Previsão de resultados do campeonato brasileiro de futebol com modelos supervisionados de machine learning

Thales Augusto Oliveira Campolim[1]*; Renato Máximo Sátiro[2]

[1] NTT DATA. Cientista de dados. Av. Nações Unidas,14171 – Torre B – 16 andar – Itaim Bibi; 04795-100 São Paulo, SP, Brasil
[2] Doutor em Administração pela Universidade Federal de Goiás. Professor Orientador, Escola Superior de Agricultura "Luiz de Queiroz", ESALQ/USP. Rua Alexandre Herculano, 120, Vila Monteiro; 13418-445 Piracicaba, São Paulo, Brasil

Previsão de resultados do campeonato brasileiro de futebol com modelos supervisionados de machine learning

Resumo

Considerando a influência que a tecnologia e as Ciências de Dados têm apresentado nos esportes nos dias de hoje, esse estudo busca utilizar diferentes tipos de algoritmos de machine learning disponíveis para desenvolver modelos preditivos de resultados de partidas de futebol. Utilizando dados das temporadas anteriores, foram aplicados algoritmos de Regressão Logística Multinominal, Random Forest, e XGB com o objetivo de avaliar qual modelo apresentava o melhor ajuste aos dados utilizados. Os modelos baseados em arvores de classificação desempenharam bem melhor que o regressivo, que apresentou uma baixa capacidade preditiva. De qualquer forma, os resultados alcançados pelos algoritmos de Random Forest podem ser considerados apenas mediano. Dessa forma, o presente estudo se propõe a discutir machine learning aliado aos esportes, e contribuir com a aplicação de tais conceitos em diferentes contextos dos já estudados.
Palavras-chave: Machine Learning, Regressão logística Multinominal, Random Forest, Futebol, Campeonato Brasileiro

Abstract

Keywords ou **Palabras Clave:** #logisticregression #randomforest #predictivemodels #football

Introdução

O futebol foi uma das áreas que se desenvolveu a partir do avanço da tecnologia em aspectos que vão desde o desenvolvimento tático da equipe, até o físico dos jogadores (Afonso et al., 2020). Para além dos campos, os dados dos jogos vêm sendo utilizados com o objetivo de prever desde estatísticas das partidas, até o próprio resultado.

Para Sehnem & Frozza (2019), mesmo sendo um esporte coletivo, que possui uma infinidade de fatores que podem influenciar o resultado de uma partida de futebol como, expulsões, cartões, erros do arbitro, e que ocorrem de forma aleatória, é possível calcular a probabilidade de um resultado acontecer baseado nas partidas anteriores.

Apesar de Gifford & Bayrak argumentarem sobre o fato do desenvolvimento de modelos voltados especificamente para a identificação dos fatores de maior impacto no resultado do jogo ainda ser limitado quando comparado com as análises utilizadas pelas comissões técnicas e dirigentes, enquanto Herold et al. (2019) traz um artigo onde discute a situação atual dos estudos voltados para machine learning no futebol masculino, e listam uma série de iniciativas e seus resultados.

Com a intenção de trabalhar mais de um modelo e comparar o desempenho entre eles, a escolha da abordagem utilizada foi fundamentada na característica da variável de interesse. Considerando que são possíveis três resultados distintos em uma partida, sendo a vitória de cada time ou o empate, baseado nos conceitos apresentados por Favero & Belfiore (2017) podemos dizer que a variável assume um caráter qualitativo e policotômico, sendo esse um comportamento comum aos utilizados nos modelos de Regressão Logística Multinominal. O segundo algoritmo escolhido foi o de Random Forest, apresentado por Breiman (2001), pois além de também trabalhar com dados do mesmo caráter do algoritmo anterior, possui robustez em lidar com uma grande quantidade de variáveis na base de dados utilizada, e desempenha bem em casos de desbalanceamento de classes.

Esse trabalho tem como objetivo aplicar esses dois modelos em uma base de dados com estatísticas dos anos anteriores do campeonato até o momento e comparar o desempenho de ambos quanto ao potencial de prever o resultado de uma partida.

Material e Métodos

O presente estudo utiliza como base inicial o resultado e as estatísticas complementares quatro mil quatrocentas e vinte e uma partidas realizadas pelo campeonato brasileiro da “Série A”, ocorridas entre o ano de 2013 e setembro de 2024 e disputadas por 35 clubes diferentes durante esse período. O campeonato ocorre anualmente por sistema de pontos corridos, e reúne 20 clubes por temporada com turno e returno de forma contínua, onde cada time jogará uma partida em casa e uma fora com cada adversário somando trinta e oito jogos para cada um dos times participantes (CBF,2024).

Como Gil (2022) definiu, este estudo pode ser classificado como descritivo, já que buscou explicar a influência de diferentes variáveis explicativas que causam um evento específico, que nesse caso se dá pelo resultado de uma partida de futebol, e de natureza quantitativa uma vez que se utiliza de modelos matemáticos aplicados a dados estruturados para atingir o objetivo inicial.

Segundo os conceitos apresentados por Malhotra (2012) a coleta dos dados foi feita de forma secundária, considerando que a observação dos eventos, tabulação, e disponibilização foi feita por uma segunda parte. Os dados foram coletados a partir de uma plataforma online que reúne estatísticas de partidas, atletas, e campeonatos ao redor do mundo chamada Footystats através do link https://footystats.org/u/1788116. A plataforma oferece download direto de arquivos CSVs em língua inglesa com as estatísticas de todas as partidas de cada temporada.

Após o download dos arquivos, toda a modelagem e aplicação do algoritmo foram desenvolvidas em Python, versão 3.11.7, utilizando a plataforma de desenvolvimento integrada para Visual Studio Code 1.93.1 para compilar o código. Os documentos CSV foram agrupados para a formação de um único banco de dados utilizando a biblioteca Pandas. Além disso, como os arquivos traziam todas as partidas programadas para ocorrer durante o campeonato, para os dados de 2024 foram filtradas apenas as partidas já finalizadas através da coluna "status" com o valor “complete”.

