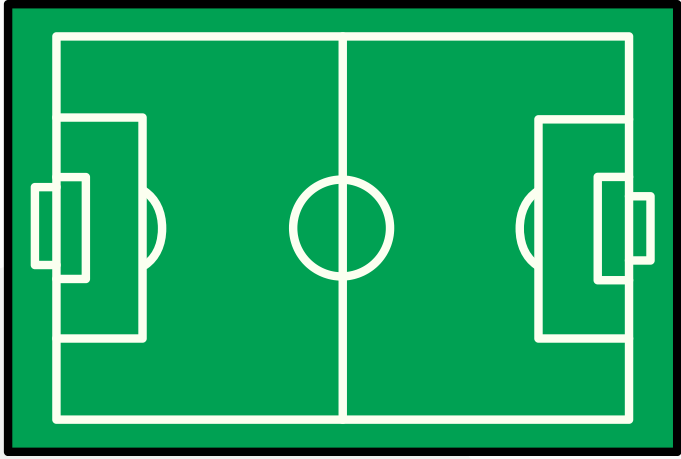
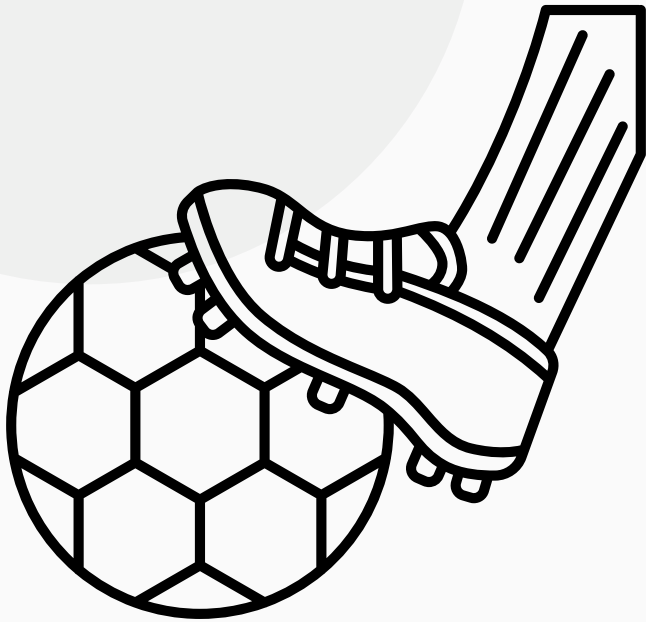
Projeção de resultados no futebol brasileiro com o uso de machine learning



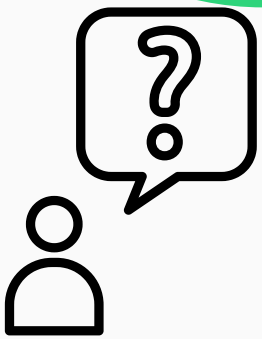
Projeto feito em cima da base de dados do transfermarkt, para ajudar na projeção de resultados nos confrontos do futebol brasileiro.

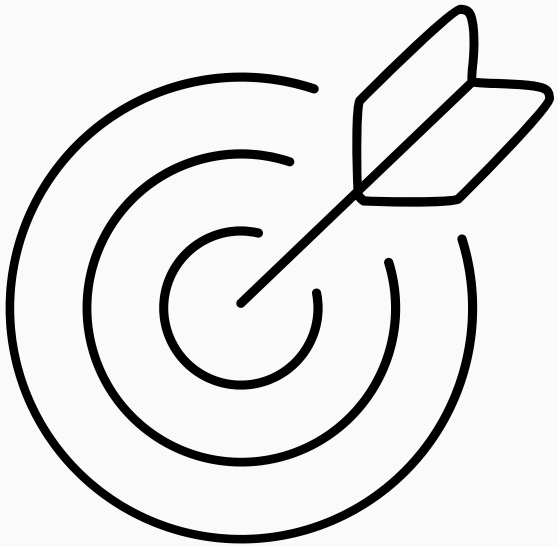


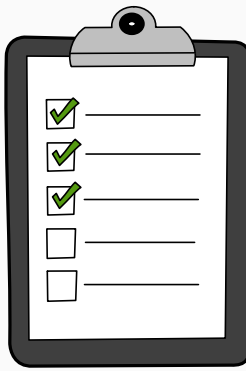
Equipe: Wesley Da Silva, Rodrigo Alves, Pedro Ryan

