selecionados da planilha fornecida pelo *Football Data* para compor o dataset do projeto 30

3 Novos atributos criados através de informações retiradas do site da *Premier League* para compor o dataset do projeto. 32

4 Os modelos de aprendizagem de máquina utilizados no trabalho e seus respectivos parâmetros variados. 35

5 Demonstrativo de número de partidas e métricas com a variação do *threshold*. 40

6 Demonstrativo de acurácia e quantidade de partidas por classe com a variação do *threshold*. 41

LISTA DE ABREVIATURAS

IA - Inteligência Artificial

ML - *Machine Learning* (do português, Aprendizagem de Máquina)

MLP - *Multi-layer Perceptron* (do português, Perceptron de Multicamadas)

RF – *Random Forest* (do português, Floresta Aleatória)

SVC - *C-Support Vector Classification* (do português, Classificação de Vetor de Suporte C)

SVM - *Support Vector Machine* (do português, Máquina de Vetor de Suporte)

Sumário

1 INTRODUÇÃO **15**

1.1 Contextualização e Motivações 15

1.1.1 Objetivo geral 16

1.1.2 Objetivos específicos 16

1.2 Estrutura da monografia 16

2 FUNDAMENTAÇÃO TEÓRICA **18**

2.0.1 Classificação 18

2.1 Modelos Utilizados 19

2.1.1 Máquina de Vetores de Suporte 19

2.1.2 Aumento de Gradiente Extremo (do inglês, *Extreme Gradient Boosting* - XGBoost) 20

2.1.3 Perceptron de Multicamadas 20

2.1.4 Floresta Aleatória 21

2.1.5 *Naive Bayes* 22

2.2 Métricas 22

2.3 *Premier League* 24

3 Trabalhos relacionados **26**

4 METODOLOGIA **29**

4.1 Ferramentas 29

4.2 Base de dados 29

4.3 Pré-processamento 31

4.4 Treinamento 34

4.5 Teste 35

5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS **38**

5.1 Comparação entre diferentes níveis de *threshold* 38

5.2 Análise da classe de empate 40

5.3 Comparação entre resultados com variações de *threshold* e melhor modelo individual 42

6 CONCLUSÕES E TRABALHOS FUTUROS **44**

6.1 Trabalhos futuros 44

REFERÊNCIAS **45**

1 INTRODUÇÃO

O estudo da utilização de técnicas de aprendizagem de máquina nas mais diversas áreas é um tema em ascensão nos tempos atuais. A junção da inteligência artificial com os mais diversos temas tem proporcionado avanços significativos em diferentes campos de aplicação. Neste contexto, a capacidade de utilizar algoritmos e modelos preditivos para realizar previsões e tomadas de decisão tem despertado um grande interesse tanto na comunidade acadêmica, quanto no setor empresarial.

O objetivo deste trabalho é contribuir para a compreensão e aprimoramento das técnicas de previsão de resultados em apostas esportivas, explorando diferentes abordagens e identificando estratégias eficazes para aumentar a precisão e segurança das previsões. A análise da segurança das previsões de estatísticas esportivas envolve avaliar a precisão e a confiabilidade das previsões feitas por diferentes algoritmos de aprendizagem de máquina.

É importante destacar que a previsão de resultados esportivos é uma tarefa complexa e desafiadora. Neste trabalho, é proposta uma ferramenta para auxiliar nas análises, onde a interferência humana continua sendo fundamental. As previsões da IA podem ser afetadas por fatores imprevisíveis, como lesões de jogadores, expulsões ou mudanças de tempo, e podem não ser precisas em todas as situações. Portanto, a análise de segurança é uma parte importante do processo de previsão de resultados esportivos, pois ajuda a garantir que as previsões sejam tão precisas e confiáveis quanto possível.

Além disso, é importante lembrar que apostas esportivas envolvem questões éticas e legais. Portanto, é importante seguir as leis e regulamentações locais e agir de forma responsável e consciente quando se trata do assunto.

1.1 Contextualização e Motivações

O futebol, considerado o esporte mais popular no Brasil, desperta uma paixão intensa entre os seus admiradores e, nos últimos anos, tem se conectado cada vez mais com outro hábito dos brasileiros: as apostas esportivas. Iniciadas nas décadas de 1930, as apostas conquistaram rapidamente o gosto da população, mas foram proibidas na década de 1940. Somente na década de 1960, foram legalizadas, porém, restritas às loterias federais. Em 2018, as apostas esportivas foram liberadas, de forma ampla, no país [4].

A previsão de resultados esportivos é uma área de grande interesse, pois antecipar o desfecho de um jogo pode proporcionar vantagens estratégicas e financeiras significativas. Diante disso, o uso de técnicas avançadas de análise de dados e aprendizado de máquina desperta o interesse, tanto de pesquisadores, quanto de profissionais da área.

Estima-se que os apostadores brasileiros movimentam anualmente cerca de R$ 12

bilhões em apostas esportivas, segundo Magno José, presidente do Instituto Brasileiro Jogo Legal [5]. No entanto, essa prática pode acarretar prejuízos financeiros significativos para os apostadores, principalmente quando realizada impulsivamente, sem embasamento ou conhecimento adequado. Um levantamento divulgado pela revista *Economist* revelou que os apostadores brasileiros perderam US$ 4,1 bilhões em 2014 em sites de loteria e apostas esportivas [6]. É importante destacar que, na época, as apostas esportivas ainda não eram legalizadas, e desde então, a atividade tem apresentado um crescimento considerável.

Diante desse contexto, este trabalho tem como objetivo principal oferecer previsões de resultados de partidas de futebol com um nível de confiança pré-determinado. Para isso, serão utilizados dados detalhados de temporadas anteriores da *Premier League* e diferentes modelos de classificação.

1.1.1 Objetivo geral

O objetivo geral deste trabalho é utilizar técnicas de aprendizagem de máquina para prever estatísticas esportivas e avaliar a eficácia de diferentes níveis de concordâncias no agrupamento de diferente modelos usados, além de buscar formas de aumentar a precisão das previsões e identificar as incertezas.

1.1.2 Objetivos específicos

Este tópico apresenta os objetivos específicos do presente trabalho, que visa explorar o uso de técnicas de agrupamento de modelos de Inteligência Artificial para prever o resultado de jogos de futebol, utilizando filtros de concordância entre os modelos do conjunto. Os objetivos específicos são:

- Construir um conjunto de modelos, selecionando diferentes algoritmos de IA adequados para a tarefa de previsão de resultados de jogos de futebol;
- Treinar e ajustar os modelos individualmente, utilizando conjuntos de dados de partidas de futebol;
- Aplicar uma estratégia de classificação com abstenção, para descarte de previsões incertas, baseadas em filtros de concordância (*threshold*) das previsões feitas pelo conjunto de modelos construído;
- Avaliar a eficácia do conjunto de modelos agrupados.

1.2 Estrutura da monografia

Os tópicos abaixo descrevem a estrutura principal da monografia:

- **Capítulo 2**: são discutidas as bases teóricas e os conceitos essenciais para compreender o trabalho proposto. São abordados os fundamentos de Aprendizado de Máquina (do inglês,*Machine Learning* - ML), com ênfase no conceito de classificação, e são listados e descritos todos os modelos de aprendizado utilizados nesta metodologia. Além disso, é apresentada uma descrição da *Premier League*, o campeonato inglês de futebol do qual foram extraídas as partidas utilizadas na solução proposta.

- **Capítulo 3**: são apresentados os estudos relacionados ao projeto, incluindo suas definições, características, métricas e resultados. Também é fornecida uma tabela comparativa entre esses trabalhos, que inclui a pesquisa em questão.

- **Capítulo 4**: são abordadas as etapas para chegar à solução proposta. São demonstrados o processo de formação da base de dados, as informações sobre sua obtenção, as técnicas de pré-processamento aplicadas, a separação do conjunto de dados e os detalhes sobre o treinamento e teste.

- **Capítulo 5**: são apresentados os resultados obtidos com os métodos utilizados. São exibidos os testes realizados com todos os modelos e a combinação de seus resultados para obter uma previsão final. Além disso, é realizada uma análise e discussão abrangente desses resultados.

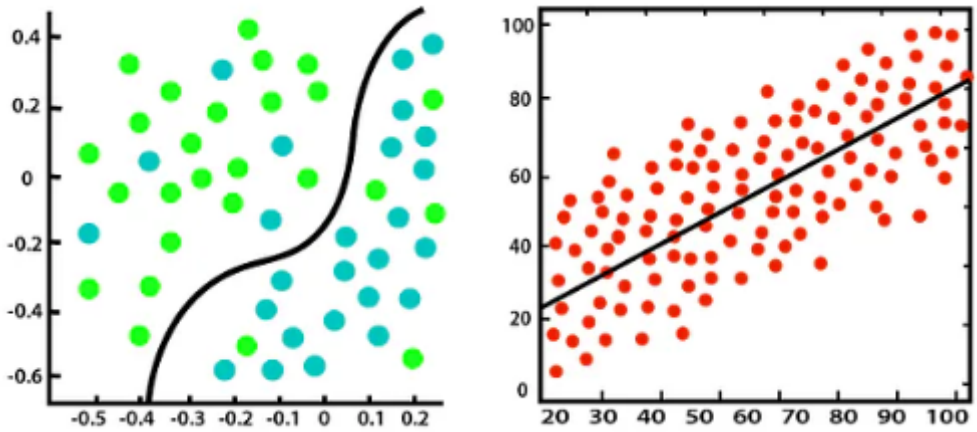
- **Capítulo 6**: são apresentadas as conclusões a partir da solução proposta e dos resultados obtidos, levando em consideração o problema abordado. São destacados os pontos fortes e fracos identificados, e o capítulo é finalizado com uma discussão sobre trabalhos futuros que podem ser realizados com base neste estudo.

2 FUNDAMENTAÇÃO TEÓRICA

Este capitulo apresenta conceitos teóricos abordados no decorrer deste trabalho. O capítulo inicia apresentando a técnica de classificação. Em seguida, são apresentados os modelos utilizados neste trabalho, abordando características de cada um. Também são descritas as métricas que serão adotadas para avaliar as saídas de cada modelo utilizado. Por fim, são apresentados alguns trabalhos relacionados.

2.0.1 Classificação

Classificação é uma tarefa da aprendizagem supervisionada que prediz valores discretos específicos aos quais a entrada pertence. Esses valores podem ser denominados classes ou categorias. Não deve ser confundido com Regressão, no qual é usado para prever valores contínuos, em vez de valores categóricos.



(a) Classificação **(b) Regressão**

Figura 1: Tipos da aprendizagem supervisionada.
Fonte: [12].

Na Figura 1, podemos observar a diferença entre eles. Na tarefa de classificação, 1 (a), busca-se encontrar o melhor limite de decisão para segregar as amostras em categorias discretas. Na regressão, os dados compreendem valores contínuos de rótulos. A saída de um modelo de regressão é uma variável contínua, e não uma categoria, como na classificação [12]. Para regressão, a ideia da modelagem é encontrar a melhor linha de ajuste que possa prever com precisão a saída, como pode-se observar na reta do gráfico da Figura 1 (b).