Em caráter exploratório, foram utilizadas outras funções do Pandas, como a “dtype”, para listagem das colunas e dos tipos de dados disponíveis.

TABELA 1: Conjunto de dados e tipo dos valores

Coluna	Tipo
timestamp	int64
date_GMT	object
status	object
attendance	float64
home_team_name	object
away_team_name	object
referee	object
Game Week	int64
Pre-Match PPG (Home)	float64
Pre-Match PPG (Away)	float64
home_ppg	float64
away_ppg	float64
home_team_goal_count	int64
away_team_goal_count	int64
total_goal_count	int64
total_goals_at_half_time	int64
home_team_goal_count_half_time	int64
away_team_goal_count_half_time	int64
home_team_goal_timings	object
away_team_goal_timings	object
home_team_corner_count	int64
away_team_corner_count	int64
home_team_yellow_cards	int64
home_team_red_cards	int64
away_team_yellow_cards	int64
away_team_red_cards	int64
home_team_first_half_cards	int64
home_team_second_half_cards	int64
away_team_first_half_cards	int64
away_team_second_half_cards	int64
home_team_shots	int64
away_team_shots	int64

TABELA 1: Conjunto de dados e tipo dos valores

Coluna	Tipo
home_team_shots_on_target	int64
away_team_shots_on_target	int64
home_team_shots_off_target	int64
away_team_shots_off_target	int64
home_team_fouls	int64
away_team_fouls	int64
home_team_possession	int64
away_team_possession	int64
Home Team Pre-Match xG	float64
Away Team Pre-Match xG	float64
team_a_xg	float64
team_b_xg	float64
average_goals_per_match_pre_match	float64
btts_percentage_pre_match	int64
over_15_percentage_pre_match	int64
over_25_percentage_pre_match	int64
over_35_percentage_pre_match	int64
over_45_percentage_pre_match	int64
over_15_HT_FHG_percentage_pre_match	int64
over_05_HT_FHG_percentage_pre_match	int64
over_15_2HG_percentage_pre_match	int64
over_05_2HG_percentage_pre_match	int64
average_corners_per_match_pre_match	float64
average_cards_per_match_pre_match	float64
odds_ft_home_team_win	float64
odds_ft_draw	float64
odds_ft_away_team_win	float64
odds_ft_over15	float64
odds_ft_over25	float64
odds_ft_over35	float64
odds_ft_over45	float64
odds_btts_yes	float64
odds_btts_no	float64
stadium_name	object

Fonte: Base de dados coletada

Ainda no reconhecimento da característica dos dados, com a função “describe” foram listadas as estatísticas descritivas das variáveis numéricas. Foi possível identificar inconsistências na amostra, por exemplo, número de observações inferior ao esperado como no caso de “attendance”, e variáveis que registraram valores mínimos negativos, o que não é possível considerado o contexto.

Para modelagem da amostra foram descartadas as variáveis que continham as quatro mil quatrocentas e vinte e uma observações e, foram removidas as observações nas quais continham alguma estatística negativa.

TABELA 2: Estatísticas descritivas da amostra

Coluna	contagem	média	std	min	25%	50%	75%	max
timestamp	4421	1,50E+09	1,50E+09	1,00E+09	1,40E+09	1,50E+09	1,60E+09	1,70E+09
attendance	2459	17533,8	13330	-1,0	7995	13637	24449	69846
Game Week	4421	19,1	10,9	1,0	10,0	19,0	28,0	38,0
Pre-Match PPG (Home)	4421	1,6	0,7	0,0	1,2	1,7	2,1	3,0
Pre-Match PPG (Away)	4421	0,9	0,6	0,0	0,5	0,9	1,3	3,0
home_ppg	4421	1,7	0,4	0,3	1,5	1,7	2,0	2,8
away_ppg	4421	1,0	0,4	0,2	0,7	1,0	1,3	2,0
home_team_goal_count	4421	1,4	1,2	0,0	1,0	1,0	2,0	7,0
away_team_goal_count	4421	1,0	1,0	0,0	0,0	1,0	1,0	6,0
total_goal_count	4421	2,4	1,5	0,0	1,0	2,0	3,0	10,0
total_goals_at_half_time	4421	1,1	1,0	0,0	0,0	1,0	2,0	7,0
home_team_goal_count_half_time	4421	0,6	0,8	0,0	0,0	0,0	1,0	5,0
away_team_goal_count_half_time	4421	0,4	0,6	0,0	0,0	0,0	1,0	4,0
home_team_corner_count	4421	5,9	3,1	-1,0	4,0	6,0	8,0	23,0
away_team_corner_count	4421	4,6	2,6	-1,0	3,0	4,0	6,0	18,0
home_team_yellow_cards	4421	2,2	1,4	0,0	1,0	2,0	3,0	10,0
home_team_red_cards	4421	0,1	0,3	0,0	0,0	0,0	0,0	3,0
away_team_yellow_cards	4421	2,5	1,5	0,0	1,0	2,0	3,0	9,0
away_team_red_cards	4421	0,2	0,4	0,0	0,0	0,0	0,0	3,0
home_team_first_half_cards	4421	0,8	0,9	0,0	0,0	1,0	1,0	6,0
home_team_second_half_cards	4421	1,6	1,2	0,0	1,0	1,0	2,0	7,0
away_team_first_half_cards	4421	1,0	0,9	0,0	0,0	1,0	1,0	5,0
away_team_second_half_cards	4421	1,7	1,3	0,0	1,0	2,0	2,0	9,0
home_team_shots	4421	13,1	5,0	-1,0	10,0	12,0	16,0	37,0
away_team_shots	4421	10,4	4,6	-1,0	7,0	10,0	13,0	102,0