Introdução

- O futebol é um esporte globalmente popular, impulsionando uma indústria bilionária.
- Sua imprevisibilidade apresenta desafios, exigindo uma compreensão profunda para maximizar o potencial das equipes.
- A aplicação de técnicas de Machine Learning oferece uma alternativa para prever resultados, utilizando dados do Transfermarkt.
- Objetivo do estudo: prever resultados entre: Vitória do mandante, empate ou vitória visitante no campeonato brasileiro (Brasileirão Série "A") através de simulações de jogos.

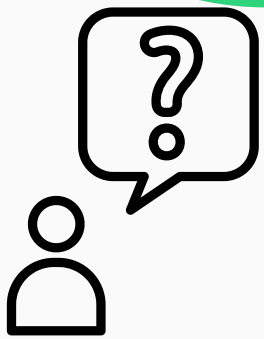


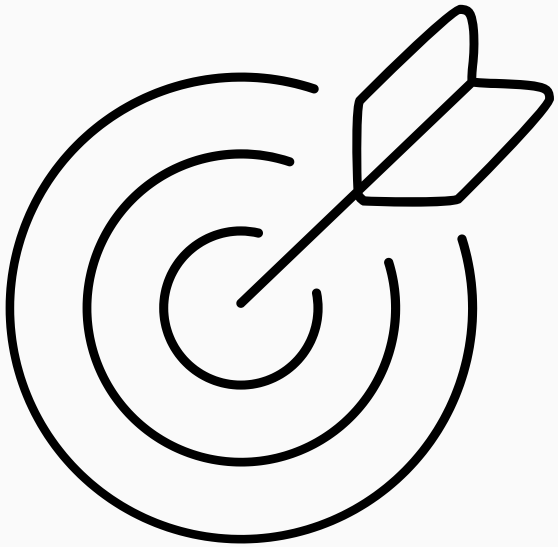


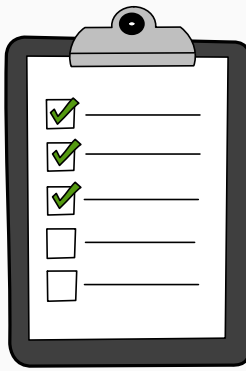


Abordagem

- Previsão de desempenho esportivo no futebol era uma tarefa realizada por poucos profissionais, mas com a evolução tecnológica, técnicas de machine learning têm sido implementadas para criar predições de resultados em jogos.
- O estudo aborda a predição de resultados no campeonato brasileiro de futebol (Brasileirão Série A)