Em problemas de classificação, existem diversos algoritmos que podem ser utilizados para atuar nos mais variados tópicos.

2.1 Modelos Utilizados

Dentre os diversos algoritmos de aprendizagem de máquina que podem ser utilizados na resolução de problemas, cinco deles são utilizados neste projeto a fim de prever resultados de partidas de futebol. Nos tópicos a seguir, serão listados estes modelos e suas abordagens.

2.1.1 Máquina de Vetores de Suporte

As Máquinas de Vetores de Suporte [22] é uma técnica de aprendizagem de máquina utilizado tanto em problemas de classificação, como de regressão.

É um algoritmo que busca encontrar um hiperplano que separe eficientemente os pontos de dados de diferentes classes. Esse hiperplano é escolhido de forma a maximizar a margem entre as classes, proporcionando uma separação mais clara. Na Figura 2, é possível observar a representação visual dessa ideia, com os pontos mais próximos do hiperplano sendo os vetores de suporte, marcados como mais e menos. A margem é definida como a distância mínima de um exemplo para uma superfície de decisão.

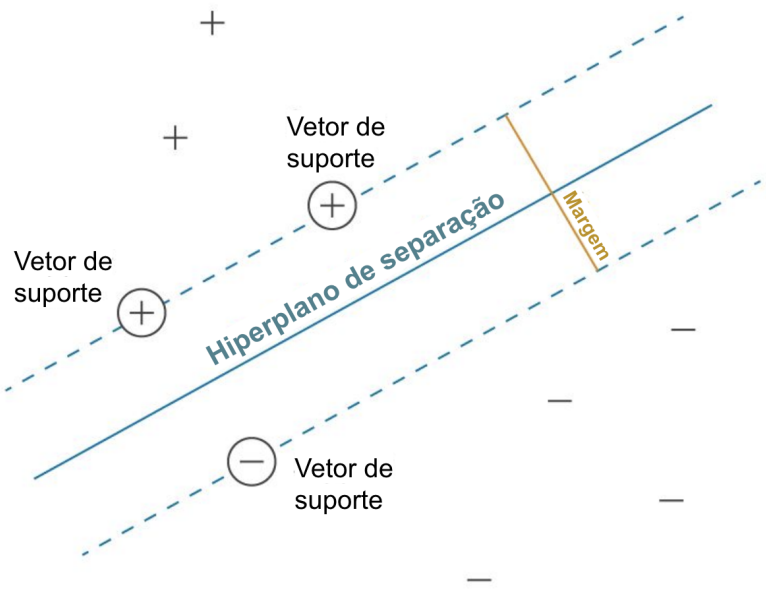


Figura 2: Representação gráfica do hiperplano, margem e vetores de suporte.
Fonte: Adaptado de [23].

O SVM é capaz de lidar com conjuntos de dados não linearmente separáveis com o uso de funções de *kernel*. Estas funções mapeiam os dados para um espaço dimensional diferente, geralmente superior, com a expectativa de que as classes sejam mais fáceis de separar após essa transformação, simplificando potencialmente os limites de decisão não

lineares complexos para lineares no espaço de recurso mapeado de dimensão superior. Os *kernels* tornam os SVMs mais flexíveis e capazes de lidar com problemas não lineares [23].

Ele é destinado a problemas de classificação binária, e os problemas de classificação multi classe geralmente são abordados por meio de uma abordagem de redução a problemas binários individuais. Existem várias técnicas disponíveis para generalizar o SVM e lidar com problemas multi classe. Uma abordagem comum é decompor o problema multi classe em vários subproblemas binários ou reformular o algoritmo de treinamento do SVM para suportar diretamente classificações multi classe. [24].

O SVC é o modelo utilizado neste projeto e é uma implementação específica do SVM para problemas de classificação.

2.1.2 Aumento de Gradiente Extremo (do inglês, *Extreme Gradient Boosting* - XGBoost)

O XGBoost [25] é uma implementação de código aberto popular. É um conjunto de árvore de decisão baseado em *Gradient Boosting* projetado para ser altamente escalável. O *Gradient Boosting* é um algoritmo de aprendizagem supervisionada que busca prever com precisão uma variável alvo combinando um conjunto de estimativas de modelos mais simples e fracos. Ele se destaca em competições de aprendizagem de máquina devido à sua capacidade robusta de lidar com diferentes tipos de dados, relações e distribuições, além de oferecer uma variedade de hiperparâmetros ajustáveis para melhorar o desempenho do modelo [26].

2.1.3 Perceptron de Multicamadas

O MLP é uma arquitetura de rede neural artificial que consiste em camadas de neurônios interconectados. Ele é a forma mais comum de redes neurais e é amplamente utilizado em problemas de classificação e regressão.

Em sua forma mais fundamental, esse modelo é composto por um número limitado de camadas sequenciais. Cada camada é constituída por um número finito de unidades, frequentemente denominadas neurônios. Cada unidade de uma camada pode estar conectada a todas as unidades da camada subsequente. Essas conexões são geralmente referidas como links ou sinapses [27].

A informação flui de uma camada para a camada seguinte. A primeira camada, conhecida como camada de entrada, é composta pelos dados de entrada. Em seguida, existem camadas intermediárias chamadas de camadas ocultas. A saída final é obtida na última camada, que é naturalmente chamada de camada de saída. Na Figura 3, observa-se a arquitetura padrão de uma MLP:

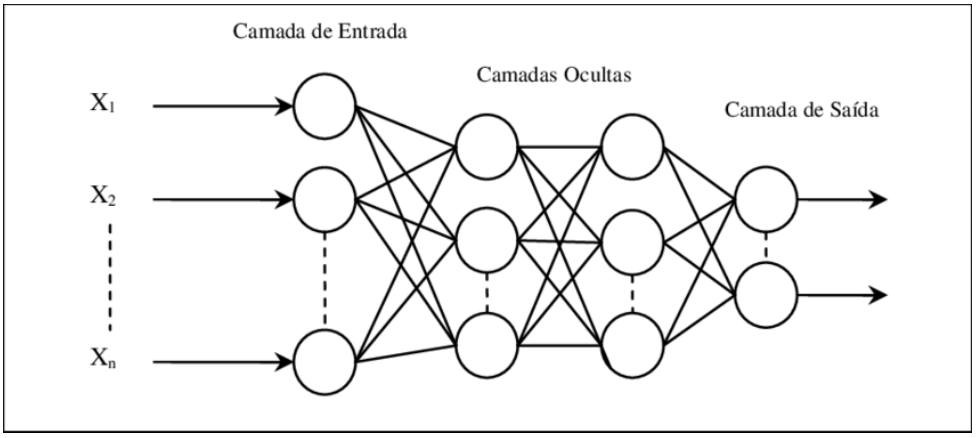


Figura 3: Perceptron Multicamadas.
Fonte: [29].

O procedimento de aprendizagem do MLP pode ser definido abaixo [28]:

1. Começando pela camada de entrada, os dados são transmitidos através da rede neural, camada por camada, até a saída. Essa etapa é chamada de propagação direta.

2. Com base na saída, é calculado o erro (a diferença entre o resultado previsto e o conhecido). O objetivo é minimizar esse erro.

3. É feita a retropropagação do erro sobre as camadas e o modelo é atualizado.

Os três passos mencionados acima são repetidos ao longo de várias épocas para aprender os pesos ideais. Por fim, a saída é obtida através de uma função de limiar para obter os rótulos de classe previstos.

2.1.4 Floresta Aleatória

O *Random Forest* [30] é um algoritmo de aprendizagem de máquina comumente usado que combina a saída de várias árvores de decisão para alcançar um único resultado. A facilidade de uso e flexibilidade do sistema impulsionam sua ampla adoção, tornando-o capaz de lidar, tanto com problemas de classificação, quanto com problemas de regressão.

Ao utilizar o algoritmo de *Random Forest*, que combina várias árvores de decisão em um conjunto, como apresentado na Figura 4, é possível obter previsões mais precisas, especialmente quando as árvores individuais não estão correlacionadas entre si [31]. Ele recebe esse nome porque cada árvore de decisão na floresta é treinada com uma amostra aleatória dos dados de treinamento, permitindo que aprendam padrões distintos.

Ele utiliza duas técnicas principais: *bagging* e aleatorização de recursos [31]. A técnica de *bagging* consiste em criar múltiplas árvores de decisão utilizando conjuntos

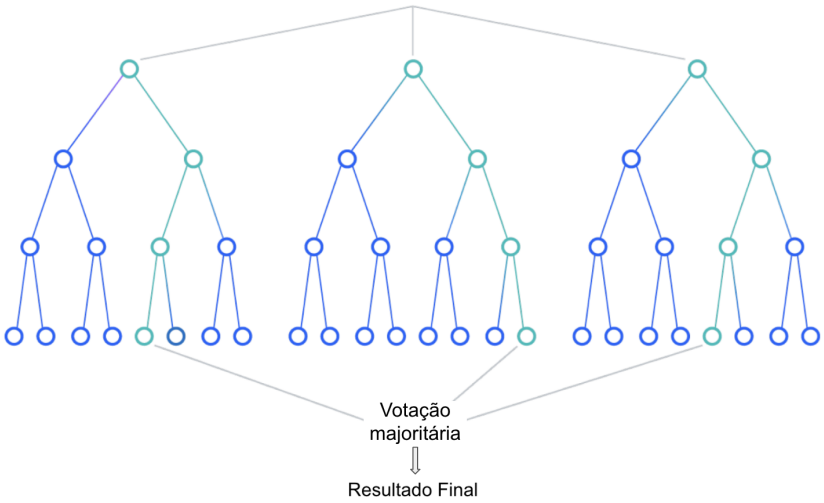


Figura 4: Funcionamento do *Random Forest*
Fonte: Adaptado de [31].

diferentes de dados de treinamento. A aleatorização de recursos é outra técnica importante empregada no modelo. Ao invés de utilizar todos os recursos disponíveis para treinar cada árvore de decisão, o algoritmo seleciona aleatoriamente um subconjunto desses recursos. Isso implica que cada árvore de decisão é exposta a um conjunto distinto de atributos, evitando que sejam excessivamente similares ou correlacionadas entre si.

2.1.5 *Naive Bayes*

O *Naive Bayes* [32] é um algoritmo de aprendizagem simples que se baseia no Teorema de Bayes e faz uma suposição de que os atributos são independentes entre si, dada a classe. Embora essa suposição de independência nem sempre seja verdadeira na prática, o *Naive Bayes* ainda é capaz de fornecer resultados de classificação competitivos. Além disso, ele possui vantagens como eficiência computacional e outras características desejáveis, o que faz com que seja amplamente utilizado na prática [33].

Ele fornece um mecanismo para utilizar as informações presentes nos dados de amostra para estimar a probabilidade posterior $P(y \mid x)$ de cada classe $\mathbf{y}$ dado um objeto $\mathbf{x}$. Uma vez que possuímos essas estimativas, podemos utilizá-las para classificação ou outras aplicações de suporte à decisão [33].