TABELA 2: Estatísticas descritivas da amostra

Coluna	contagem	média	std	min	25%	50%	75%	max
home_team_shots_on_target	4421	5,6	2,5	-1,0	4,0	5,0	7,0	19,0
away_team_shots_on_target	4421	4,5	2,2	-1,0	3,0	4,0	6,0	19,0
home_team_shots_off_target	4421	7,5	3,9	-1,0	5,0	7,0	10,0	30,0
away_team_shots_off_target	4421	6,0	3,6	-1,0	4,0	5,0	8,0	96,0
home_team_fouls	4421	14,9	4,7	-1,0	12,0	15,0	18,0	33,0
away_team_fouls	4421	14,8	4,8	-1,0	12,0	14,0	18,0	34,0
home_team_possession	4421	51,4	9,9	-1,0	45,0	52,0	58,0	100,0
away_team_possession	4421	48,2	9,8	-1,0	42,0	48,0	54,0	88,0
Home Team Pre-Match xG	4421	0,9	0,9	0,0	0,0	1,3	1,7	3,1
Away Team Pre-Match xG	4421	0,7	0,7	0,0	0,0	1,1	1,4	2,7
team_a_xg	4421	1,0	0,9	0,0	0,0	1,1	1,7	3,9
team_b_xg	4421	0,8	0,8	0,0	0,0	0,9	1,4	7,4
average_goals_per_match_pre_match	4421	2,3	0,7	0,0	2,0	2,3	2,6	6,5
btts_percentage_pre_match	4421	45,4	19,2	0,0	37,0	47,0	56,0	100,0
over_15_percentage_pre_match	4421	64,9	21,3	0,0	58,0	68,0	77,0	100,0
over_25_percentage_pre_match	4421	40,8	18,4	0,0	32,0	42,0	50,0	100,0
over_35_percentage_pre_match	4421	19,8	13,6	0,0	12,0	20,0	27,0	100,0
over_45_percentage_pre_match	4421	8,6	9,2	0,0	0,0	7,0	13,0	100,0
over_15_HT_FHG_percentage_pre_match	4421	27,8	15,8	0,0	19,0	28,0	37,0	100,0
over_05_HT_FHG_percentage_pre_match	4421	62,7	20,7	0,0	55,0	66,0	75,0	100,0
over_15_2HG_percentage_pre_match	4421	35,2	17,1	0,0	27,0	36,0	46,0	100,0
over_05_2HG_percentage_pre_match	4421	70,1	21,4	0,0	65,0	74,0	81,0	100,0
average_corners_per_match_pre_match	4421	9,9	3,0	0,0	9,1	10,3	11,6	23,0

TABELA 2: Estatísticas descritivas da amostra

Coluna	contagem	média	std	min	25%	50%	75%	max
average_cards_per_match_pre_match	4421	4,9	1,6	0,0	4,3	5,0	5,8	13,0
odds_ft_home_team_win	4421	2,3	0,9	0,0	1,7	2,1	2,6	20,5
odds_ft_draw	4421	3,5	0,7	0,0	3,2	3,4	3,7	10,0
odds_ft_away_team_win	4421	4,3	2,2	0,0	2,9	3,8	5,3	22,3
odds_ft_over15	4421	0,8	0,7	0,0	0,0	1,3	1,4	2,5
odds_ft_over25	4421	1,3	1,1	0,0	0,0	1,9	2,3	3,5
odds_ft_over35	4421	2,4	2,1	0,0	0,0	3,3	4,2	8,0
odds_ft_over45	4421	4,4	4,3	0,0	0,0	5,7	8,0	34,0
odds_btts_yes	4421	1,2	1,0	0,0	0,0	1,9	2,1	3,2
odds_btts_no	4421	1,0	0,9	0,0	0,0	1,6	1,7	2,7

Fonte: Análise exploratória dos dados

Apesar de trazer diversas variáveis explicativas sobre os jogos, não há, no entanto, uma coluna na base de dados que represente o resultado da partida, que é a variável de interesse. Com isso, foi necessário criar uma medida a partir da combinação das outras variáveis. Se o valor do campo "home_team_goal_count" fosse superior ao do campo "away_team_goal_count", seria atribuído o valor de 'Time da casa vencedor'. Alternativamente, se o valor do campo estivesse abaixo, o valor seria 'Time visitante vencedor'. Se o valor de ambos os campos fosse igual, o campo de resultado receberia o valor 'Empate'.

Para efeitos de atribuição da variável preditiva, foi utilizado a ferramenta “map” do Pandas, para atribuição de um código para cada resultado, sendo time de casa vencedor igual a 0, time visitante 1, e empate 2.

Tabela 3: Atribuição dos valores de saída

Resultado	Código
Time da casa vencedor	0
Time visitante vencedor	1
Empate	2

Fonte: Base de dados

A fim de evitar ponderações arbitrárias, para o nome dos times, foi utilizada outra ferramenta do Pandas chamada “get dummies”. Com ela, as vitórias de cada equipe como anfitriã, ou visitante, foram transformadas em colunas binárias para indicar a existência do evento. Para não haver problema com o nome das colunas geradas na hora do treinamento

e teste do modelo, uma função foi criada para tratar os caracteres e transformar em minúsculo, e o espaços por “underscore”.

Foi utilizado a biblioteca “statsmodels” do Python, para identificar possível colinearidade entre as variáveis utilizando o "variation inflation factor", que em alguns casos mostrou um alto valor de correlação entre si, principalmente as que tratam de valores até o intervalo, ou apenas do segundo tempo. Dessa forma, tais variáveis foram desconsideradas no treinamento do modelo.

Como será demonstrado, a amostra continha um número desbalanceado de resultados com o número de vitórias do time da casa muito superior quando comparado às vitórias do time visitante, ou aos empates. De acordo com Krawczyk (2016), isso representa uma dificuldade para algoritmos de machine learning atingirem um ajuste ideal, uma vez que esses tenderiam à categoria majoritária. Com isso, foi adotado o conceito de “undersampling” abordado por Fernandez et al. (2018), utilizando o pacote “resample”, da biblioteca sklearn para aleatoriamente selecionar resultados da categoria majoritária a um número que se igualasse a outras.

