UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

FELIPE VENTURA KUHNERT
PEDRO NOVAES POSSATO

MODELO PARA PREDIÇÃO DOS RESULTADOS DE PARTIDAS DE FUTEBOL

RIO DE JANEIRO
2023

FELIPE VENTURA KUHNERT
PEDRO NOVAES POSSATO

MODELO PARA PREDIÇÃO DOS RESULTADOS DE PARTIDAS DE FUTEBOL

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Profa. Giseli Rabello Lopes

RIO DE JANEIRO
2023

Kuhnert, Felipe Ventura
K96m Modelo para predição dos resultados de partidas de futebol / Felipe Ventura Kuhnert e Pedro Novaes Possato. – 2023.

77 f.

Orientadora: Giseli Rabello Lopes.

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Federal do Rio de Janeiro, Instituto de Computação, Bacharel em Ciência da Computação, 2023.

1. Análise de dados. 2. Distribuição de Poisson. 3. Distância de De Finetti. 4. Estatística no futebol. 5. Avaliação de predições. 6. Método interativo. I. Possato, Pedro Novaes. II. Lopes, Giseli Rabello (Orient). III. Universidade Federal do Rio de Janeiro, Instituto de Computação. IV. Título.

FELIPE VENTURA KUHNERT
PEDRO NOVAES POSSATO

MODELO PARA PREDIÇÃO DOS RESULTADOS DE PARTIDAS DE FUTEBOL

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 24 de agosto de 2023

BANCA EXAMINADORA:

gov.br Documento assinado digitalmente
GISELI RABELLO LOPES
Data: 27/08/2023 12:25:01-0300
Verifique em https://validar.iti.gov.br

Profa. Giseli Rabello Lopes
D.Sc. (UFRJ)

gov.br Documento assinado digitalmente
JOAO ANTONIO RECIO DA PAIXAO
Data: 28/08/2023 17:31:47-0300
Verifique em https://validar.iti.gov.br

Prof. João Antonio Recio da Paixão
D.Sc. (UFRJ)

gov.br Documento assinado digitalmente
JONY ARRAIS PINTO JUNIOR
Data: 28/08/2023 23:44:59-0300
Verifique em https://validar.iti.gov.br

Prof. Jony Arrais Pinto Junior
D.Sc. (UFF)

AGRADECIMENTOS

Gostaria de agradecer principalmente aos meus pais, os quais eu amo muito, que sempre me apoiaram durante todas as jornadas de minha vida, e sempre me trataram como humano e com dignidade, para me ensinar a fazer o mesmo com os outros. Agradeço também à Belinha, Chanel e Juninho, meus irmãos caninos que dão trabalho mas que me trazem muita felicidade em momentos ruins.

Agradeço a muitos de meus amigos, que sem eles não estaria onde estou hoje. Ao Henrique e o Lucas, que me incentivaram a cursar Ciência da Computação; ao Bernardo e ao Pedro, que me ajudaram muito em meus piores momentos; ao Breno Coll, Breno Tostes, Giulia Lima, Leonardo Ventura, Matheus Avellar, Mylena Lucciola, Pedro Dias e Pedro Possato por serem um grupo tão unido, que em seus primórdios no começo da faculdade, não tinha como eu saber que acabariam compondo uma parte tão significativa da minha vida, e que seriam tão importantes para mim; à Fernanda, Lori e Nina, que sempre me trouxeram felicidade me acompanhando nas minhas maluquices; e ao Gabriel, que é um amigo tão próximo desde muito cedo na minha vida.

Agradeço também aos meus colegas de trabalho, que em um mundo muitas vezes corporativista e burocrático tenho a oportunidade de participar de um ambiente fora desse padrão, considero todos como amigos de verdade que compartilham comigo momentos bons e momentos ruins, e que sempre me trazem alegria no dia a dia.

Finalmente, agradeço à professora Giseli, que nos ajudou muito na concepção deste trabalho, e nos permitiu transformar esse projeto de paixão em um trabalho digno para UFRJ.

Felipe Kuhnert

Gostaria de agradecer primeiramente ao meu avô Waldemar, um exemplo para mim e para minha família. Um ser de luz que hoje nos ilumina lá de cima, mas que continua presente diariamente em nossos pensamentos, em nossas ações e em nossos corações. Agradecer também à minha avó Fátima, a maior guerreira que já conheci, que continua provendo toda sua dedicação e amor à sua família.