2.2 Métricas

As métricas de avaliação têm como objetivo medir o desempenho e a eficácia dos modelos de aprendizagem de máquina em várias tarefas. Elas são ferramentas essenciais para comparar e validar os modelos, permitindo uma análise detalhada do seu comportamento e auxiliando na maximização do desempenho. Ao utilizar essas métricas, é possível

selecionar os modelos que melhor se adequam aos objetivos do projeto. É importante salientar que métricas diferentes têm propriedades diferentes, enfatizando aspectos distintos do desempenho do algoritmo de classificação [35].

Para entender melhor as métricas utilizadas, é necessário entender como uma previsão binária pode se enquadrar:

1. **Verdadeiro Positivo (TP)**: O classificador indicou o resultado verdadeiro da classe positiva.
2. **Verdadeiro Negativo (TN)**: O classificador indicou o resultado verdadeiro da classe negativa.
3. **Falso Positivo (FP)**: O classificador indicou que era a classe positiva, mas na verdade era a negativa.
4. **Falso Negativo (FN)**: O classificador indicou que era a classe negativa, mas na verdade era a positiva.

Com essas informações, pode-se gerar uma matriz de confusão, uma tabela que indica os erros e acertos do modelo, comparando com o resultado esperado. Na Figura 5 pode-se observar uma matriz de confusão de classificação binária.

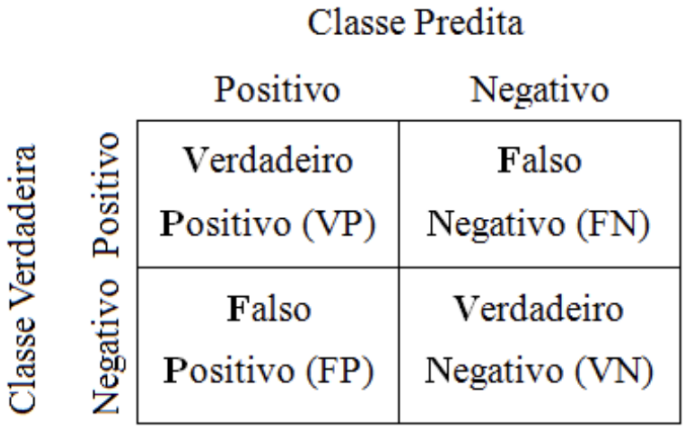


Figura 5: Modelo de matriz de confusão.
Fonte: Adaptado de [34].

Neste projeto, foram utilizadas, a critério de comparação, as métricas **Acurácia** e ***F1 Score***.

A partir da Equação (1), é apresentada a Acurácia, indicando a proporção entre as classes previstas corretamente e todas as instâncias testadas [36].

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Na Equação (2), pode-se observar a representação da métrica Medida-F1, definida como média harmônica das métricas de precisão e revocação [36].

$$Medida - F1 = \frac{2 * TP}{2 * TP + FP + FN} = 2 * \frac{Precisão * \text{Revocação}}{Precisão + \text{Revocação}} \tag{2}$$

Onde, precisão e revocação são definidas nas Equações 3 e 4:

$$Precisão = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Revocação} = \frac{TP}{TP + FN} \tag{4}$$

Dado os métodos necessários para a construção deste trabalho, como a base teórica de aprendizagem de máquina, definição dos modelos e métricas utilizadas, foi selecionada a liga de futebol a ser aplicada a solução deste projeto, denominada *Premier League*.

2.3 *Premier League*

A *Premier League* é a principal liga de futebol da Inglaterra e uma das mais renomadas, assistidas e valiosa de todo o mundo [1]. Fundada em 1992, a competição substituiu a antiga *Football League First Division* e se tornou a primeira divisão do futebol inglês. Composta inicialmente por 22 equipes, a *Premier League* teve o início da sua competição no dia 15 de agosto de 1992 e adota um formato de pontos corridos, no qual todas as equipes se enfrentam em partidas de ida e volta ao longo de uma temporada, que tem duração de aproximadamente 1 ano [2]. Em 2007 a liga teve uma mudança na quantidade de equipes, passando a ser composta por 20 times.

A *Premier League* foi escolhida para ser base desse projeto devido o sua competitividade e nível técnico elevado. Os jogos são disputados em estádios históricos e contam com a presença de jogadores talentosos de diferentes nacionalidades, incluindo algumas das maiores estrelas do futebol mundial. A liga atrai um grande número de espectadores nos estádios e milhões de telespectadores em todo o mundo, sendo transmitida para diversos países.

Em termos de estrutura organizacional, a *Premier League* é administrada pela The Football Association (FA) juntamente com os clubes participantes. A liga estabelece regulamentos e diretrizes para garantir a integridade das competições, como regras de *fair play*[1]financeiro e sistemas de punição para condutas antidesportivas.

[1]Regras que visam garantir a responsabilidade financeira do clubes

No contexto do futebol internacional, a *Premier League* tem um papel de destaque. Os clubes ingleses têm uma longa tradição de sucesso em competições europeias, como a Liga dos Campeões da UEFA e a Liga Europa. O sucesso e a popularidade da *Premier League* contribuem para a projeção global do futebol inglês e atraem jogadores de alto nível de diferentes países.

3 Trabalhos relacionados

Foram feitas buscas nos portais de artigos acadêmicos Google Acadêmico[2], *Research Gate*[3] e *Semantic Scholar*[4], à procura de pesquisas desenvolvidas no ramo da previsão de estatísticas de futebol usando métodos de aprendizagem de máquina, utilizando os termos: *football/soccer match result prediction* (do português, previsão de resultado em partidas de futebol), *football/soccer match classifier* (do português, classificador de partida de futebol) e *predictor premier league matches* (do português, previsor de partidas da primeira liga), para filtrar e obter resultados relacionados ao tema deste trabalho. As pesquisas citadas abaixo trazem contribuições provendo diferentes maneiras de atuação no ramo de classificação de partidas de futebol.

Os autores em [18] propõem a estratégia de predição de partidas de futebol criando o método de **forma**, referente ao desempenho atual da equipe, utilizando dados extraídos do *Football Data*, contendo informações do dia do jogo. A **forma** representa o desempenho dos times nas 7 últimas partidas, atuando como um coeficiente de performance, dando ênfase à fase atual de cada um. Foram testados 8 modelos e comparados entre si, dentre eles se destacando o SVM Linear com taxa de erro de 0,5, *Random Forest* com taxa de erro de 0,49 e o Grandiente Descendente Estocástico (do inglês, *Stochastic Gradient Descent*) com taxa de erro de 0,48. Para o SVM Linear foi apresentada uma matriz de confusão com as predições, sendo 0%, 65,4% e 67,5%, para empate, vitória (mandante) e derrota (mandante), respectivamente. Como medidas de avaliação foram utilizadas a acurácia e a taxa de erro.

O artigo [19] apresenta uma abordagem de análise preditiva para resultados de partidas de futebol da *Premier League*, utilizando diferentes algoritmos de aprendizagem de máquina. Na pesquisa apresentada, é utilizado um conjunto de atributos para determinar os fatores mais importantes para prever os resultados de uma partida de futebol e, consequentemente, criar um sistema preditivo preciso. O estudo utiliza como base de dados informações extraídas de *Football UK*[5], enriquecida por informações do site *Fifa Index*[6]. O trabalho realiza uma análise comparativa entre quatro algoritmos: *Naive Bayes* Gassiano, SVM, *Random Forest* e *Gradient Boosting*. Os resultados apresentados indicam que todos os algoritmos apresentaram baixa taxa de acerto para previsão de empates, onde o *Gradient Boosting* apresentou a melhor taxa de acerto no geral, com acurácia em torno de 58%. Também foi apresentada uma matriz de confusão com as predições para este modelo, sendo 26%, 76,5% e 53,4%, para empate, vitória (mandante) e derrota (mandante), respectivamente.

[2]Google Acadêmico - https://scholar.google.com.br/
[3]Research Gate - https://www.researchgate.net/
[4]Semantic Scholar - https://www.semanticscholar.org/
[5]Football UK - http://www.bbc.com/sport/football
[6]Fifa Index - https://www.fifaindex.com

Como medidas de avaliação foram usadas a métricas de acurácia, precisão, *recall* e *f1-Score*.

Em [20] foi realizada a comparação entre Redes Bayesianas e outros modelos de aprendizagem de máquina: Árvore de Decisão (do inglês, *Decision Tree*), *Naive Bayes*, *Data Driven Bayesian*, Rede Bayesiana (do inglês, *Bayesian Network*) e o K-NN. Como estratégia, os autores incluíram as informações de ranking do time contra e de presença e posição em campo de jogadores importantes. Como medida de validação, foi utilizada a acurácia e taxa de erro. Entre todos os modelos citados, o de melhor resultado foi a *Bayesian Network*, com acurácia de 59% (porcentagem média de todos os testes), no âmbito de previsão de resultados de partidas de futebol. Para isso, foi utilizado como base de dados todas as partidas do clube inglês *Tottenham Hotspur Football Club* entre 1995 e 1997.

No primeiro e segundo trabalho relacionado, podemos compará-los usando as acurácias obtidas a partir de suas matrizes de confusão. Como a previsão de estatísticas de futebol é um problema difícil de prever, principalmente quando se refere à classe de empate, é interessante analisar o desempenho do modelo na previsão das classes individuais. Em [18], a acurácia para a previsão de vitória ou derrota do mandante mostrou-se em torno de 65-67%, enquanto em [19], a previsão de vitória do mandante foi significativamente maior em relação à derrota (76,5% - 53,4%). Ambos os modelos apresentaram desempenho insatisfatório na previsão da classe de empates. Ao comparar com o trabalho apresentado em [20], cuja acurácia para todas as classes foi de 59% no melhor modelo, podemos considerar que o desempenho ficou em uma faixa similar aos outros trabalhos.

Na Tabela 1 podemos observar o resumo de cada solução proposta, a partir de suas bases de dados utilizadas e modelos construídos.

Tabela 1: Resumo dos trabalhos relacionados.