Modelos de Regressão Logística Multinominais [MRLM]

Segundo explicam Favero e Belfiore (2017), os MRLM partem do pressuposto de que a variável dependente que representa o fenômeno estudado é qualitativa e policotômica, assumindo mais de dois resultados possíveis. Assim, em uma circunstância com três possíveis classificações de resultado, estabelece-se um grupo de referência e dois outros cenários em relação ao conjunto escolhido como referência.

De acordo com os autores, supondo que tenhamos uma categoria, onde, e a escolhemos como a categoria de referência. Dessa forma, dois logitos, ou dois vetores de variáveis explicativas serão definidos junto aos coeficientes estimados a partir da equação geral.

$$P_{i_0} = \frac{1}{1 + e^{\left(\alpha_1 + \beta_{11} X_{i1} + \beta_{21} X_{i2} + \ldots + \beta_{kp} X_{ip}\right)} + e^{\left(\alpha_2 + \beta_{12} X_{i1} + \beta_{22} X_{i2} + \ldots + \beta_{kp} X_{ip}\right)}} \quad (1)$$

$$P_{i_1} = \frac{e^{\left(\alpha_1 + \beta_{11} X_{i1} + \beta_{21} X_{i2} + \ldots + \beta_{kp} X_{ip}\right)}}{1 + e^{\left(\alpha_1 + \beta_{11} X_{i1} + \beta_{21} X_{i2} + \ldots + \beta_{kp} X_{ip}\right)} + e^{\left(\alpha_2 + \beta_{12} X_{i1} + \beta_{22} X_{i2} + \ldots + \beta_{kp} X_{ip}\right)}} \quad (2)$$

$$P_{i_2} = \frac{e^{\left(\alpha_2 + \beta_{12} X_{i1} + \beta_{22} X_{i2} + \ldots + \beta_{kp} X_{ip}\right)}}{1 + e^{\left(\alpha_1 + \beta_{11} X_{i1} + \beta_{21} X_{i2} + \ldots + \beta_{kp} X_{ip}\right)} + e^{\left(\alpha_2 + \beta_{12} X_{i1} + \beta_{22} X_{i2} + \ldots + \beta_{kp} X_{ip}\right)}} \quad (3)$$

Onde:

- P_i é a probabilidade ocorrência do evento para cada observação i.
- X_{ij} são as variáveis independentes para a observação i.
- β_{kj} são os coeficientes associados às variáveis independentes X_{ij} para a classe k.

de forma que cada categoria da variável dependente é representada por k, onde k = 0,1,2, 3, ..., $K - 1$, sendo K o número de respostas da variável categorica. E cada observação da amostra é ilustrada por i, onde i = 1,2, 3, ..., n, em que n é o tamanho da amostra. O logito Z

é calculado para os valores de cada uma das variáveis explicativas, sejam essas métricas ou "dummies", representada por X_j , onde j = 1,2, 3, ..., k. Com isso, a equação engloba as categorias da variável dependente, além das variáveis α e β_j, que são estimadas levando-se em conta a máxima verossimilhança entre os termos.

Verossimilhança, ou como conhecida no inglês e representada na maioria das funções construídas no Python por "likelihood", é um conceito estatístico utilizado para calcular a razoabilidade de um conjunto de parâmetros dado um conjunto de dados, além de uma ferramenta paramétrica comum para realizar inferências quanto significância estatística, intervalos de confiança, avaliação do modelo, e predições (Millar R.B. 2011). Há também outras estatísticas utilizadas para avaliar o ajuste de um modelo de logística multinominal como pseudo R^2 de McFadden e o teste chi-quadrado que que pode ser utilizado para comparar dois modelos, graus de liberdade para definir a complexidade alcançada pelo conjunto de variáveis descritivas e logitos (Favero e Belfiore, 2017).

Arvores de classificação – "Random Forest"

A arvore de decisão, segundo Kamber et al. (2011), são modelos representados de forma gráfica com nós e ramos. Com um nó raiz definindo o topo da estrutura dá-se início através dos ramos a uma sequência de nós subsequentes que representam testes decisão de uma variável independente. Os nós folhas, são os valores de predição para as variáveis dependentes ou a distribuição de probabilidades. Wilkson (2004) divide as arvores em dois tipos de decisão, as de regressão que tem a variável dependente com valores numéricos, e as de classificação para valores categóricos.

De acordo com Loh (2011), ao analisar os resultados de uma arvore de classificação, deve-se forcar na previsão da classe correspondente a uma região, principalmente quando essa é uma área de nó terminal, e na proporção das observações em treinamento que se encontram nessa região. Uma forma de medir as características dos nós ao se construir uma árvore de decisão é a utilização do índice de Gini. Segundo Breiman et al. (1984), para avaliar a qualidade de uma divisão, ou seja, a quantidade de observações de uma das possibilidades, essa representada pela pureza do nó. O índice de pureza de um nó é dado pela dominância de uma classe sobre ele, sendo isso expresso por valores entre 0 e 1.

.

$$\boldsymbol{Gini(p) = 1 - \sum_{i=1}^{K} p_i^2} \quad (4)$$

Os modelos "ensemble" utilizados no contexto de machine learning, como o Random Forest proposto inicialmente por Breiman (2001), são modelos que utilizam de algoritmos de "begging", que combinam múltiplas arvores de decisão durante o treinamento. Cada árvore usa um subconjunto de dados e características. Amostras aleatórias são retiradas da base de dados original para criar outros subconjuntos de treinamento. O resultado é a moda das classes disponíveis nos casos de classificação. Isso significa que é a classe mais comum entre as observações da região localizada. Esse algoritmo é conhecido por sua robustez e bom desempenho sem a necessidade de muitos ajustes de hiper parâmetros.

No Random Forest os subconjuntos de dados e características são definidos via Bootstrap, o que incorpora aleatoriedade no crescimento das árvores, uma vez que atribui de

forma aleatória subconjuntos do conjunto inicial de preditores. Como resultado, quando agregadas, essas múltiplas árvores construídas a partir de combinações aleatórias, reduzem a variância, e, portanto, melhoram o desempenho preditivo.