Agradeço aos meus avós Silas e Sheila, peças fundamentais na minha criação. Se hoje posso desfrutar de algo, foi por mérito deles.

Obrigado ao meu pai, que sempre pôde me orientar e estar ao meu lado para me ensinar e me consolar, sempre com muito amor; e por compartilhar comigo sua paixão pelo futebol, sem a qual não seria possível criar um trabalho com tanto amor e carinho; à Luciana, que nunca mediu esforços para me ajudar e estar presente; à minha irmã Isabella, que eu amo profundamente e é uma criança incrível. Ao Bart, que emana amor e carinho.

Mãe, obrigado por estar sempre presente em meu caminho. Pelos conselhos que me dá, pelo amor que irradia, pelo riso e pelas lágrimas compartilhadas. Por tudo. Obrigado, João, por todo o carinho que sempre demonstrou e por todos os caminhos que abriu em minha vida.

Obrigado aos meus fiéis amigos Breno Coll, Breno Tostes, Felipe Kuhnert, Giulia Lima, Leonardo Ventura, Matheus Avellar, Mylena Lucciola e Pedro Dias. Não há um dia sequer desde que nos tornamos amigos que não sou presenteado com a possibilidade de rir, conversar ou desabafar com vocês, e por isso sou muito grato.

Agradeço aos meus queridos primos Carol, Gabi, João e Mateus, que são maravilhosos amigos e familiares, residindo em um pedaço especial do meu coração.

Por fim, agradeço a todos os meus professores e profissionais que fizeram parte da minha formação, especialmente à professora Kelly, que despertou em mim a vontade de criar o projeto do meu trabalho final, e à professora Giseli, que não mediu esforços para nos ajudar na realização do TCC.

Pedro Possato

"Terei que correr o sagrado risco do acaso.
E substituirei o destino pela probabilidade."

Clarice Lispector

RESUMO

O campo de análises esportivas baseadas em dados vem sofrendo um crescimento considerável nos últimos anos. Tal interesse tem sido impulsionado pela crescente proliferação de casas de apostas de futebol e sua constante divulgação através das propagandas veiculadas através dos meios de comunicação no Brasil. Com esta visão mais numérica e estatística do esporte, dados são utilizados para quantificar fatores que, em princípio, não são facilmente quantificáveis. Consequentemente, estes podem auxiliar análises empíricas esportivas e tomadas de decisão informadas, ou até proporcionar *insights* valiosos sobre as dinâmicas do esporte. Neste contexto, o presente trabalho apresenta um modelo proposto para predição de resultados de partidas de futebol através de um método iterativo que permite realizar análises probabilísticas sobre eventos de curto prazo, como resultados de partidas futuras, ou até eventos de longo prazo, como os possíveis campeões ou times a serem rebaixados em um campeonato. Este método iterativo utiliza valores numéricos para representar as capacidades ofensivas e defensivas dos times, que são atualizados a cada partida disputada. Foi implementado um *website* que fornece uma interface para facilitar interações do usuário com o modelo e a visualização de diferentes análises sobre as predições geradas. Por fim, são realizados experimentos comparativos entre os resultados de predição do modelo proposto e de vários outros modelos de predição para futebol do mercado, obtendo um resultado muito satisfatório e positivo.

Palavras-chave: análise de dados; distribuição de Poisson; distância de De Finetti; estatística no futebol; avaliação de predições; método iterativo.

ABSTRACT

The field of data-driven sports analysis has experienced significant growth in recent years, driven by the increasing presence of football betting platforms and their pervasive advertising across Brazilian media. With this more numerical and statistical view of the sport, data is used to quantify elements that are inherently challenging to measure. Consequently, this data can assist in empirical sports analysis and informed decision-making, or even provide valuable insights into the dynamics of the sport. In this context, this paper introduces a novel predictive model for football match outcomes, employing an iterative method that facilitates probabilistic assessments of short-term events, such as future match results, as well as long-term scenarios like potential champions or which teams will face relegation in a championship. This iterative technique employs numerical values to represent teams' offensive and defensive capabilities, which are continually updated with each match played. Additionally, a user-friendly website has been developed to offer an interface for user engagement with the model and to provide the visualization of diverse prediction analyses. Finally, comparative experiments are carried out between the prediction results of the proposed model and several other market football prediction models, obtaining a very satisfactory and positive outcome.