Autor	Descrição	Métricas	Base de Dados	Modelos
ULMER, B. and FERNANDEZ, M., 2014 [18]	Propõe estratégia de predição de resultado de partidas direcionado ao desempenho recente da equipe	Acurácia e taxa de erro	*Football Data*	SVM, *Random Forest* e *Stochastic Gradient Descent*
BABOOTA, Rahul and KAUR, Harleen., 2019 [19]	Comparação de diferentes modelos utilizando um conjunto de recursos para determinar os fatores mais importantes, aplicados para previsão de resultados de partidas	Acurácia, precisão, *recall* e *f1-score*	*Football UK* e *Fifa Index*	*Gaussian Naive Bayes* (GNB), SVM, *Random Forest* e *Gradient Boosting*
JOSEPH, A., FENTON, N. E. and NEIL, M., 2006 [20]	Propõe a comparação entre Redes Bayesianas e outros modelos, utilizando apenas jogos da equipe do *Tottenham* para prever resultados das partidas do time	Acurácia e taxa de erro	Todas as partidas do clube inglês *Tottenham* entre 1995 e 1997.	*Decision Tree*, *Naive Bayesian*, *Data Driven Bayesian* e K-NN
Este Trabalho	Utilização de um conjunto de modelos para analise de segurança de previsão de resultado de partida dos times da *Premier League*	Acurácia e *f1-score*	*Football Data* e site da *Premier League*	*C-Support Vector Classification* (SVC), *XGBoost*, *Multi-layer Perceptron* (MLP), *Random Forest* e *Gaussian Naive Bayes*

Este trabalho propõe a utilização de um conjunto de modelos que, juntos, realizam uma previsão após passar por um método de análise de risco. Diferente do proposto em [19], que compara os métodos, neste projeto os diferentes modelos são unidos e, através do nível de convergência por resultado, é realizada a previsão. Na construção da base de dados, semelhante ao que foi feito em [19], a base foi enriquecida com médias de cada time por temporada para representar melhor o contexto do time por época.

4 METODOLOGIA

Nesse capítulo serão descritos os métodos e ferramentas utilizadas na busca da solução do problema proposto. Serão detalhadas linguagem e todas as bibliotecas utilizadas para o desenvolvimento do projeto e os passos e etapas seguidas ao longo do seu desenvolvimento.

4.1 Ferramentas

Para o desenvolvimento deste projeto, foram utilizados materiais que possibilitaram a construção de um modelo eficiente para análise de segurança sobre previsões de estatísticas esportivas. A linguagem de programação *Python* (Versão 3.6.9) foi utilizada como instrumento principal, com o auxílio de algumas bibliotecas, tais como: *Sklearn* [37] e *Keras*[14], que foram fundamentais na importação e manipulação de modelos; *Numpy*[15], que possibilitou a manipulação e tratamento de matrizes; *Pandas*[16], que foi utilizada para o pré-processamento e manipulação do banco de dados; e *Matplotlib*[17], que permitiu a formação e exibição de gráficos. Essas bibliotecas foram escolhidas pela sua eficiência e facilidade de uso para a criação e manipulação de modelos de aprendizagem de máquina, o que foi fundamental para o desenvolvimento desse projeto.

Para desenvolver, testar e treinar o modelo proposto neste trabalho, foi utilizada a ferramenta *Google Colaboratory* - Colab[7]. Trata-se de um serviço gratuito oferecido pela Google que permite a escrita e execução de códigos *Python*.

4.2 Base de dados

A primeira etapa foi a construção da base dados referente às estatísticas esportivas da competição selecionada para a construção do projeto, a *Premier League*, a liga de futebol mais famosa do Reino Unido. Os dados foram obtidos a partir de sites especializados em estatísticas esportivas e federações esportivas responsáveis pela organização do campeonato.

A base de dados utilizada no projeto teve como origem informações obtidas do site *Football Data*[8], uma página web que disponibiliza uma ampla variedade de estatísticas de diversos campeonatos do mundo inteiro. Para este experimento, foram selecionados dados do mais rico[1] e mais difícil campeonato de futebol do mundo[3], a liga inglesa de futebol, denominada *Premier League*. O *Football Data* oferece planilhas com diversas informações correspondentes a cada jogo, tais como estatísticas, informações correspondentes a cada

[7]Google Colab - https://colab.research.google.com
[8]Football Data - https:///www.football-data.co.uk/

time, árbitro e cotações ligadas a casa de apostas. Todas as partidas estão separadas por temporada.

Inicialmente, parte dos atributos foram selecionados do banco de dados obtido a partir do *Football Data*. Segue a Tabela 2 com os atributos mantidos da base inicial, juntamente com uma breve explicação sobre cada um deles.

Tabela 2: Atributo selecionados da planilha fornecida pelo *Football Data* para compor o dataset do projeto

Nome do atributo	**Descrição**
HomeTeam	Nome do time mandante
AwayTeam	Nome do time visitante
Date	Data da partida
B365H	Cotação da partida na plataforma Bet365 para vitoria do time mandante
B365D	Cotação da partida na plataforma Bet365 para empate
B365A	Cotação da partida na plataforma Bet365 para vitoria do time visitante
Referee	Árbitro da partida
FTResult	Resultado final da partida
Season	Atributo que representa a temporada que cada planilha extraída representa

A coluna *Season* não é um atributo presente na planilha original disponibilizada pelo *Football Data*. Ela foi criada e incorporada ao banco de dados para simbolizar sua planilha de origem, uma vez que cada planilha extraída representa uma temporada específica.

As colunas que representam as cotações da plataforma de apostas BET365 são os valores multiplicadores para cada real apostado. Essa métrica reflete o panorama do jogo, indicando o quão favorito um time é ou o quão equilibrada será a partida. Quanto menor a cotação, mais favorito é o time, enquanto cotações semelhantes indicam um jogo mais equilibrado.

Durante o processo de desenvolvimento, foi notado que as informações disponibilizadas pelo *Football Data* não seriam suficientes para compor os dados de entrada de um modelo de aprendizagem de máquina, uma vez que a maioria dos elementos existentes até o momento eram números pós-jogo e, portanto, potenciais saídas.

Para enriquecer o banco de dados e obter informações sobre cada time, foram retiradas do site da *Premier League*[9] estatísticas de cada clube. Esses atributos foram integrados posteriormente a cada partida como informações referentes aos times, cujo processo será abordado no tópico de pré-processamento, com o objetivo de expressar o comportamento de cada clube na temporada em que o jogo foi realizado. As estatísticas coletadas foram: informações de gols realizados, gols cedidos, escanteios, chutes, cartões amarelos e pontuação no campeonato.

Na composição da base de dados, coletamos informações da *Premier League* de 12 temporadas completas no período de 2009 a 2022, período que contém o total de 4560 partidas.

4.3 Pré-processamento

Os dados coletados passaram por um processo de pré-processamento, que incluiu uma verificação dos dados (onde foram retirados os dados duplicados, inválidos ou incompletos), a seleção de atributos (para reduzir a complexidade dos modelos de previsão) e o balanceamento das classes para compor a base de dados para o treinamento.

O primeiro passo foi a criação do banco de dados final, contendo informações dos sites *Football Data* e *Premier League* juntos, com um total de 12 temporadas e 380 jogos por temporada, totalizando 4560 partidas. O *Football Data* fornece informações por partida, enquanto as informações coletadas a partir do site da *Premier League* são referentes à temporada. Para isso, foi necessária uma etapa para incorporar essas estatísticas coletadas que refletem o desempenho da temporada nas informações jogo a jogo.

Dessa maneira, todos os dados coletados da página da *Premier League* foram transformados em médias para cada partida em que o time joga, representando um coeficiente de gols concedidos, gols realizados, cartões, escanteios e pontuação no campeonato, tanto se o time for mandante ou visitante. Para formar a média, cada estatística foi dividida pelo número de partidas do campeonato, que são 38 rodadas. Um exemplo disto é que, na temporada 2015/2016, o time Arsenal teve um total de 61 gols marcados, e dividindo esse valor por 38 temos um resultado de 1,6 gols por partida. Esse valor é incorporado como média de gols do Arsenal em todas as partidas da temporada 2015/2016 como o atributo *AvgGoalsHome* ou *AvgGoalsAway*, independente se o Arsenal será o mandante ou visitante no jogo em questão, esse valor da média. Essa técnica é necessária para ter uma visão mais completa e detalhada do desempenho dos clubes em cada temporada, permitindo uma melhor análise e previsão de resultados de jogos futuros. Na Tabela 3, observa-se os atributos criados para integrar o banco de dados e uma breve descrição sobre os mesmos.

[9]Premier League - https:///www.premierleague.com/

Tabela 3: Novos atributos criados através de informações retiradas do site da *Premier League* para compor o dataset do projeto.

Nome do atributo	**Descrição**
AvgCornersHome	Média de escanteios do time mandante
AvgCornersAway	Média de escanteios do time visitante
AvgShotsHome	Média de chutes do time mandante
AvgShotsAway	Média de chutes do time visitante
AvgGoalsHome	Média de gols do time mandante
AvgGoalsAway	Média de gols do time visitante
AvgGoalsConHome	Média de gols sofridos do time mandante
AvgGoalsConAway	Média de gols sofridos do time visitante
AvgCardsHome	Média de cartões amarelos do time mandante
AvgCardsAway	Média de cartões amarelos do time visitante
AvgPointsHome	Média de pontos do time mandante
AvgPointsAway	Média de pontos do time visitante

Concluída a construção do banco de dados, onde sua composição final e a junção dos atributos apresentados nas tabelas 2 e 3, foi verificada novamente a existência de instâncias repetidas ou incompletas, para garantir a integridade e a qualidade dos dados utilizados no treinamento.

Foi realizada uma análise de correlação entre os atributos do nosso conjunto de dados, utilizando uma matriz de correlação, demonstrada na Figura 6. Os atributos identificados com mais correlação entre si foram o de cotações da BET365 (B365H, B365A, B365D), o que faz sentido, pois são dados que seguem um padrão, ou seja, quando um time é favorito ou algo que o favorece, a cotação desse time sobe em casas de apostas, e as demais cotas diminuem. Além disso, também foi notada uma alta correlação entre média de pontos no campeonato e média de gols, o que também faz sentido, pois para fazer pontos o time precisa fazer gols. Porém, fazer muitos gols não significa que o time ganhará a partida necessariamente. Analisando a linha da matriz referente ao atributo de saída *FTResult* (Figura 6), nota-se alguns atributos de entrada mais correlacionados a ele, como a cotação de B365, média de pontos, gols e chutes na temporada.