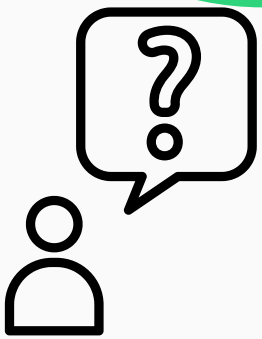


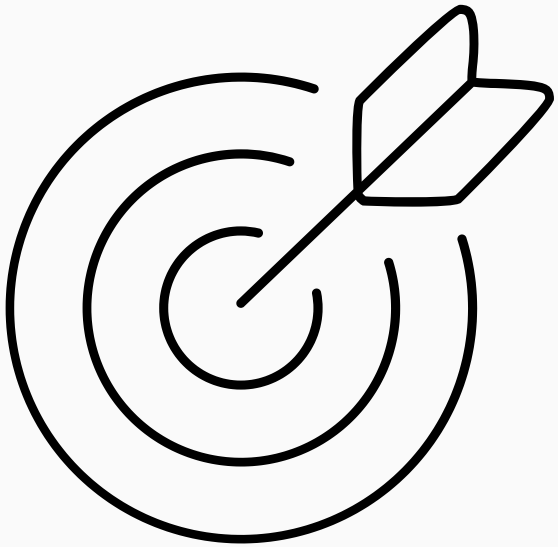


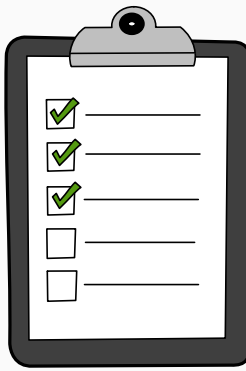


Métricas e técnicas

- Grid SearchCV: Técnica para otimizar hiperparâmetros de modelos de machine learning, realizando busca exaustiva e incluindo cross-validation.
- Random forest: Algoritmo de machine learning que cria múltiplas árvores de decisão a partir de subconjuntos aleatórios do conjunto de dados.
- XGBoost: Algoritmo de machine learning que melhora a precisão de modelos de previsão combinando várias árvores de decisão de forma eficiente.
- Matriz de confusão: Ferramenta para avaliar o desempenho de modelos de classificação em machine learning.
- Precision e recall: Métricas para avaliar o desempenho de modelos de classificação, especialmente em desequilíbrios de classes.
- RandomUnderSampler: Técnica de balanceamento de dados em problemas de machine learning.
- F1-score: Métrica de desempenho para avaliar a precisão de modelos de classificação.
- LightGBM: Algoritmo de machine learning baseado em árvores de decisão de gradiente.