Em busca de aumentar o poder preditivo do modelo, quanto aos ajustes dos hiper parâmetros, foi sugerido por Boehmke e Greenwell (2019) que no algoritmo de Random Forest hiper parâmetros os ajustados sejam: número de árvores(“trees”), número de variáveis consideradas de forma aleatória a cada separação (“mtry”), e o número mínimo de pontos no nó (“min_n”).

De forma complementar, foram utilizados algoritmos de Gradient Boosting desenvolvido por Chen e Guestrin (2016) que, diferente do Random Forest, que busca arvores profundas, estabelece arvores rasas de forma sequencial onde cada árvore aprende a partir dos erros da precedente, podendo essas ser um poderoso conjunto. Esse conjunto, quando ajustado de forma adequada, pode produzir resultados melhores do que outros algoritmos. Dentre esses, o Extreme Gradient Boosting (XGBoost), que tem como hiper parâmetros para ajuste: número de árvores(“trees”), número de variáveis consideradas de forma aleatória a cada separação (“mtry”), número mínimo de pontos no nó (“min_n”), profundidade de cada árvore (“tree_depth”), taxa de aprendizado (“learn_rate”), redução de perda mínima(“loss_reduction”) e o tamanho da amostra(“sample_size”).

Nos algoritmos de classificação, como evidenciado por Guido e Muller (2016), o desempenho pode ser mensurado por matrizes de confusão. Essas tabelas evidenciam, em termos práticos, a assertividade do modelo, trazendo a relação entre o que foi previsto e o que foi observado na amostra. A estrutura da matriz é composta por 4 componentes, sendo:

VP = Verdadeiros Positivos, resultados verdadeiros classificados de forma correta
VN = Verdadeiros Negativos, resultados falsos classificados de forma correta
FP = Falso Positivos, resultados falsos classificados como verdadeiros
FN = Falso Negativo, resultados verdadeiros classificados como falsos

Os componentes usados em conjunto podem representar a qualidade do modelo de várias perspectivas. Por exemplo, a acurácia é uma dessas formas, calculando a proporção de respostas corretamente preditas pelo modelo.

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (5)$$

Outras métricas utilizadas para aferir a qualidade de um modelo Random Forest são Precisão, que mede a acurácia apenas de respostas positivas, o Recall que traz a habilidade do modelo em identificar todos os componentes relevantes, e o F1 score que combina ambas métricas anteriores.

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (7)$$

$$\text{F1 Score} = 2\,x\,\frac{\text{Precisão}\,x\,\text{Recall}}{\text{Precisão} + \text{Recall}} (8)$$

Resultados e Discussão

Estatística descritiva dos dados

O número de partidas disputadas em cada ano analisado se mostra constante durante todos os anos no campeonato brasileiro de futebol, “Série A”, entre os anos de 2013, até meados de setembro 2024. Exceto no ano de 2016 por conta do cancelamento da partida final de um dos clubes em decorrência de um acidente aéreo.
Também, em 2020 onde por conta da pandemia, a temporada foi interrompida, e retomada em 2021. O que também explica o maior número de jogos realizados nesse ano, em decorrência do término do campeonato do ano anterior.

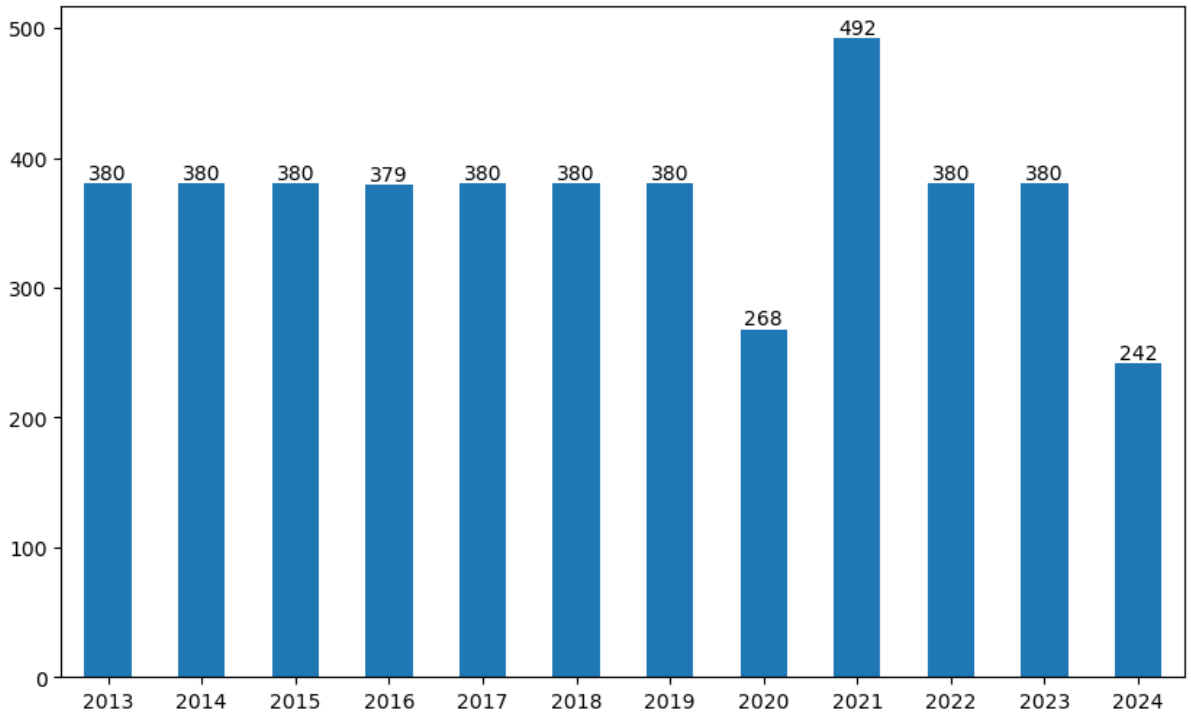

Figura 2. Distribuição das partidas realizadas durante os anos.
Fonte: Dados coletados

Considerando a concatenação das partidas realizadas no campeonato brasileiro de futebol, 'Série A' A', entre os anos de 2013, até meados de setembro 2024, e após a utilização das técnicas de limpeza do “dataset” inicialmente foram selecionadas 4392 partidas, onde 2129 (48,47%) tinham como resultado a vitória do time da casa, 1085 (24,70%) com vitórias do time visitante, e 1178 (26,82%) empates.