No processo de tratamento de atributos categóricos, foi realizada uma codificação de palavras para números. Essa técnica foi necessária porque alguns atributos são ca-

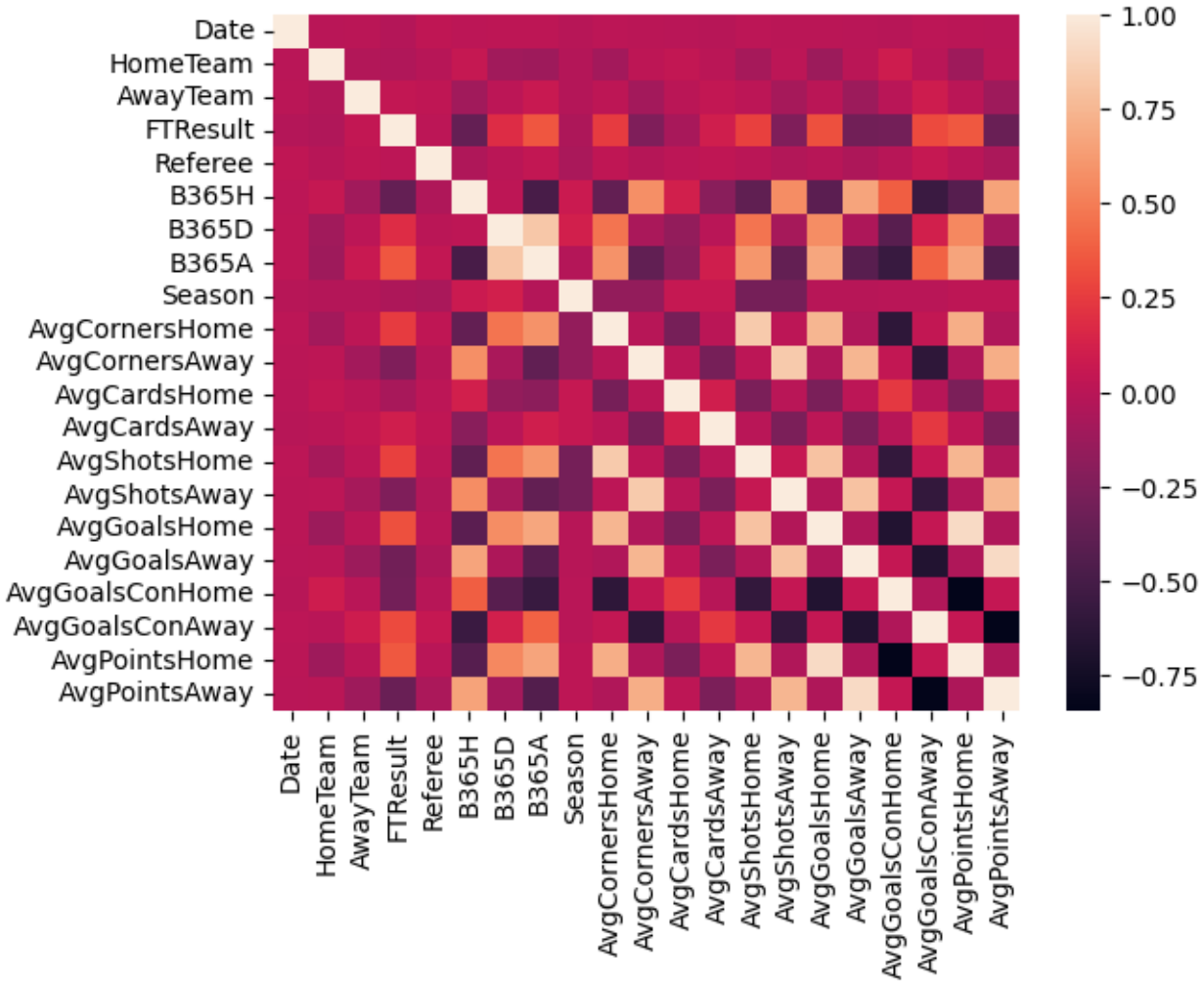


Figura 6: Matriz de correlação.

tegóricos, como o nome do time ou nome do árbitro, e precisam ser convertidos em uma representação numérica para serem usados nos algoritmos de aprendizagem de máquina. Para o campo de *Date*, foi transformada cada data em uma representação numérica crescente, onde jogos mais recentes tem um número maior. Também foi utilizada, alternativamente, a técnica de *One Hot Encoding* em vez de categorização numérica. Porém, como a variação dos atributos é grande, no caso do campo *Date*, por exemplo, que refere-se a data das partidas ao longo de 12 temporadas, o número de colunas do *dataset* multiplicou consideravelmente, elevando bastante o tempo de treinamento dos modelos. Por esse motivo, foi mantida a opção da categorização trocando palavras por números inteiros. Além disso, ao comparar os resultados com as 2 técnicas, o *One Hot Encoded* mostrou-se inferior nos resultados.

Para realizar o treinamento dos modelos selecionados para previsão e futuramente validá-lo, o conjunto de dados foi dividido em um conjunto de treinamento e um conjunto de teste. A divisão foi realizada de forma a utilizar as primeiras 11 temporadas do conjunto de dados para treinamento e a última temporada para teste. Essa abordagem

foi adotada para simular o cenário em que um modelo deve ser desenvolvido com base em dados históricos e aplicado a uma temporada futura para realizar previsões.

O balanceamento foi testado para igualar a quantidade de jogos com resultados de vitória, empate e derrota, sendo este processo realizado apenas no conjunto de treinamento. Isso foi importante porque em um banco de dados desbalanceado, os algoritmos podem ter viés em direção à classe majoritária e não serem capazes de aprender corretamente o padrão das classes minoritárias.

Para a técnica de balanceamento, foram testadas as técnicas de *up sampling*, *down sampling* e nenhum balanceamento. Este tópico foi uma das dificuldades encontradas no desenvolvimento do projeto, pois a classe de empate, além de ser minoritária (1.115 ocorrências), é difícil de ser prevista. Ao aplicar os balanceamentos, os modelos previram mais empates, e consequentemente erraram mais, prejudicando diretamente a acurácia. Porém, em jogos de partidas de futebol, empates acontecem com menor frequência, onde a ocorrência de vitórias é bem maior, seja do time mandante ou visitante. Dessa maneira, após testes realizados, foi preferido manter a base de treinamento sem balanceamento, mesmo que prejudique a classe de empate.

4.4 Treinamento

Neste trabalho, foram treinados cinco algoritmos diferentes, cada um com dez variações de parâmetros, com o objetivo de aumentar a diversidade de modelos e possivelmente melhorar a precisão das previsões. No próximo tópico, onde abordaremos a etapa de teste, será contextualizada a importância de se utilizar modelos diferentes para gerar respostas no conjunto de teste neste projeto.

A Tabela 4 apresenta todos os modelos de aprendizagem de máquina utilizados durante o treinamento, bem como os parâmetros que foram variados em cada um deles e seus respectivos valores. Essas variações foram realizadas pois os métodos apresentam abordagens distintas, podendo apresentar resultados diferentes.

Os parâmetros a serem variados com a finalidade de obter 50 modelos, foram escolhidos a partir de testes de aplicação onde, a partir da variação de valores, o modelo continuasse com uma boa acurácia, sem demonstrar problemas de *overfitting*[10] ou *underfitting*[11] - Desajuste do modelo aos dados, resultando em baixa capacidade de aprendizagem. Para que cada algoritmo pudesse ser variado dez vezes, alguns parâmetros foram variados em conjunto, para que pudesse fazer sentido sua aplicação. Um exemplo disso é no modelo SVC, onde para cada *kernel* variado, foi testado o parâmetro *gamma*, pois

[10] *Overfitting* - Ajuste excessivo de um modelo aos dados de treinamento, prejudicando sua capacidade de generalização.

[11] Underfitting

Tabela 4: Os modelos de aprendizagem de máquina utilizados no trabalho e seus respectivos parâmetros variados.

Modelos de aprendizagem	**Parâmetros variados**	**Valores variados**
C-Support Vector Classification (SVC)	*kernel*, *gamma*, *tolerance*, *class_weight* e *probability*	*kernel* =[rbf, *linear*, *poly*] *gamma*=[*scale*, *auto*] *tol*=$[1 \times 10^{-4}, 1 \times 10^{-5}]$ *class weight*=[*balanced*, *none*] *probability*=[*true*, *false*]
XGBoost	*booster*, *max_depth* e *eta*	*booster* =[*gbtree*, *gblinear*, *dart*] *max depth*=[3,12,18,24] *eta*=[0,1, 0,5, 1}
Multi-layer Perceptron (MLP)	*max iterations*, *hidden layer sizes* e *learning rate*	*max_iter* =[20, 150, 400] *hidden_layer_sizes*=[20, 40, 60, 80] *learning_rate_init*=$[1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}]$
Random Forest	*estimators*	*n_estimators*=[10, 50, 70, 100, 150, 170, 200, 250, 270, 300]
Gaussian Naive Bayes	*variance smoothing*	*var_smoothing*=$[1 \times 10^{-4}, 5 \times 10^{-5}, 1 \times 10^{-5}, 5 \times 10^{-6}, 1 \times 10^{-6}, 5 \times 10^{-7}, 1 \times 10^{-7}, 5 \times 10^{-7}, 1 \times 10^{-8}, 1 \times 10^{-9}]$

são variáveis que podem ser combinadas. No caso do *Random Forest*, por exemplo, foi variado apenas o parâmetro de *estimators*, e a escolha dos valores foi baseada em valores próximos do valor *default* apresentado pela biblioteca, cujo valor é 100.

Todos os modelos e parâmetros, exceto o de *XGBoost*, foram utilizados a partir do *Sklearn* (Versão 1.2.2) [37]. O *XGBoost* foi importado de sua documentação oficial [21].

4.5 Teste

Nesta etapa, como antecipado no tópico anterior, vão ser abordados os testes realizados com os modelos treinados, utilizando um conjunto de dados separado exclusivamente para este fim. Conforme já mencionado anteriormente, foi utilizada a última temporada da *Premier League* presente em nosso banco de dados como conjunto de teste.

Antes de discutirmos a execução dos testes, é necessário definir o conceito de

previsão segura adotado neste projeto. Foi considerada como **previsão segura** os resultados em que $n\%$ das previsões apontam para o mesmo resultado, seja ele mandante, visitante ou empate. O valor de $n\%$ refere-se a um valor de *threshold* a ser variado entre 34% (valor mínimo para formar uma maioria entre as 3 classes) e 100%.

O *threshold* é a medida adotada para representar o mínimo de concordância que as previsões devem ter para serem consideradas seguras. Por exemplo, se o *threshold* for definido como 70%, entende-se que, para a previsão da partida ser segura, 70% dos modelos devem ter previsto o mesmo resultado. Ou seja, em um cenário com 50 modelos, 35 devem apontar para a mesma classe.

Após a execução dos testes, teremos uma base de resultados com 50 respostas para cada um dos 380 jogos presentes no conjunto de teste. Em seguida, foi realizada uma etapa de avaliação de risco, na qual foi verificado cada previsão com base nas 50 respostas geradas por instância de teste. Se uma previsão não atingir o status de **previsão segura**, ela é descartada, pois só é considerada para cálculo de acurácia previsões previamente classificadas como seguras. A avaliação de risco é importante para identificar a segurança das previsões, sendo que a premissa adotada é que quanto menor a variabilidade, mais segura é a previsão.

O diagrama de fluxo presente na Figura 7 permite uma visualização de todo o processo desenvolvido nesse trabalho, desde a entrada dos dados até a obtenção dos resultados finais. Isso permite que possamos acompanhar todo o processo de forma mais clara e identificar possíveis pontos de melhoria no futuro.

O processo começa com a obtenção da base de dados da *Premier League*, que contém as informações de todas as 12 temporadas, com 380 jogos cada. Em seguida, a base de dados é submetida a um processo de pré-processamento para gerar um banco de dados de saída. Esse banco de dados é utilizado para treinar 50 modelos com 5 métodos e 10 variações de parâmetros para cada um deles, gerando um total de 50 modelos de previsão.