Keywords: data analysis; Poisson Distribution; De Finetti Distance; football statistics; prediction evaluation; iterative methods.

LISTA DE ILUSTRAÇÕES

LISTA DE CÓDIGOS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

xG	*eXpected Goals*
MME	Média Móvel Exponencial
JSON	*JavaScript Object Notation*
CSV	*Comma-Separated Values*
HTML	*HyperText Markup Language*
API	*Application Programming Interface*
URL	*Uniform Resource Locator*
SPI	*Soccer Power Index*

SUMÁRIO

1 INTRODUÇÃO

A análise de dados no contexto esportivo tem se mostrado uma ferramenta valiosa para compreender e prever como os times irão performar, assim como servir de base para a tomada de decisões estratégicas. A percepção das possibilidades proporcionadas pela aplicação de métodos quantitativos e análises estatísticas impulsionou a proposição de modelos para efetuar predições e prognósticos para diferentes modalidades esportivas (EMPACHER; KAMPS; VOLOVSKIY, 2023).

No futebol, por exemplo, um dos pontos de referência nessa área é a capacidade de quantificar aspectos que normalmente são considerados subjetivos. Através da utilização de algoritmos e modelos estatísticos, pode-se mensurar, por exemplo, a "força" de um time, avaliar suas chances de vitória em um campeonato ou até mesmo probabilidades de rebaixamento dos times. Essa abordagem objetiva e baseada em dados nos traz uma nova perspectiva para compreender mais sobre o esporte e suas dinâmicas.

Além disso, a utilização deste tipo de abordagem no futebol permite a análise de probabilidades e previsões de resultados de partidas. Essas estimativas podem ser utilizadas tanto para auxiliar times e técnicos ou treinadores na tomada de decisões estratégicas, como a escalação de jogadores, a definição de táticas e a análise de adversários, quanto para fins de apostas. Essa abordagem fundamentada em dados fornece uma dimensão analítica e científica para o esporte, podendo proporcionar *insights* valiosos para auxiliar na busca por estratégias para melhorar o desempenho dos times.

Esses pontos destacados demonstram a relevância e o potencial de abordagens estatísticas aplicadas no contexto de futebol como ferramentas de análise e suporte à tomada de decisões neste esporte. A aplicação dessas técnicas permite uma abordagem mais objetiva, embasada em dados e estatísticas, contribuindo para um maior entendimento e aprimoramento do esporte.

Dessa forma, impulsionada pela necessidade de ter uma abordagem quantitativa e objetiva para avaliar um assunto tão fascinante e inicialmente subjetivo, o presente trabalho propõe uma abordagem para realizar predições para o futebol. Visando explorar essa ideia sob um ponto de vista estatístico, os diferentes times são modelados através de parâmetros numéricos e avaliados entre si, a fim de gerar informações estatísticas sobre as partidas futuras. Além disso, é crucial que esses parâmetros possam ser atualizados após cada evento de partida, a fim de ajustar seus valores de acordo com o desempenho do time naquele evento. O modelo de Poisson Duplo (SILVA, 2014) foi adotado, desempenhando um papel fundamental na conexão entre os parâmetros criados e a conversão deles em probabilidades associadas a uma partida de futebol.

Algumas proposições originais permitiram criar uma abordagem com maiores detalhamentos e consequentes otimizações de resultados obtidos. Dentre elas, destacam-se

a utilização da Correção Rho (CORTIS, 2018); a consideração de viagens na definição do grau do mando de campo de uma partida; ou ainda a existência de desfalques de jogadores, possibilitando diminuir dinamicamente e temporariamente a força do time em questão para a partida calculada. Estes são apenas alguns exemplos das variáveis que podem afetar as previsões e que foram contempladas na solução proposta.

O modelo proposto foi avaliado utilizando o método da distância de De Finetti (DE FINETTI, 1972), comparando-o a diversos outros modelos e algoritmos propostos em trabalhos relacionados. Com essa avaliação, é possível perceber o sucesso do modelo desenvolvido, ao se apresentar como superior nas avaliações realizadas.