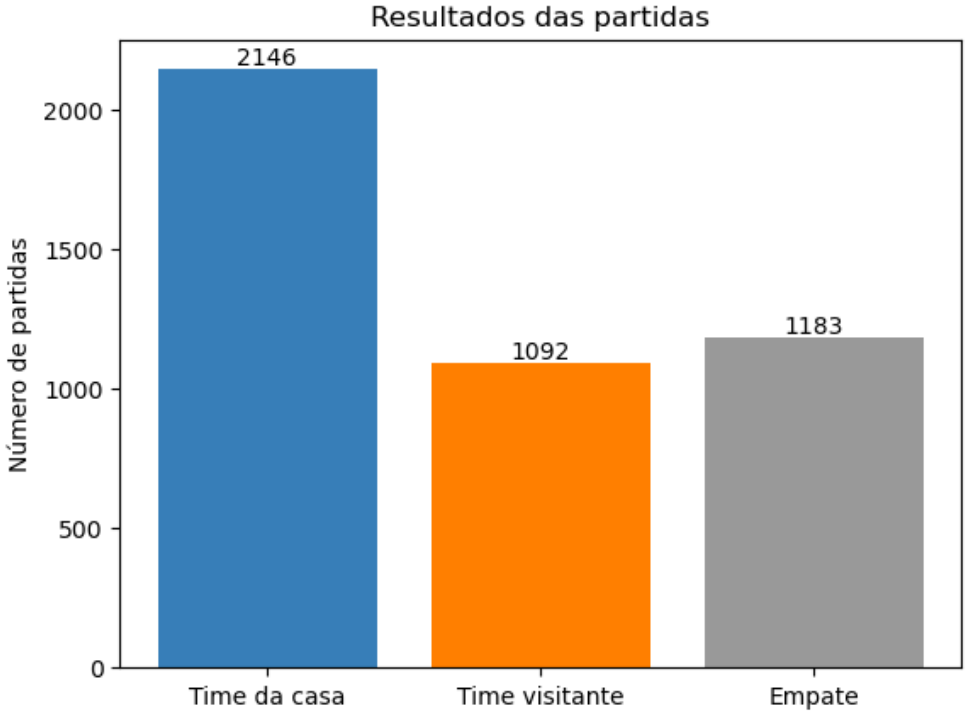

Figura 1. Distribuição dos resultados das partidas analisadas.
Fonte: Dados coletados

A partir desses dados, é possível afirmar que no acumulado das temporadas de 2013 até setembro de 2024, o resultado em que o time da casa sai vitorioso ocorre com maior frequência do que a vitória do time visitante, ou o empate. Fato esse que indica a necessidade de balanceamento das categorias na variável dependente para modelos mais sensíveis ao número de observações de cada um dos possíveis resultados.

As variáveis explicativas disponibilizadas no conjunto de dados utilizados eram sessenta e seis. A análise inicial buscou identificar no “dataset” utilizado, dentre as features disponíveis, que fariam sentido para aplicação do modelo. Com isso, variáveis relacionadas a previsões de apostas esportivas foram eliminadas, uma vez que não faziam parte da probabilidade de eventos da partida, nem dos eventos em si, como probabilidade de gols, gols, entre outros. Também foram eliminadas variáveis com número de observações inconsistentes, e não representativas como o nome do estádio em que o jogo foi realizado, árbitro, “timestamp”.

Já no sentido de modelagem dos dados para aplicação do modelo, foram eliminadas variáveis identificadas a partir da verificação do Fator de Inflação da Variância (“variantion inflation factor”), que é um índice que indica o grau de colinearidade entre variáveis independentes através do cálculo da variância do coeficiente de regressão.

Com isso, as variáveis utilizadas nos primeiros testes do modelo foram:

Tabela 5: Variáveis utilizadas na primeira modelagem

Variável	Descrição
home_ppg	Pontos por jogo em casa
away_ppg	Pontos por jogo como visitante
home_team_goal_count	Contagem de gols do time da casa

away_team_goal_count	Contagem de gols do time visitante
home_team_corner_count	Contagem de escanteios do time da casa
away_team_corner_count	Contagem de escanteios do time visitante
home_team_yellow_cards	Contagem de cartões amarelos do time da casa
home_team_red_cards	Contagem de cartões vermelho do time da casa
away_team_yellow_cards	Contagem de cartões amarelos do time visitante
away_team_red_cards	Contagem de cartões vermelho do time visitante
home_team_shots	Chutes time da casa
away_team_shots	Chutes time visitante
home_team_shots_on_target	Chutes no gol time da casa
away_team_shots_on_target	Chutes no gol time visitante
home_team_fouls	Faltas time da casa
away_team_fouls	Faltas time visitante
home_team_possession	% de posse de bola time da casa
away_team_possession	% de posse de bola time visitante
home_team_pre_match_xg	Chance de gol do time da casa (pré-partida)
away_team_pre_match_xg	Chance de gol do time visitante (pré-partida)
average_goals_per_match_pre_match	Média de gols do time da casa (pré-partida)
average_corners_per_match_pre_match	Média de gols do time visitante (pré-partida)
average_cards_per_match_pre_match	Média de cartões pré-partida

Fonte: Plataforma provedora dos dados

Onde as variáveis “home_ppg” e “away_ppg” são definidos pelas equações:

$$\frac{\sum pontos\ em\ casa}{\sum partidas} (5)$$

$$\frac{\sum pontos\ fora\ de\ casa}{\sum partidas} (6)$$

Já “home_team_pre_match_xg” e “away_team_pre_match_xg” são calculados por uma fórmula da própria provedora dos dados, que segundo essa, considera no cálculo da chance de um time marcar gols a assertividade (no alvo/fora do alvo), a frequência de chutes ao gol (número de chutes), ataques perigosos, a pressão ofensiva (porcentagem de posse e profundidade da posse) para resumir as estatísticas de gol esperados (Footystats, 2024).