Os modelos treinados são testados em conjuntos de dados separados exclusivamente para esse fim, gerando uma base de resultados com 50 respostas para cada um dos 380 jogos presentes no conjunto de teste.

Em seguida, é realizada a avaliação de risco para validar se a previsão é segura ou não. Essa validação é feita com base nas 50 respostas geradas por instância de teste, e as previsões que não atingem o status de **previsão segura** são descartadas. Somente as previsões classificadas como seguras são consideradas para o cálculo de acurácia.

Por fim, obtemos o resultado da previsão somente para as previsões que foram classificadas como seguras na etapa anterior.

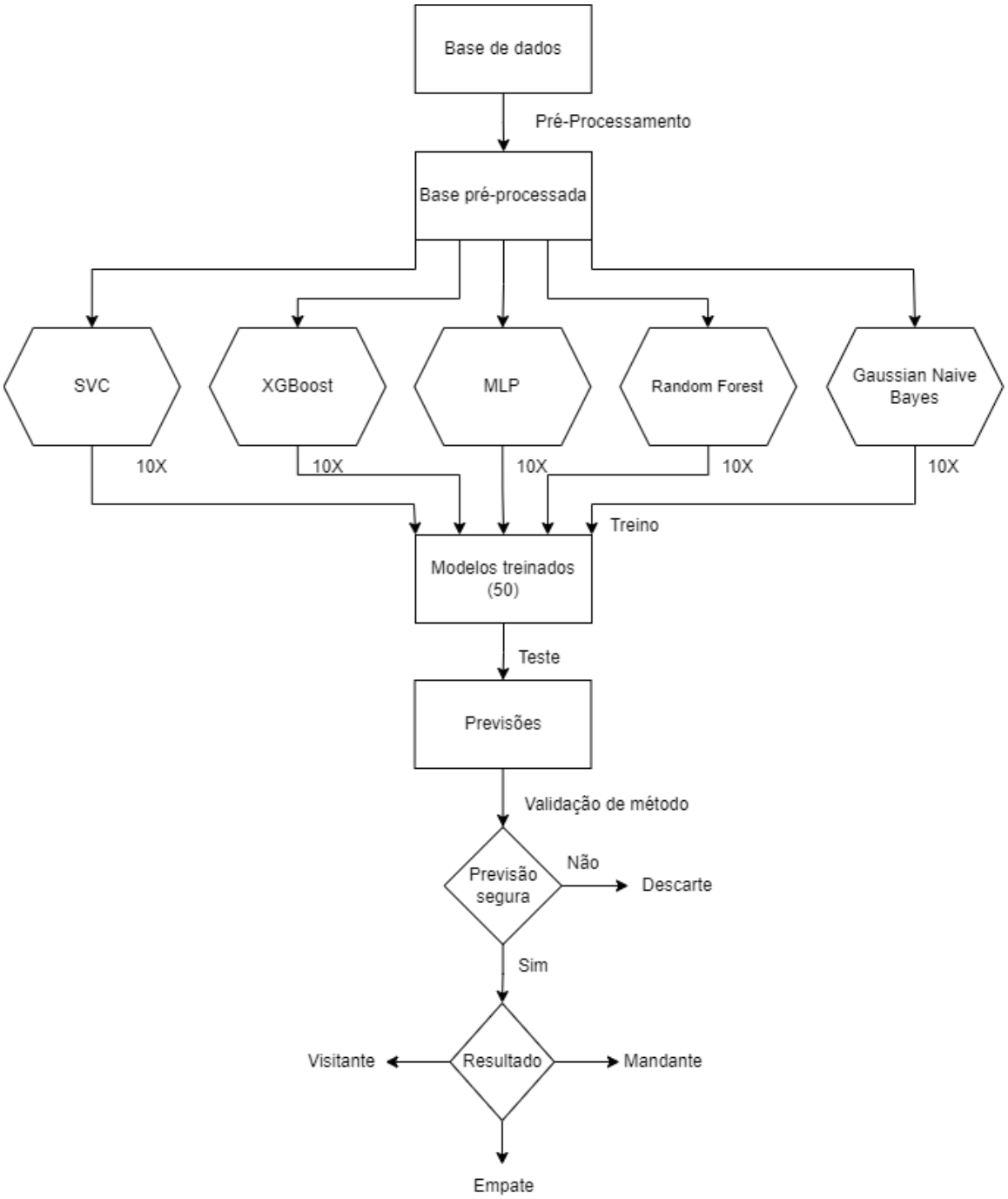


Figura 7: Diagrama de fluxo do processo de previsão.

5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

No presente capítulo, são apresentados os resultados obtidos por meio da aplicação dos modelos de previsão desenvolvidos. Utilizando gráficos, tabelas e matrizes, analisaremos os resultados com base em métricas de avaliação. Serão destacados os principais resultados alcançados, além das características relevantes encontradas durante a elaboração deste trabalho. Essa análise dos resultados fornecerá informações importantes para compreender a eficácia e o desempenho dos modelos implementados.

5.1 Comparação entre diferentes níveis de *threshold*

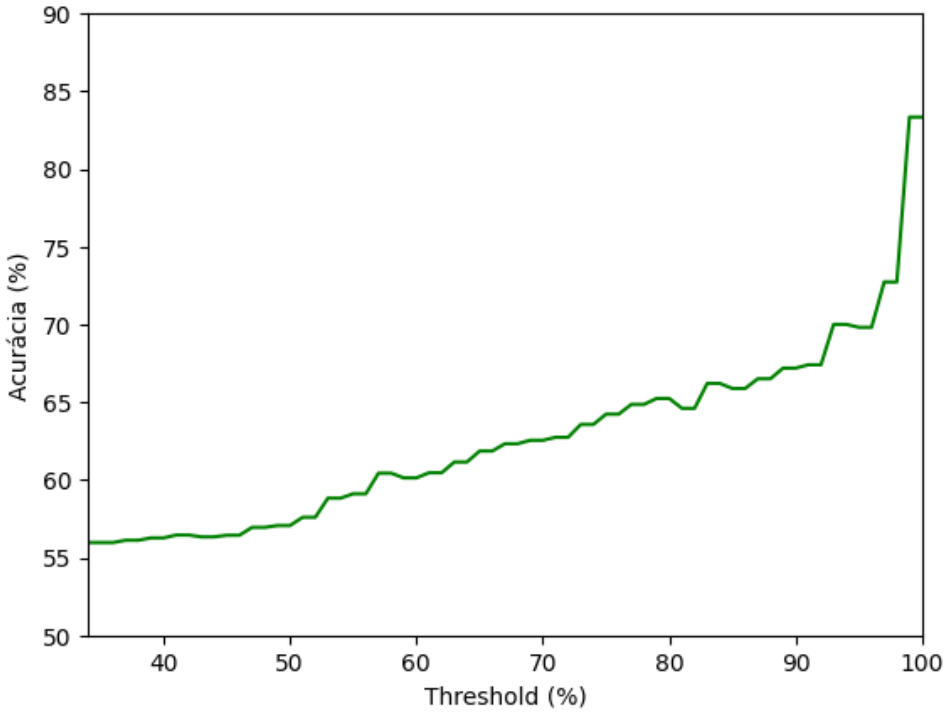
No processo de validação do projeto, foi utilizado um conjunto de 50 modelos, formado por diferentes variações de 5 algoritmos. Esses modelos foram testados em um conjunto de dados composto por uma temporada completa da *Premier League*, especificamente a temporada 2021/2022, que contou com um total de 380 jogos. Através dessa validação, foi possível avaliar o desempenho e a eficácia dos modelos desenvolvidos.

Para entender a influência que o *threshold* tem sobre os resultados obtidos, foi feita a análise variando o *threshold* de 34%, valor mínimo para compor uma maioria de previsões em uma determinada classe, em um problema de 3 classes, até 100%, cenário onde todos modelos apontaram o mesmo resultado.

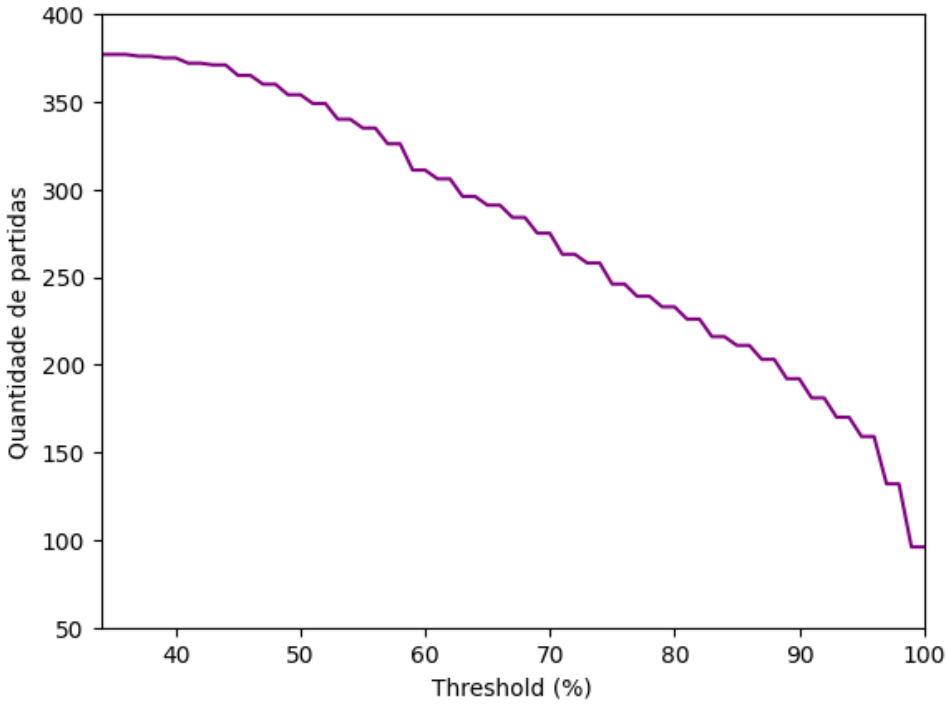
O gráfico apresentado na Figura 8 foi gerado com a finalidade de aprofundar a análise com foco no entendimento do comportamento do modelo conforme o *threshold* varia. Nele, através da Figura 8a, é possível observar a influência do *threshold* na acurácia do modelo, ou seja, quanto maior o nível de concordância entre as predições, maior a probabilidade de uma previsão correta. Ao mesmo tempo, também é possível visualizar na Figura 8b a queda do número de partidas consideradas seguras para uma previsão, conforme o nível de *threshold* aumenta.

Um ponto importante a se destacar é a evolução do gráfico conforme a taxa de *threshold* cresce, onde podemos observar uma relação inversa entre acurácia e a quantidade de partidas apontadas como seguras. O gráfico de acurácia sobe de maneira relativamente constante, assim como a quantidade de partidas consideradas seguras tem uma queda relativamente contínua. Outro ponto a ser destacado é o comportamento dos gráficos no ponto em que a taxa passa do valor de 90%. Ocorre que os resultados começam a sofrer uma variação maior, com um aumento brusco na acurácia, assim como uma queda mais acentuada na quantidade de jogos considerados seguros para palpite.