O restante deste trabalho está organizado da seguinte forma:

- No capítulo 2, apresenta-se a fundamentação teórica, incluindo a discussão de abordagens e métodos que podem ser usados no contexto de realização e avaliação de predições.

- No capítulo 3, descrevem-se os trabalhos que serviram de inspiração, apoio e embasamento para a concepção e implementação do projeto final, fazendo uma breve explicação de suas propostas e estabelecendo relações com o tema abordado.

- No capítulo 4, apresenta-se a a abordagem de predição proposta neste trabalho e as tecnologias utilizadas em sua implementação.

- No capítulo 5, detalha-se a implementação da abordagem proposta, incluindo a apresentação de fragmentos do código fonte e suas otimizações. Além disso, apresenta-se a página web desenvolvida para interação com os dados gerados através das predições, tais como: resultados de partidas futuras, *rankings* de times, simulações de campeonatos em qualquer data desde a criação da ferramenta e apresentação de gráficos que retratam as forças e desempenho de cada time ao longo da temporada atual.

- No capítulo 6, demonstram-se os resultados experimentais com dados reais obtidos. Inclui um comparativo entre a abordagem proposta e outros trabalhos do estado da arte.

- No capítulo 7, realiza-se uma síntese do tema abordado, concluindo-o e apresentando possíveis direções para trabalhos futuros, como aprimoramentos na usabilidade para usuários finais e na automatização do processo.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo destina-se a explicar diversos conhecimentos essenciais ao projeto, para que o leitor possa primeiro se acostumar com seus significados e objetivos, e a partir de então entender, nos capítulos posteriores, seus funcionamentos conjuntos. Os tópicos discutidos incluem abordagens utilizadas para realizar e avaliar predições, especialmente aplicadas no contexto do futebol.

2.1 PREDIÇÃO

Esta seção dedica-se à explicação dos métodos utilizados ao longo do desenvolvimento do projeto para realizar predições e análises das diversas situações e possibilidades de partidas e campeonatos de futebol. Para que seja possível realizarmos predições, porém, é necessário que consigamos mapear partidas reais de futebol matematicamente.

2.1.1 Gols Esperados (xG)

Um desses mapeamentos se chama "Gols Esperados", ou *"Expected Goals"* (HAMILTON, 2009), normalmente abreviado como xG. Este se refere a uma estatística comumente utilizada em jogos de futebol, mas também pode estar presente em outros esportes, como futebol americano e hóquei, por exemplo. xG, como o nome diz, indica a quantidade de gols esperados para uma situação dada, podendo ser aplicado no contexto de um lance específico (exemplo: um chute a gol de uma determinada posição ter xG igual a 0,07), uma partida com um todo (exemplo: o time A teve xG igual a 3,7, mas só marcou 1 gol), ou até um jogador (exemplo: em uma partida, jogador J teve um xG acima da média de seus companheiros).

Os valores de xG são contínuos, e não discretos. Um valor entre 0 e 1, por exemplo, quando um lance específico está sendo examinado, indica a probabilidade desse lance resultar em um gol para o time atacante. Pênaltis, por exemplo, possuem um xG conhecido que varia entre 0,76 e 0,79[1], ou seja, a cada pênalti cobrado regularmente, existe uma média de 76%-79% de chance dele resultar em um gol para o time a favor.

Caso seja examinado o xG de uma partida, porém, é comum observar números maiores que 1. Em uma partida onde o xG é representado como (2,71; 1,32), por exemplo, nos dá o valor de xG de cada time:

- 2,71, o xG do time com o mando de campo (time de casa), indica que esse time criou N oportunidades com um valor entre 0 e 1 representando a probabilidade de

[1] Valor calculado a partir da média de gols marcados por cobrança de pênalti ao decorrer de campeonatos longos

cada uma dessas oportunidades, tal que a soma das probabilidades criadas resultou em 2,71. Esperava-se desse time, portanto, um total de gols entre 2 e 3.

- 1,32, o xG do time sem o mando de campo (time visitante), indica que esse time criou N oportunidades com um valor entre 0 e 1 representando a probabilidade de cada uma dessas oportunidades, tal que a soma das probabilidades criadas resultou em 1,32. Esperava-se desse time, portanto, um total de gols entre 1 e 2.