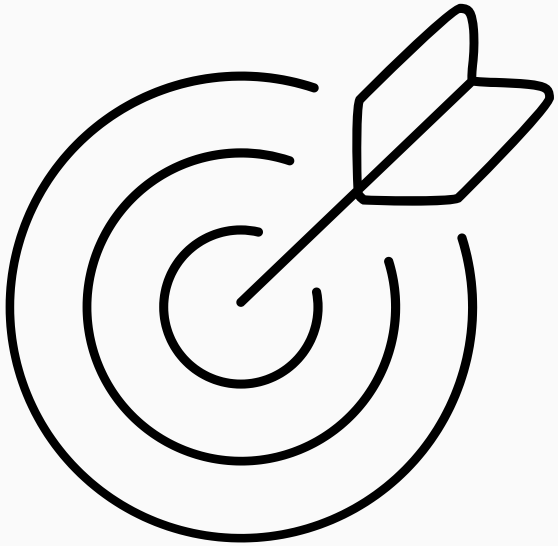


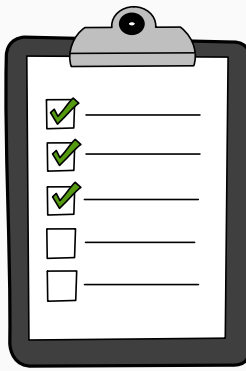




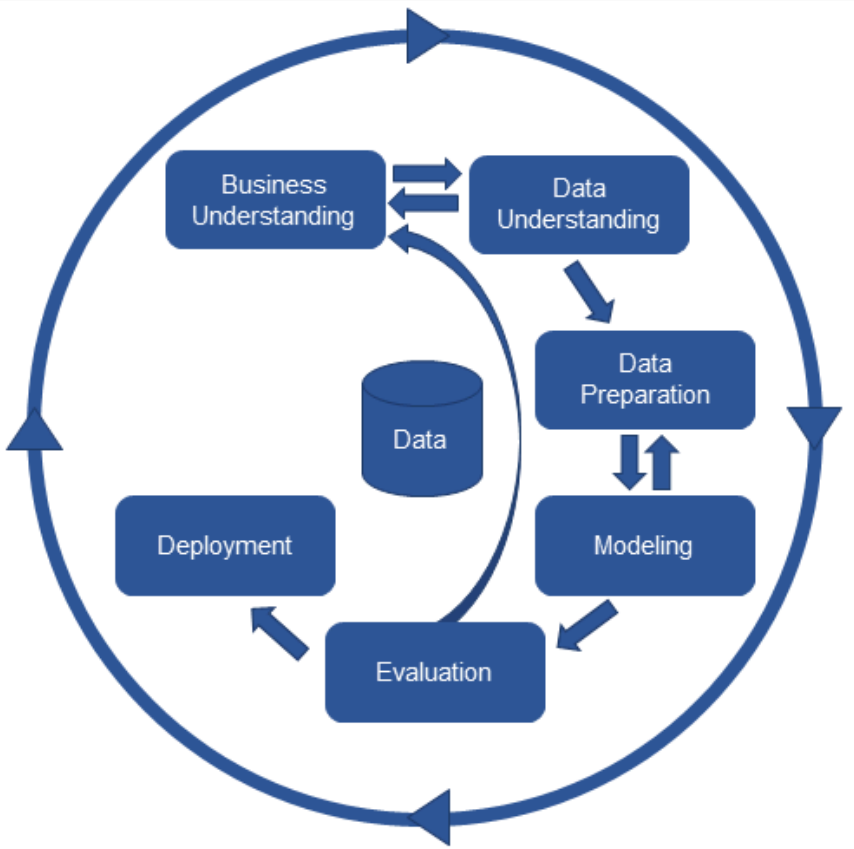
Trabalhos relacionados

- Michael Lewis's Book: Em 2003, o livro de Michael Lewis destacou a utilização da análise estatística no beisebol, demonstrando como uma abordagem baseada em dados pode levar a resultados surpreendentes.
- Inteligência Artificial no Futebol Inglês: Clubes como Chelsea e Burnley adotaram a inteligência artificial na análise de desempenho, utilizando métodos quantitativos para identificar padrões e avaliar o desempenho dos jogadores.
- Modelos Estatísticos na English Premier League: Um estudo propôs o uso de modelos estatísticos baseados em estatística Bayesiana para prever resultados na English Premier League, demonstrando a aplicação de métodos probabilísticos na análise de partidas de futebol.





Metodologia



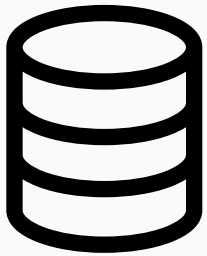
Metodologia CRISP-DM

Base de dados

- A base de dados foi disponibilizada pelo Transfermarkt e abrange o período de 2003 a 2023 da série A do campeonato brasileiro.

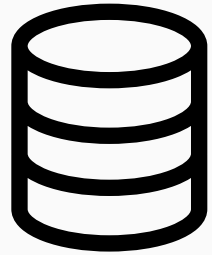
- Composta por 35 colunas e 8079 linhas, os dados são centrados nos confrontos do campeonato brasileiro.

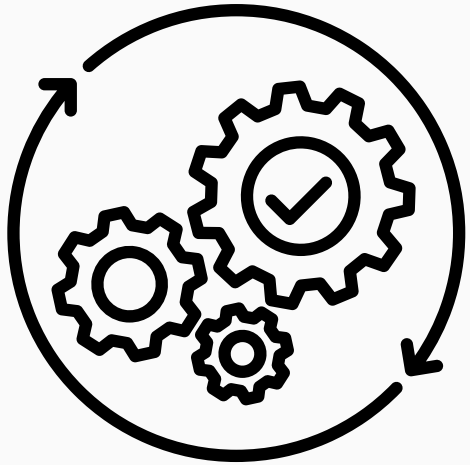
- A base cobre confrontos entre os times, disponibilizando informações como: como gols do time mandante, gols do time visitante, time visitante daquela determinada partida assim como o time mandante, etc...