Para uma análise mais detalhada dos gráficos, a Tabela 5 possui um resumo com pontos chaves do gráfico. O ponto de partida da taxa de *threshold* utilizada nos testes é de 34%. Em casos que mais de uma classe passe de uma faixa de taxa, a previsão considerada



(a) Variação da acurácia.



(b) Variação de partidas.

Figura 8: Variação conforme mudança do *threshold*.

será a que estiver em maioria. Dos 380 casos de teste iniciais, 3 foram descartados de início, pois houve um empate entre duas classes, não conseguindo estabelecer a maioria, e sendo assim, a lógica foi de descartar a previsão.

Tabela 5: Demonstrativo de número de partidas e métricas com a variação do *threshold*.

Threshold	**Partidas**	**Acurácia**	***F1 Score***
34%	377	55,9%	49,5%
40%	375	56,2%	49,7%
50%	354	57,0%	49,8%
60%	311	60,1%	52,4%
70%	275	62,5%	55,4%
80%	233	65,2%	58,2%
90%	192	67,1%	60,4%
95%	159	69,8%	63,0%
97%	132	72,7%	66,3%
100%	96	83,3%	79,1%

Outro ponto a se destacar foi o resultado com taxa de 100% de *threshold*, onde foi possível determinar que 96 partidas, 25% da amostra original, tem uma previsão considerada segura, com a acurácia chegando a 83,3%. Comparando com os trabalhos relacionados temos uma variação positiva na acurácia, onde todos os resultados apresentados tinham um resultado aproximado de 60%. Isso pode ser justificado pelo diminuição do jogos previstos devido a taxa de *threshold*.

5.2 Análise da classe de empate

Durante os testes, foi observada uma grande dificuldade na previsão da classe de empate, um desafio comum em problemas semelhantes, conforme mencionados no capítulo de Trabalhos Relacionados. A Tabela 6 apresenta uma análise detalhada da acurácia dos testes para cada classe.

A Tabela 6 foi elaborada para analisar a relação entre o número de partidas e a acurácia para cada classe, considerando diferentes valores de *threshold*. As linhas referentes aos tipos de partida indicam a quantidade de partidas de cada classe que se enquadram em determinado *threshold*. Já as linhas de acurácia mostram a acurácia obtida para cada classe, levando em conta o respectivo *threshold*. Por meio dessa análise, podemos observar a redução do número de partidas, à medida que o *threshold* aumenta, acompanhada de um aumento na acurácia.

Esses resultados evidenciam a complexidade em prever corretamente a classe de empate.

Tabela 6: Demonstrativo de acurácia e quantidade de partidas por classe com a variação do *threshold*.

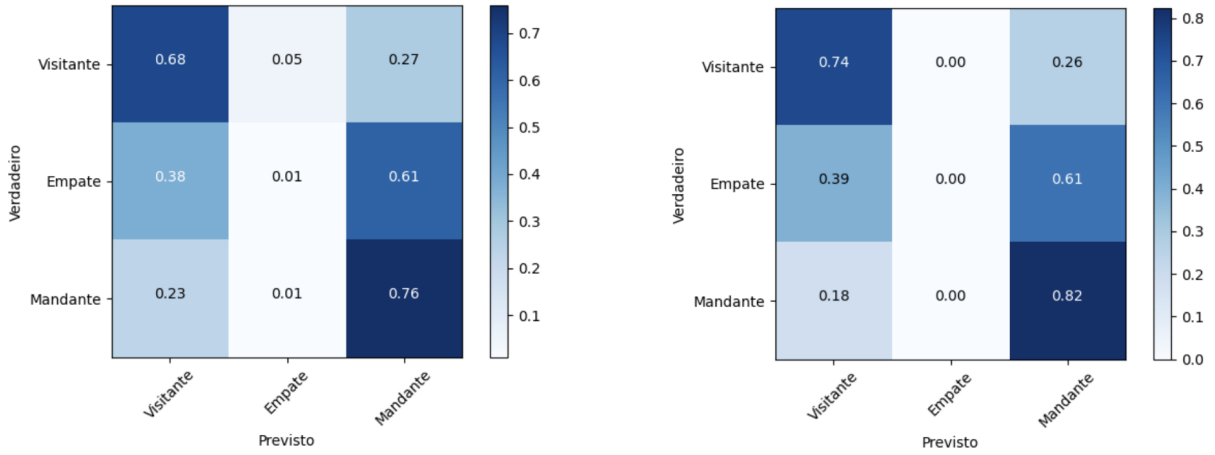
Treshold	Mandante		Visitante		Empate	
	Partidas	Acurácia	Partidas	Acurácia	Partidas	Acurácia
34%	161	76%	129	68%	87	1%
70%	124	82%	95	74%	56	0%
95%	78	87%	53	81%	28	0%
100%	59	97%	28	82%	9	0%

Ao comparar a acurácia total apresentada na Tabela 5, observamos o impacto significativo que os empates têm no resultado geral. A variação do *threshold* desempenha um papel crucial na redução desse impacto, pois os jogos com resultado de empate são gradualmente descartados devido à falta de concordância entre as previsões dos modelos. Isso ocorre devido à dificuldade em prever empates, resultando em previsões altamente variadas e menos confiáveis. À medida que o *threshold* aumenta e a quantidade de jogos com resultado de empate diminui, a relevância do erro na previsão de empates é diminuída, reduzindo assim o impacto na acurácia total.

É importante ressaltar que foram realizadas tentativas de balanceamento por meio da aplicação de técnicas como *down sampling* e *up sampling* para melhorar o desempenho das previsões da classe de empate, visto que existe um desequilíbrio entre as classes, onde o resultado de empate é menos frequente, comparado com os outros dois resultados possíveis. Embora tenha havido uma pequena melhora no desempenho da classe de empate, seu desempenho ainda é considerado baixo. Além disso, essa tentativa de balanceamento acabou impactando negativamente a assertividade geral, devido à queda de desempenho nas classes de mandante e visitante.

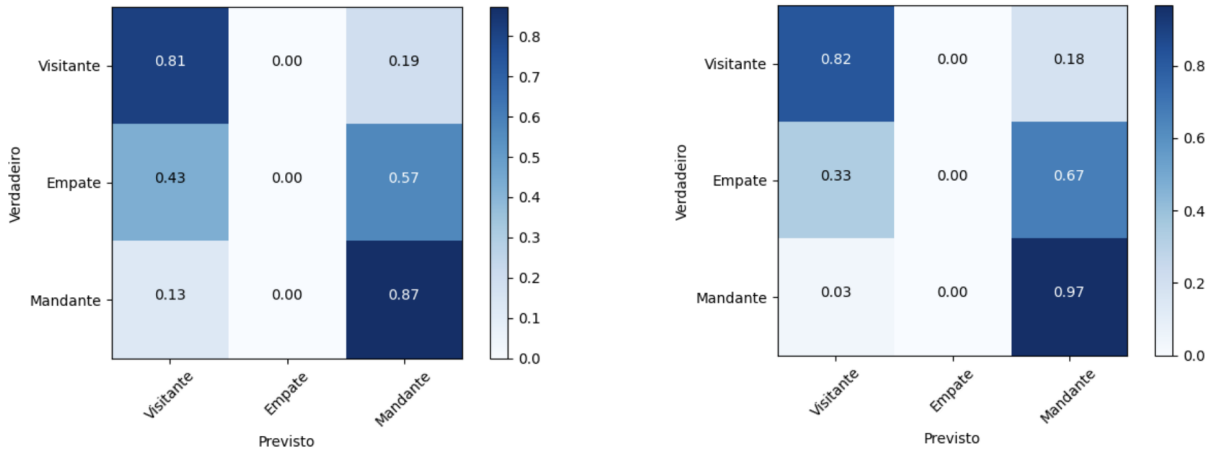
Ainda em relação a Tabela 6, é importante destacar o resultado obtido para a previsão da classe de mandante. Inicialmente, a acurácia foi de 76%, porém, à medida que a taxa de *threshold* aumenta, houve também um aumento gradual no desempenho, alcançando uma acurácia máxima de 97% quando o mesmo atingiu seu valor máximo de 100%.

Na Figura 9, é apresentada a matriz de confusão que corresponde aos valores de *threshold* (34%, 70%, 95%, 100%) mencionados na Tabela 6. Por meio dessa representação gráfica, é possível observar visualmente a variação positiva das acurácias nas classes de



(a) ***Threshold* = 34%**

(b) ***Threshold* = 70%**



(c) ***Threshold* = 95%**

(d) ***Threshold* = 100%**

Figura 9: Matrizes de confusão, com valores normalizados, geradas a partir da variação do *threshold*.

mandante e visitante, enquanto é evidenciada a persistente baixa performance na previsão da classe de empate, à medida que a taxa de *threshold* aumenta. É importante salientar que as matrizes de confusão estão com valores normalizados (entre 0-1), para obtermos uma melhor visualização da proporção de acerto por classe.

5.3 Comparação entre resultados com variações de *threshold* e melhor modelo individual

Dentre os modelos individuais testados, o *XGBoost* teve o melhor desempenho, configurado com o parâmetro *booster=gblinear*, alcançando uma acurácia total de 58%. Ao comparar com os resultados obtidos utilizando a Tabela 5, é constatado que o *XGBoost* teve a acurácia superada em todos os testes que tiveram a taxa de *threshold* igual ou superior a 60%.

Comparando o resultado de acurácia por classes do modelo *XGBoost* com a melhor

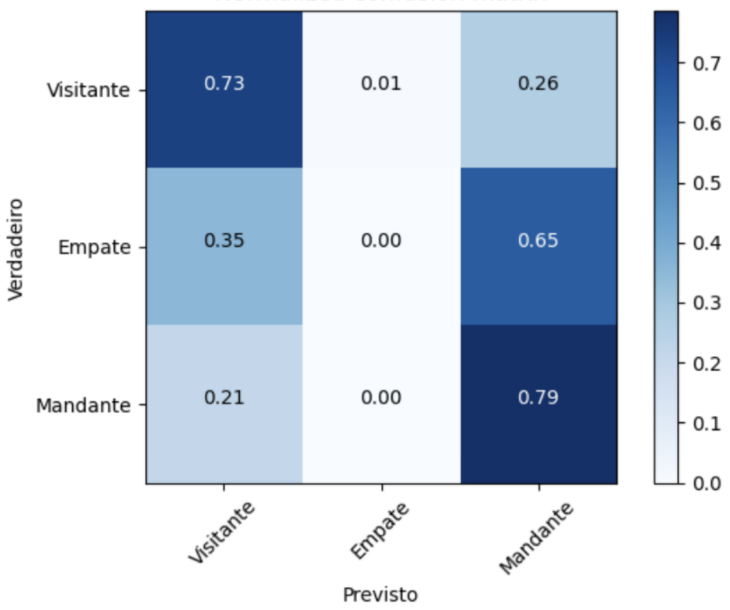


Figura 10: Matriz de confusão do *XGBoost* com valores normalizados.

performance apresentada na Figura 10, utilizando a Tabela 6, vemos que todas as classes tem o resultado abaixo, comparados aos níveis a partir de 70% de *threshold*.