Note que, como o nome diz, o valor de xG é apenas o valor esperado de gols. Por mais que um time possua um xG alto, são inúmeros os casos onde a realidade não reflete o valor esperado. Por isso, é normalmente utilizado mais como um indicador de chances de gol que um time conseguiu gerar durante determinada partida, o que pode refletir mais fielmente o nível de jogo que o time apresentou, do que simplesmente olhar o placar da partida. Este valor, quando calculado em modelos de predição ou algoritmos para representar a previsão de uma partida, pode ser utilizado como um panorama sobre a "força de ataque" de um determinado time, influenciando diretamente a probabilidade de vitória.

2.1.2 Poisson Duplo

Outro conceito que pode ser utilizado na predição de resultados de partidas de futebol é o Modelo Poisson Duplo (SILVA, 2014). Porém, antes de explicar esse conceito, primeiro precisamos entender a distribuição de Poisson.

A distribuição de Poisson é uma distribuição de probabilidade de uma variável aleatória discreta. Seja X: o número de eventos que ocorrem em um determinado período de tempo, como por exemplo o número de gols marcados pelo time A em uma partida de futebol, pode-se estimar a probabilidade de observar X eventos pela Equação 2.1. Considera-se que esses eventos sejam independentes do momento em que ocorreram os eventos anteriores.

$$\text{Poisson}(k;\lambda) = \frac{e^{-\lambda} \cdot \lambda^k}{k!} \tag{2.1}$$

Onde:

- k = Número de ocorrências do evento calculado
- λ = Número médio de ocorrências
- e = Base do logaritmo natural
- $k!$ = Fatorial de k

Com essa distribuição pode-se, dada uma média de ocorrências de um determinado evento, calcular as chances desse evento ocorrer k vezes, sendo k um inteiro não negativo.

Ou seja, por meio dessa distribuição, podemos calcular a probabilidade de um time fazer k gols em uma partida, desde que saibamos em média quantos gols aquele time faz por partida, representado por λ.

Com esse conhecimento, pode-se aplicar a Poisson Dupla, a qual trata-se de aplicarmos a distribuição de Poisson para dois λ diferentes e calcular as permutações dos resultados possíveis. Assim, pode-se encontrar a probabilidade de dois times diferentes marcarem uma certa quantidade de gols, e com isso teremos a probabilidade de cada placar.

Por exemplo, para calcularmos a chance de um jogo entre Time A e Time B terminar Time A 2 x 1 Time B, podemos aplicar a Poisson Dupla. Suponhamos que o λ do Time A seja 1,5 e do Time B seja 1,2, ou seja, que é esperado que o Time A faça em média 1,5 gols, enquanto para o Time B esperamos 1,2 gols.

Logo, a chance do Time A marcar 2 gols será:

$$\text{Poisson}(2; 1{,}5) = \frac{e^{-1{,}5} \cdot 1{,}5^2}{2!} = 0{,}2510 = 25{,}10\%$$

Para o Time B marcar 1 gol, teremos:

$$\text{Poisson}(1; 1{,}2) = \frac{e^{-1{,}2} \cdot 1{,}2^1}{1!} = 0{,}3614 = 36{,}14\%$$

Portanto, a chance do placar final ser Time A 2 x 1 Time B será:

$$0{,}2510 \cdot 0{,}3614 = 0{,}0907 = 9{,}07\%$$

Aplicando essa lógica para todos os placares possíveis (assumimos um valor máximo como 25 para o limite de gols que um time irá fazer, uma vez que 25 se mostra suficientemente grande para que a soma das probabilidades seja muito próxima de 100%), teremos a chance de cada placar ocorrer. A Tabela 1 apresenta as probabilidades dos possíveis placares mais comuns para o exemplo supracitado.

Tabela 1 – Exemplo: Chances de alguns dos possíveis resultados da partida exemplo

Gols do Visitante \ Gols do Time de Casa	0	1	2	3	4
0	6,72%	10,08%	7,56%	3,78%	1,42%
1	8,06%	12,1%	9,07%	4,54%	1,7%
2	4,84%	7,26%	5,44%	2,72%	1,02%
3	1,94%	2,9%	2,18%	1,09%	0,41%
4	0,58%	0,87%	0,65%	0,33%	0,12%

Com tais valores calculados, resta apenas somar todas as probabilidades de placares que dão a vitória ao Time A para descobrirmos a chance do Time A vencer. A mesma lógica pode ser aplicada para descobrirmos as chances de empate e de vitória do Time B.