Modelo Logístico Multinominal

Após a preparação na base de dados para treinamento, esse modelo em específico, necessitou que antes do treinamento fosse rebalanceado a categoria “Vitoria do time da casa”, uma vez que essa apresentava um valor muito superior as outras categorias.

Com o intuito de interferir o menos possível na amostra, as observações com resultado de vitória do time da casa foram aleatoriamente selecionadas, reduzindo então a categoria

com maior frequência a um número de observações próximo ao das outras. Os números utilizados como resultado da categoria de interesse foram os seguintes:

TABELA 6: Amostras após o balanceamento das categorias

Resultado	Total	Balanceado
0	2146	1178
1	1092	1085
2	1183	1178

Fonte: Algoritmo de modelagem dos dados

Os resultados, mesmo com o balanceamento, mostraram um desempenho fraco do modelo na predição de resultados, como é possível ver a seguir:

TABELA 7: Resultados da regressão logística multinominal

Métrica	Valor
No. Observations	3441
Df Residuals	3253
Df Model	186
Pseudo R-squared	0.08877
Log-Likelihood	-3442.4
LL-Null	-3777.8
LLR p-value	5,64E-53
Convergence	False

Fonte: Algoritmo de regressão logística multinominal

Através do sumario do resultado, podemos ver que métricas explicativas do modelo apresentam um baixo poder preditivo. Com uma máxima verossimilhança negativa, e como é o desejado para um modelo com um bom ajuste, longe de 0. Além disso, o Pseudo R^2 aponta um algoritmo capaz de explicar apenas 8.88% da variância dos dados, e a convergência como falsa indicando que o algoritmo, seja por “overfitting” ou colinearidade, não encontrou uma solução ótima.

Com o objetivo de atingir um melhor desempenho para o modelo, através do próprio sumario do resultado foi possível selecionar apenas as variáveis que tinham significância estatística. Dessa forma, as variáveis e seus coeficientes foram:

TABELA 8: Variáveis do conjunto que possuem significância estatística

Variável	Valor do p
home_team_goal_count	0.000
away_team_goal_count	0.000
away_team_red_cards	0.002
away_team_shots_on_target	0.008
home_team_shots_on_target	0.014
home_team_shots	0.025
away_team_fouls	0.037

Fonte: Algoritmo de regressão logística multinominal

Mesmo com uma ligeira melhora nas métricas, o procedimento de “step-wise” não foi o suficiente para garantir a efetividade do modelo. Com resultado semelhante a modelagem com todas as variáveis disponíveis, apenas confirmando que a não convergência do modelo se dava por “overfitting”, uma vez eliminadas as features não significativas, o algoritmo atingiu ao melhor ajuste possível.

TABELA 9: Resultado do modelo utilizando apenas variáveis estatisticamente significativas

Métrica	Valor
No. Observations	3441
Df Residuals	3285
Df Model	154
Pseudo R-squared	0.08594
Log-Likelihood	-3453.1
LL-Null	-3777.8
LLR p-value	5,14E-59
Convergence	True

Fonte: Algoritmo de regressão logística multinominal

Modelo Random Forest

Pela robustez apresentada por esse modelo de classificação, não foi necessário o rebalanceamento das categorias para o treinamento do algoritmo, e nenhum outro procedimento além dos já implementados foi necessário.

No Python, além das bibliotecas para cálculos complexos como o de uma arvore de classificação, ele também oferece ferramentas de busca dos melhores parâmetros no caso de modelos que oferecem a personalização desses. Com isso, é possível passar uma lista de possíveis valores para que o computador possa iterar entre eles e encontre a melhor combinação preditiva.

As estatísticas de resultado do modelo apresentam uma acurácia de 59%, ou seja, o algoritmo foi capaz de prever o resultado de forma precisa em pouco mais da metade dos resultados observados. Os hiperparâmetros utilizados e as estatísticas complementares do modelo foram:

TABELA 10: Conjunto de hiperparâmetros ótimos para o modelo

Parâmetro	Valor
n_estimators	300
min_samples_split	5
min_samples_leaf	1
max_features	sqrt
max_depth	20
bootstrap	False

Fonte: Algoritmo de classificação Random Forest

TABELA 11: Estatísticas descritivas do modelo Random Forest

Classe	Precisão	Recall	F1-Score
0	0.57	0.92	0.71
1	0.68	0.28	0.40
2	0.64	0.26	0.37
Accuracy			0.59
Macro Avg	0.63	0.49	0.49
Weighted Avg	0.62	0.59	0.54

Fonte: Algoritmo de classificação Random Forest

Com a técnica do Extreme Gradient Boost, foi possível melhorar ligeiramente em 3% o poder preditivo do modelo, apresentando uma acurácia de 62%. Os hiperparâmetros, assim como os resultados foram:

TABELA 12: Conjunto de hiperparâmetros ótimos para o modelo com XGB

Parâmetro	Valor
colsample_bytree	0.8
learning_rate	0.01
max_depth	6
n_estimators	200
reg_alpha	0
reg_lambda	1
subsample	0.9

Fonte: Algoritmo de classificação Random Forest com XGB

TABELA 13: Estatísticas descritivas do modelo com XGB

Classe	Precisão	Recall	F1-Score
0	0.58	0.96	0.72
1	0.70	0.27	0.38
2	0.86	0.30	0.45
Accuracy			0.62
Macro Avg	0.71	0.51	0.52
Weighted Avg	0.68	0.62	0.57

Fonte: Algoritmo de classificação Random Forest com XGB

Além da acurácia que aponta a eficiência geral do algoritmo, há outras estatísticas que trazem pontos interessantes sobre o resultado. Podemos ver que na precisão, houve uma melhora significativa na taxa de acerto do modelo com XGB quando comparado com o primeiro modelo Random Forest para as classes 1 e 2, impactando positivamente a média geral e a ponderada, enquanto a classe 0 se manteve estável. De forma adicional, ao examinarmos a sensibilidade, também conhecida como Recall, a categoria de interesse "0" mostrou-se superior às demais, atingindo um valor quase máximo, para os grupos alternativos, o desempenho foi bastante inferior. O algoritmo mostrou elevada sensibilidade, contudo uma precisão mediana para a categoria que abrangia o maior número de

observações. No entanto, em relação às classes 1 e 2, o comportamento foi o contrário, com um recall baixo enquanto apresentava uma precisão alta.