6 CONCLUSÕES E TRABALHOS FUTUROS

Ao longo deste trabalho foram exploradas aplicações de diferentes algoritmos de aprendizado de máquina para a previsão de resultados em jogos de futebol. Foram utilizadas técnicas para combinar modelos, e através da aplicação de uma análise de concordância, o objetivo almejado era melhorar a precisão das previsões e identificar a incerteza associada a elas. Nossos resultados revelaram pontos interessantes e desafios significativos nesse contexto, destacando a dificuldade em prever empates e o comportamento do resultado final de acordo com a variação do *threshold*.

Através da análise da acurácia por classe, foi possível observar consistentemente um desempenho insatisfatório na previsão da classe de empate. Essa dificuldade pode ser atribuída à natureza imprevisível e complexa dos jogos de futebol, especialmente quando se trata de prever um resultado menos comum, como o empate. Além disso, outro ponto que pode ser determinante para dificuldade de se prever empate é o fato dele representar um resultado intermediário aos outros resultados possíveis, pois apenas um gol pode diferenciá-lo de uma vitória para uma das equipes, mandante ou visitante. Essa característica de resultado que demonstra equilíbrio entre as equipe, torna a concordância entre os modelos do conjunto mais desafiadora.

Buscando atenuar o desequilíbrio entre as classes, foram realizadas tentativas de balanceamento utilizando técnicas como *down sampling* e *up sampling*. Embora tenhamos alcançado uma pequena melhora no desempenho da classe de empate, essas estratégias não tiveram um impacto positivo na acurácia geral, comprometendo o desempenho das classes de mandante e visitante, onde seguir o treinamento com as classes desbalanceadas foi a estratégia seguida no desenvolvimento desse projeto.

A análise da concordância dos resultados dos modelos de classificação revelou-se fundamental para a precisão das previsões. Ao avaliarmos esses resultados, constatamos uma melhoria gradual na acurácia, chegando a atingir valores próximos a 100% de acurácia máxima. É importante ressaltar a relação inversa entre a quantidade de jogos considerados seguros para previsão e a acurácia do modelo. Essa relação é controlada pelo parâmetro de *threshold*, que regula a concordância dos modelos. Quanto maior o valor do *threshold*, maior a acurácia do modelo, porém, menor será a quantidade de jogos considerados seguros para previsão.

6.1 Trabalhos futuros

Como trabalho futuro, sugere-se a inclusão de outras variáveis e características dos jogos para aprimorar as previsões, por exemplo, considerar dados meteorológicos. O desempenho recente das equipes e estatísticas individuais dos jogadores podem proporcionar

uma representação mais contemporânea da situação. Além disso, é interessante expandir o escopo dos testes e treinamentos para além da *Premier League*, abrangendo outras ligas e competições. Adicionalmente, pode-se realizar previsões para jogos atuais, uma vez que os dados utilizados neste trabalho são referentes à temporada 2021/2022, que encerrou no meio de 2022.

No caso específico da melhoria de desempenho nas previsões de empate, uma possível abordagem é avaliar se existe um equilíbrio nas previsões entre as classes. Isso significa, que se as previsões para as três classes (vitória do mandante, vitória do visitante e empate) estiverem com resultados equilibrado entre as classes, pode significar um equilíbrio entre as equipes e, consequentemente, uma possível indicação de empate. Um exemplo disso é, se entre os 50 modelos, o resultado for de 17 mandantes, 17 visitantes e 16 empates, esse seria um resultado para se avaliar um possível empate. Essa validação nas previsões com um maior equilíbrio entre as classes pode ser explorada como uma estratégia para aprimorar a performance em casos de empate.

REFERÊNCIAS

[1] LANCE!. Ligas nacionais mais valiosas do mundo em 2023: veja levantamento exclusivo do L!. Rio de Janeiro, 2023. Disponível em: <https://www.lance.com.br/lancebiz/as-20-ligas-nacionais-mais-valiosas-do-mundo-veja-levantamento-exclusivo-do-l.html>. Acesso em: 11 Maio 2023.

[2] Premier League. Premier League Origins Disponível em: <https://www.premierleague.com/history/origins>. Acesso em: 22 Maio 2023.

[3] IFFHS News. THE STRONGEST NATIONAL LEAGUE IN THE WORLD 2019 – THE ENGLISH PREMIER LEAGUE NUMBER 1. Disponível em: <https://www.iffhs.com/posts/58>. Acesso em: 11 Maio 2023.

[4] Aventuras na historia. CONHEÇA A HISTÓRIA DAS APOSTAS ESPORTIVAS NO BRASIL. Disponível em: <https://aventurasnahistoria.uol.com.br/noticias/reportagem/conheca-historia-das-apostas-esportivas-no-brasil.phtml>. Acesso em: 13 Maio 2023.

[5] Agrela, Lucas. Alvo do governo, apostas online esportivas movimentam R$ 12 bilhões. São Paulo: Estadão, 2023. Disponível em: <https://economia.uol.com.br/noticias/estadao-conteudo/2023/03/13/sem-regulacao-apostas-online-esportivas-giram-r-12-bilhoes.htm>. Acesso em: 13 Maio 2023.

[6] ÉPOCA NEGÓCIOS ONLINE. Brasileiros perderam US$ 4,1 bilhões em sites de apostas e loterias. Disponível em: <https://epocanegocios.globo.com/Informacao/Resultados/noticia/2015/09/brasileiros-perderam-us-41-bilhoes-em-sites-de-apostas-e-loterias.html>. Acesso em: 13 Maio 2023.

[7] SOCIETY, The Internet. Artificial Intelligence and Machine Learning: Policy Paper. Internet Society, 2017. Disponível em: <https://www.internetsociety.org/resources/doc/2017/artificial-intelligence-and-machine-learning-policy-paper/>. Acesso em: 2 Nov 2022.

[8] BROWN, Sara. Machine learning, explained. MTI Management Sloan School, 2021. Disponível em: <https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained>. Acesso em: 2 Nov 2022.

[9] SAMUEL, A. L.. Some studies in machine learning using the game of checkers. **IBM Journal of Research and Development**, 3, 3, 210-229, Jul, 1959.

[10] SODHI, P. AWASTHI, N. e SHARMA, V.. Introduction to Machine Learning and Its Basic Application in Python. **SSRN Electronic Journal**, Jan, 2019.

[11] BHAVSAR, P., SAFRO, I., et al. Machine Learning in Transportation Data Analytics. **Data Analytics for Intelligent Transportation Systems**, 283–307, Jan, 2019.

[12] PRO, P. Classification vs. Regression Algorithms in Machine Learning. 2022. Disponível em: <https://www.projectpro.io/article/classification-vs-regression-in-machine-learning/545#mcetoc_1fp6av4s6a>. Acesso em: 5 Nov 2022.

[13] Scikit-learn: Machine learning in Python. Disponível em: <https://scikit-learn.org/stable/>. Acesso em: 21 de abril de 2023.

[14] Keras. Disponível em: <https://keras.io/>. Acesso em: 21 abr. 2023.

[15] Numpy. Disponível em: <https://numpy.org/>. Acesso em: 26 jun. 2023.

[16] Pandas. Disponível em: <https://pandas.pydata.org/>. Acesso em: 26 jun. 2023.

[17] Matplotlib. Disponível em: <https://matplotlib.org/>. Acesso em: 26 jun. 2023.

[18] ULMER, B.; FERNANDEZ, M. Predicting Soccer Match Results in the English Premier League. Disponível em: <http://cs229.stanford.edu/proj2014/Ben Ulmer, Matt Fernandez, Predicting Soccer Results in the English Premier League.pdf>. Acesso em: 22 abr. 2023.

[19] BABOOTA, Rahul; KAUR, Harleen. Predictive analysis and modelling football results using machine learning approach for English Premier League. **International Journal of Forecasting**, 35, 2, 741-755, abril-junho de 2019.

[20] JOSEPH, A., FENTON, N. E., NEIL, M. Predicting football results using Bayesian nets and other machine learning techniques. Knowledge-Based Systems, 19(7), 544–553. 2006.

[21] Xgboost Developers. XGBoost Documentation. Disponível em: <https://xgboost.readthedocs.io/en/stable/>. Acesso em: 13 Maio 2023.

[22] VAPNIK, V. N.. The nature of Statistical learning theory. **Springer-Verlag**, New York, 1995.

https://www.mathworks.com/discovery/support-vector-machine.html

[23] Learn optimal hyperplanes as decision boundaries. The MathWorks, Inc. Disponível em: <https://www.mathworks.com/discovery/support-vector-machine.html>. Acesso em: 17 Mai 2023.

[24] LORENA, Ana et al. Uma Introdução às Support Vector Machines **Revista de Informática Teórica e Aplicada**, 14, 2, 43-67, 2007.

[25] CHEN, Tiangi, GUESTRIN, Carlos. XGBoost: A Scalable Tree Boosting System **Association for Computing Machinery**, 785–794, 2016.

[26] Developer Guide: XGBoost Algorithm. Disponível em: <https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost.html>. Acesso em: 17 Mai 2023.

[27] PINKUS, Allan. Approximation theory of the MLP model in neural networks. **Acta Numerica**, 143-195, 1999.

[28] BANOULA, Mayank. An Overview on Multilayer Perceptron (MLP). Simplilearn, 2023. Disponível em: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/multilayer-perceptron>. Acesso em: 17 Mai 2023.

[29] SOBREIRO, Vinicius et al. Uma Estimação Do Valor Da Commodity De Açúcar Utilizando Redes Neurais Artificiais. **Revista P&D em Engenharia de Produção**, 7, 36-52, 2008.

[30] BREIMAN, Leo. Random forests. **Machine learning**, 45, 1, 5-32, 2001.

[31] What is random forest? IBM. Disponível em: <https://www.ibm.com/topics/random-forest>. Acesso em: 17 Mai 2023.

[32] RISH, Irina. An empirical study of the naive Bayes classifier. **In: IJCAI 2001 workshop on empirical methods in artificial intelligence**, S. 41-46, 2001.

[33] WEBB, G.I. Naive Bayes. **Encyclopedia of Machine Learning and Data Mining**. Springer, Boston, 2016.

[34] FILHO, Geraldo et al. ResiDI: Um Sistema de Decisão Inteligente para Infraestruturas Residenciais via Sensores e Atuadores Sem Fio. **XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, 2015.

[35] HAND, D.J., CHRISTEN, P. & KIRIELLE, N. F*: an interpretable transformation of the F-measure. **Mach Learn** 110, 451–456, 2021.

[36] CHICCO, D., JURMAN, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. **BMC Genomics**. 21, 6, 2020.

[37] PEDREGOSA, F. et al. **Scikit-learn: Machine Learning in Python**. 12, 2825–2830, 2011.