Base de dados

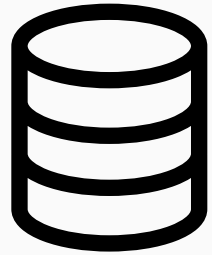
```
ano_campeonato                  8079
data                            8079
rodada                          8079
estadio                         8068
arbitro                         6369
publico                         6368
publico_max                     3817
time_mandante                   8079
time_visitante                  8079
tecnico_mandante                5926
tecnico_visitante               5926
colocacao_mandante              6369
colocacao_visitante             6369
valor_equipe_titular_mandante   5981
valor_equipe_titular_visitante  5981
idade_media_titular_mandante    5979
idade_media_titular_visitante   5979
gols_mandante                   8078
gols_visitante                  8078
gols_1_tempo_mandante           6359
gols_1_tempo_visitante          6359
escanteios_mandante             1788
escanteios_visitante            1788
faltas_mandante                 1788
faltas_visitante                1788
chutes_bola_parada_mandante     1788
chutes_bola_parada_visitante    1788
defesas_mandante                1788
defesas_visitante               1788
impedimentos_mandante           1788
impedimentos_visitante          1788
chutes_mandante                 1788
chutes_visitante                1788
chutes_fora_mandante            1788
chutes_fora_visitante           1788
dtype: int64
```

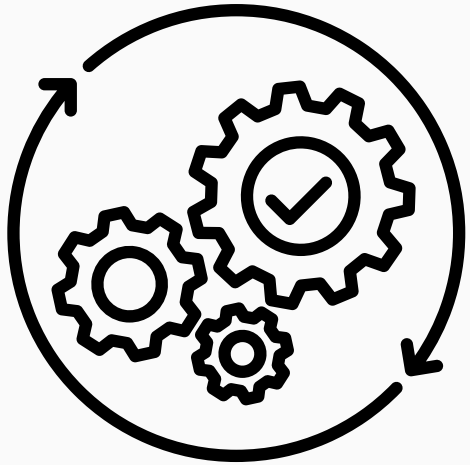




Pré-processamento

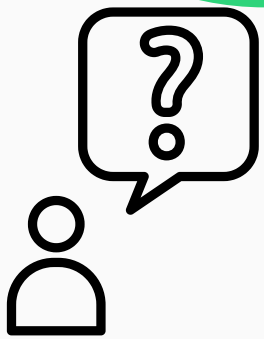
- Após uma raspagem inicial, foi necessário remover os dados anteriores a 2006 devido a mudanças no regulamento do Brasileirão.
- Colunas numéricas com 6% de dados faltantes foram preenchidas utilizando a média dos registros existentes.
- Colunas com menos de 50% de dados preenchidos foram removidas para evitar a perda de capacidade preditiva.
- Novas colunas foram geradas, incluindo média e saldo de gols acumulado dos times ao longo das rodadas.

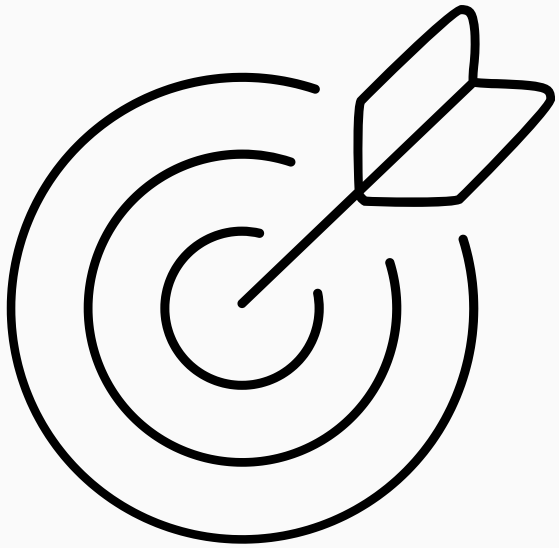


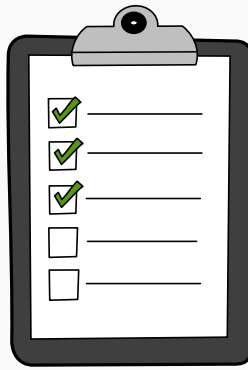


Construção

- Três modelos baseados em árvores foram testados para definir vitória do visitante, vitória do mandante ou empate.
- Técnicas como RandomUnderSampler para balanceamento da base e Grid SearchCV para otimização dos hiperparâmetros foram aplicadas.
- Os modelos foram treinados utilizando o conjunto de partidas ocorridas durante todos os anos do campeonato, permitindo a predição das partidas do ano posterior.
- Métricas como precisão, sensibilidade, F1-score e acurácia foram calculadas para avaliar o desempenho dos modelos.