Realizando essas somas, ficamos com 42,30% de chance de vitória para o Time A, 25,47% para o empate, e 29,61% para a vitória do Time B. Podemos perceber que a

soma dessas probabilidades resulta em 97,38%, evidenciando a necessidade de também calcularmos combinações além de quatro gols para cada time, pois as probabilidades envolvendo placares em que pelo menos um time marque cinco ou mais gols se mostrou em $2,62\%$. Esse exemplo evidencia o motivo pelo qual é necessário avaliar números maiores de gols por time, para garantir que a soma das probabilidades seja muito próxima a 100%.

2.1.3 Correção Rho - Inflação e deflação de empates

Na Seção 2.1.2, foi apresentado o cálculo das probabilidades de vitória, empate e derrota de um time em uma partida, o qual é realizado por meio de uma distribuição de Poisson Dupla. É importante ressaltar, no entanto, que esse método assume a independência entre o número de gols marcados por cada um dos times. Em outras palavras, a aplicação desse método deixa implícito que em uma partida entre o Time A e o Time B, a chance do Time A fazer exatamente 1 gol independe do fato do Time B ter feito 0 gols, 1 gol, 10 gols, etc.

Como essa relação não é verdadeiramente independente, observamos um fenômeno que distorce um pouco a realidade de uma modelagem puramente fiel à distribuição de Poisson Dupla. Esse fenômeno trata-se da ocorrência de mais empates do que o modelo sugere. Um trabalho semelhante (CORTIS, 2018) julga que o problema está no placar de $0x0$, mas em nosso escopo foi julgado a existência dessa distorção em todos os empates, de maneira proporcional.

Assim, podemos empregar a correção Rho com o objetivo de corrigir essa distorção criada. Para tal, geramos um valor otimizado "α" para a correção, por meio de uma massa de dados de campeonatos e partidas já ocorridas. Toda essa massa de dados capturada é avaliada e são aplicados diferentes valores para a variável Rho. Aquela que apresentar maior proximidade à realidade no que diz respeito ao número de empates esperados será usada, até que mais dados extraídos mudem a realidade do valor que melhor representa Rho. Essa estratégia, onde novos valores são aplicados em cima de massas de dados antigas para realizar análise e otimização das variáveis é chamada *Backtest*, e é análoga a estratégia de mesmo nome utilizada no mercado financeiro (CHEN, 2021).

Com esse valor definido, podemos aumentar a probabilidade de empate calculada por α. Após esse aumento, porém, a tripla que representa os três resultados possíveis para a partida passa a somar mais do que 100%. Por isso, precisamos corrigir os demais valores, multiplicando-os por uma mesma constante β, tal que a nova tripla continue somando 100.

Em termos matemáticos, teremos inicialmente uma tripla de probabilidades $(V; E; D)$, denotando respectivamente vitória, empate e derrota para o time da casa, as quais somadas resultam em 100%. Ao corrigir o valor de empates, teremos $(V; \alpha E; D)$, sendo $\alpha > 1$, resultando em uma soma da tripla superior a 100%. Para tal, usaremos um valor β responsável por retomar a soma de 100%, resultando em uma tripla de $(\beta V; \alpha E; \beta D)$.

Para calcularmos o valor de β em função dos valores conhecidos, podemos isolar β a partir da seguinte equação:

- $V \cdot \beta + E \cdot \alpha + D \cdot \beta = 1$
- $(V + D) \cdot \beta + E \cdot \alpha = 1$
- $\beta = \frac{(1-E \cdot \alpha)}{(V+D)}$

Logo, ficamos com a tripla final conforme apresentada na equação 2.2.

$$\left(V \cdot \frac{1 - E\alpha}{V + D} \ ; \ E\alpha \ ; \ D \cdot \frac{1 - E\alpha}{V + D} \right) \tag{2.2}$$

Por exemplo, a partida entre os times A e B teve uma tripla calculada como (0,40; 0,40; 0,20) via Poisson Dupla. Ou seja, a chance de vitória do Time A está em 40%, a chance de empate está em 40% e a chance de vitória do Time B está em 20%. Agora, precisamos calcular a nova tripla supondo uma correção Rho com $\alpha = 1,10$.

$$\left(0{,}40 \cdot \frac{1 - 0{,}40 \cdot 1{,}10}{0{,}40 + 0{,}20} \ ; \ 0{,}40 \cdot 1{,}10 \ ; \ 0{,}20 \cdot \frac{1 - 0{,}40 \cdot 1{,}10}{0{,}40 + 0{,}20} \right)$$

Realizando esse cálculo, chegaremos à seguinte tripla: $(0{,}3733; 0{,}44; 0{,}1867)$, onde a soma dos termos resulta em 100%, e a proporção vitória do Time A por vitória do Time B permanece inalterada.