Ao observarmos a média harmônica entre precisão e sensibilidade, representada pelo F1-Score, com resultados relativamente dispersos entre as classes o modelo acaba se balanceando e apresentando um poder mediano no agregado dessas medidas para o algoritmo quando treinado com essa base de dados.

Conclusões

Ao implementar algoritmos com variadas abordagens, tais como, de regressão utilizando logístico multinominal e de classificação com Random Forest, revelou-se que os modelos de classificação alcançam melhores resultados, quando empregados em bases de dados públicas disponíveis na internet, em particular, plataformas que disponibilizam dados como insumo para apostas, uma vez que estas oferecem informações parecidas na maioria das situações. Por usar a técnica de begging, ele produz várias subamostras para buscar um resultado ideal. Isso o torna menos sensível à colinearidade das variáveis. Além disso, lida bem com amostras desbalanceadas, como no nosso caso.

Contudo, acredita-se que se realizado o cruzamento dessas informações disponíveis com outras também relevantes ao resultado, mas que se apresentam “extra” campo como o clima no dia do jogo por interferir na qualidade do gramado, ou até mesmo, o orçamento anual dos times visto o potencial de influência, o modelo atingiria uma acurácia ainda maior.

Conciliar essas informações de forma consistente é uma das primeiras dificuldades encontradas no trabalho. Como adendo, fica a limitação da plataforma de disponibilizar dados apenas a partir da temporada de 2013. A consideração do ajuste de um algoritmo às variáveis disponíveis é importante. No entanto, a imprevisibilidade intrínseca ao futebol, como em qualquer esporte, traz uma combinação de fatores que nem sempre são matematicamente explicáveis.

Se houver interesse em explorar mais a chance de vitória dos times nesta competição, sugere-se que estudos futuros usem os dados de maneira mais intensiva, aplicando diferentes modelos e objetivos. Um exemplo seria a chance de uma equipe marcar certa quantidade de gols.

Agradecimento

Referências

Afonso, M. S., Barros, S. S., Koth, A. P., Rodrigues, V. L., Neves, F. B., & Lourenção, L. G. (2020). Sports physiotherapy in program of prevention of injury in professional football. Research, Society and Development.

Breiman, L. 2001. Random forests. Machine learning 45(1): 5-32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. 1984. Classification and regression trees. 1ed. Chapman and Hall. Belmont, California, Estados Unidos da América.

Carpita, M.; Sandri, M.; Simonetto, A.; Zuccolotto, P. Discovering the Drivers of Football Match Outcomes with Data Mining, Quality Technology & Quantitative Management, Vol. 12:4, p. 561-577, 2015.

Confederação Brasileira de Futebol [CBF]. 2024. Regulamento Específico da Competição: Brasileiro Série A 2024. Disponível em: < https://conteudo.cbf.com.br/cdn/202403/20240317094857_834.pdf >. Acesso em: 01 outubro 2024.

Fávero, L. P.; Belfiore, P. 2017. Manual de análise de dados. 1ed. Elsevier, Rio de Janeiro, RJ, Brasil.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., Herrera, F. 2018. Learning from imbalanced data sets. Springer, Berlin, Alemanha.

Footystats. 2024. Match CSV – 64 Data Columns. Disponível em: < https://footystats.org/download-stats-csv#whats_included>. Acesso em: 16 setembro 2024.

Gifford, M., Bayrak, T. 2020. What Makes a Winner? Analyzing Team Statistics to Predict Wins in the NFL. Americas Conference on Information Systems, AMCIS, pp. 10–14.

Gil, A. C. 2022. Como elaborar projetos de pesquisa. 7ed. Atlas, Barueri, São Paulo, Brasil.

Guido, S., Muller A.C. 2016. Introduction to Machine Learning with Python: A Guide for Data Scientists. 1ed. O'Reilly Media. Sebastopol, California, Estados Unidos da América.

Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier. Waltham, Massachusetts, Estados Unidos da América.

Herold, M., Goes, F., Nopp, S., Bauer, P., Thompson, C., & Meyer, T. (2019). Machine learning in men's professional football: Current applications and future directions for improving attacking play. *International Journal of Sports Science & Coaching*, *14*(6), 798-817.

Malhotra, N. K. 2012. Pesquisa de marketing: uma orientação aplicada. 6ed. Artimed, Porto Alegre, Rio Grande do Sul, Brasil.

MILLER, R. B. Maximum likelihood estimation and inference: with examples in R, SAS, and ADMB. 1ed. John Wiley & Sons INC International Concepts, Nova York, Nova York, Estados Unidos da América.

Krawczyk, Bartosz. 2016. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence 5: 221-232.

Sehnem, R., & Frozza, R. (2019). Análise de variáveis em partidas de futebol para previsão de resultados. Anais do Salão de Ensino e de Extensão, 217.

Wilkinson, L. (2004). Classification and regression trees. Systat, 11, 35-56.

Apêndice ou Anexo (opcional)

Os apêndices são textos e/ou documentos que foram elaborados pelos autores e que são importantes para complementar a argumentação do trabalho. Anexos são textos ou documentos que ilustram o trabalho, mas que não foram elaborados pelos autores. Apêndices deverão seguir as mesmas normas de formatação do restante do texto, inclusive para as figuras e tabelas.
O TCC deverá conter no máximo 30 páginas, incluindo o(s) Apêndice(s) e/ou Anexo(s).
Atenção: antes de enviar o arquivo para o Sistema de TCCs, remova todas as instruções originais que estão abaixo do conteúdo dos tópicos.