Construção

```
# Selecionar colunas relevantes
features = ['colocacao_mandante', 'colocacao_visitante',
            'media_gols_mandante', 'media_gols_visitante',
            'partidas_jogadas_mandante', 'partidas_jogadas_visitante',
            'saldo_acumulado_mandante', 'saldo_acumulado_visitante']
```

Parâmetros e valores usados no Grid Search

Modelo	Parâmetros	Valores
RandomForest	n_estimators	[100, 200]
	max_depth	[10, 20, None]
	min_samples_split	[2, 5]
	min_samples_leaf	[1, 2]
	bootstrap	[True, False]
	criterion	[gini, entropy]
LightGBM	max_depth	[3, 6, 9]
	learning_rate	[0.1, 0.01, 0.001]
	n_estimators	[100, 200, 300]
Xgboost	max_depth	[3, 6, 9]
	learning_rate	[0.1, 0.01, 0.001]
	n_estimators	[100, 200, 300]
	objective	[multi:softmax]

Resultados

- O Grid Search foi empregado para a otimização de hiperparâmetros neste projeto, utilizando as configurações listadas na tabela mostrada anteriormente. Três modelos de machine learning foram avaliados após a otimização dos hiperparâmetros: RandomForest, LightGBM e XGBoost.

Resultados

- Foram executados três modelos de machine learning baseados em árvores de decisão: RandomForest, LightGBM e XGBoost. Todos os modelos foram executados utilizando o Random Under Sampler para balanceamento das classes alvo da previsão.

Avaliação dos modelos

- Foi analisado como os três modelos se saiam nas métricas de desempenho pré estabelecidas nas seções anteriores do estudo: precisão, sensibilidade, F1-score e acurácia com e sem a otimização dos hiperparâmetros utilizando o Grid SearchCV.

Modelo XGBoost

- **Resultados sem otimização**

Classe	Precisão	Sensibilidade	F1-Score
Empate	0.31	0.36	0.33
Vitória mandante	0.69	0.54	0.61
Vitória visitante	0.42	0.53	0.47

- **Resultados com otimização**

Classe	Precisão	Sensibilidade	F1-Score
Empate	0.35	0.37	0.36
Vitória mandante	0.72	0.58	0.64
Vitória visitante	0.45	0.60	0.52

Modelo LightGBM

- **Resultados sem otimização**

Classe	Precisão	Sensibilidade	F1-Score
Empate	0.32	0.36	0.34
Vitória mandante	0.70	0.57	0.63
Vitória visitante	0.44	0.56	0.49

- **Resultados com otimização**

Classe	Precisão	Sensibilidade	F1-Score
Empate	0.35	0.38	0.37
Vitória mandante	0.70	0.57	0.63
Vitória visitante	0.46	0.60	0.52

Modelo Random Forest

- **Resultados sem otimização**

Classe	Precisão	Sensibilidade	F1-Score
Empate	0.30	0.34	0.32
Vitória mandante	0.70	0.56	0.62
Vitória visitante	0.44	0.56	0.50

- **Resultados com otimização**

Classe	Precisão	Sensibilidade	F1-Score
Empate	0.33	0.36	0.35
Vitória mandante	0.71	0.59	0.64
Vitória visitante	0.47	0.59	0.52

Acurácia dos modelos

- **Resultados sem otimização**

Modelo	Acurácia
XGBoost	0.49
LightGBM	0.50
Random Forest	0.50

- **Resultados com otimização**

Modelo	Acurácia
XGBoost	0.53
LightGBM	0.52
Random Forest	0.52

Discussão dos resultados

- Os resultados demonstram melhorias em quase todas as métricas avaliadas após a otimização dos modelos.
- O XGBoost obteve a maior acurácia 0.53% e se destacou na classe "Vitória Mandante" com precisão de 0.72 e F1-Score de 0.64.
- LightGBM e Random Forest apresentaram resultados próximos, com o Random Forest mostrando ligeiramente melhor desempenho na sensibilidade e F1-Score para a classe "Vitória Mandante".

Random Forest

Classe	Precisão	Sensibilidade	F1-Score
Empate	0.33	0.36	0.35
Vitória mandante	0.71	0.59	0.64
Vitória visitante	0.47	0.59	0.52

LGBM

Classe	Precisão	Sensibilidade	F1-Score
Empate	0.35	0.38	0.37
Vitória mandante	0.70	0.57	0.63
Vitória visitante	0.46	0.60	0.52

XGB

Classe	Precisão	Sensibilidade	F1-Score
Empate	0.33	0.36	0.35
Vitória mandante	0.71	0.59	0.64
Vitória visitante	0.47	0.59	0.52

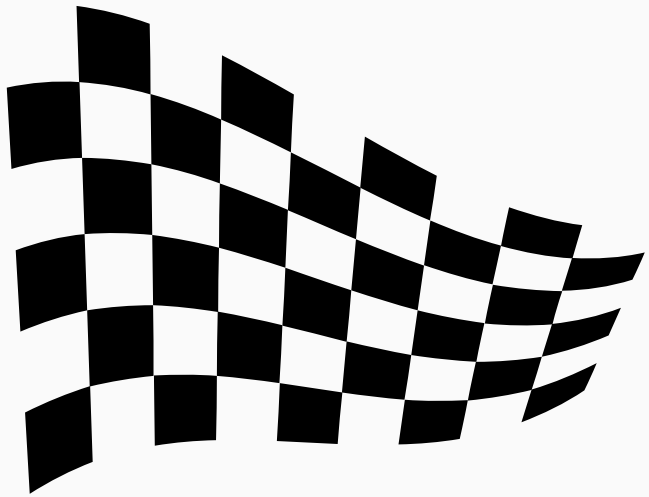
Escolha do modelo

Com base nas análises:

- Optou-se por aprofundar no modelo XGBoost após a otimização.
- Antes da otimização, o XGBoost tinha a pior performance geral, mas após a otimização, apresentou melhorias significativas, especialmente em métricas como precisão e F1-Score para as classes de Vitória Mandante e Vitória Visitante.
- Embora o LightGBM também tenha apresentado resultados próximos, o XGBoost mostrou um balanceamento melhor entre as classes e uma diferença menor entre elas, levando à sua escolha final devido à leve melhoria geral de desempenho.

Conclusão

O estudo apresentou resultados promissores ao utilizar técnicas de machine learning para prever resultados de partidas de futebol no contexto do Campeonato Brasileiro. Ao analisar dados de confrontos entre 2006 e 2023, foi possível observar que esses modelos podem ser ferramentas úteis para entender cenários e obter previsões mais realistas em partidas de futebol.



Passos Futuros

- Simulação Completa do Campeonato: Os modelos podem ser aprimorados para simular não apenas resultados de partidas individuais, mas também a tabela completa do campeonato. Isso permitiria planejamentos mais detalhados para os clubes, envolvendo aspectos financeiros e esportivos desde o início da temporada.

- Melhoria na Qualidade dos Dados: Investir em dados mais completos e atualizados pode melhorar significativamente o desempenho dos modelos. Informações detalhadas sobre táticas e estratégias dos jogos podem proporcionar uma base mais robusta para as previsões.

- Decomposição das Previsões: Dividir as previsões em sub-decisões binárias pode simplificar o processo e potencialmente melhorar o desempenho dos modelos.

- Prototipação Prática: Implementar uma interface prática que permita a simulação de resultados em tempo real, com dados atualizados, pode validar ainda mais a eficácia dos modelos propostos e transformar a teoria em uma ferramenta prática para os clubes de futebol.

Considerações finais

Este estudo destaca o potencial do uso de machine learning no futebol, não apenas para prever resultados, mas também para fornecer estatísticas e dados úteis aos profissionais do setor. A incorporação dessa tecnologia pode contribuir significativamente para uma compreensão mais lógica do jogo, auxiliando na tomada de decisões estratégicas por parte da gestão dos clubes e técnicos.

THE END...