2.1.4 Média Móvel Exponencial

A Média Móvel Exponencial (MME) (HUNTER, 1986) é um indicador de análise técnica amplamente utilizado no mercado financeiro, onde temos um ativo com diversos preços de fechamento demarcados ao longo de um período de tempo. Nessa análise, a MME aplica uma equação exponencial que calcula a média ponderada dos preços de fechamento do ativo para o período existente. Diferente da média móvel simples, a MME aplica ênfase maior aos preços mais recentes, o que a torna mais responsiva às mudanças do mercado. Ela pode ser calculada conforme a equação 2.3.

$$\text{MME}(t) = (1 - \gamma) \cdot \text{MME}(t - 1) + \gamma \cdot X(t) \tag{2.3}$$

Onde:

- MME(t) é o valor da Média Móvel Exponencial no momento t.
- MME($t - 1$) é o valor anterior da Média Móvel Exponencial no momento $t - 1$.
- N é o período da Média Móvel Exponencial utilizada.

- $\gamma = \frac{2}{N+1}$
- $X(t)$ é o valor atual da série de dados no momento t.

Veremos mais adiante que esse método pode ser utilizado para atualizar variáveis e coeficientes em um modelo de predição iterativo. Por exemplo, o utilizaremos para atualizar variáveis de força de ataque e defesa dos times. Analogamente, os valores de cada iteração são avaliados e interpretados da mesma forma que o preço de fechamento de um ativo no dia: dessa forma, estes valores estariam sempre sendo atualizados dando maior peso a iterações ocorridas mais recentemente do que a iterações mais antigas, que podem não representar a realidade tão fielmente quanto as mais atuais.

2.2 AVALIAÇÃO

Enquanto as seções anteriores deste capítulo abordam vários conceitos utilizados na predição de resultados, os conceitos seguintes tratam sobre como podemos avaliar as predições calculadas para melhorar a qualidade de um modelo de predição, assim como realizar comparações entre diferentes modelos de predição para determinar qual está performando melhor.

2.2.1 Distância de De Finetti

Uma dessas métricas que pode ser utilizada para analisar matematicamente as previsões geradas por modelos e algoritmos de previsão de partidas de futebol é a Distância de De Finetti (DE FINETTI, 1972). Criada pelo matemático italiano Bruno De Finetti, este método de avaliação é muito útil para medir a "distância" do resultado de um evento aleatório com as probabilidades calculadas por um modelo de predição de cada evento acontecer. Como explicitado anteriormente, esta medida é utilizada por outros algoritmos e modelos de predição, como (ARRUDA, 2023) e (ARRUDA, 2000).

A Distância de De Finetti é definida pela distância quadrática euclidiana entre os valores previstos e o resultado real. No futebol, como existem 3 possíveis resultados de uma partida (vitória do time da casa, empate, vitória do time visitante), podemos modelar as previsões como pontos no R^3, e a medida de interesse será a distância entre o evento real e o ponto das previsões.

Vamos supor que um modelo de previsão qualquer estimou $(0{,}65; 0{,}1; 0{,}25)$ para o resultado de uma partida, isto é, 65% de chance do time da casa vencer, 10% de chance de empate, e 25% de chance do time visitante vencer. A Distância de De Finetti, dependendo do resultado da partida, seria:

$$\begin{cases} (0{,}65-1)^2+(0{,}1-0)^2+(0{,}25-0)^2=0{,}195, & \text{se o time da casa vencer} \\ (0{,}65-0)^2+(0{,}1-1)^2+(0{,}25-0)^2=1{,}295, & \text{se houver um empate} \\ (0{,}65-0)^2+(0{,}1-0)^2+(0{,}25-1)^2=0{,}995, & \text{se o time visitante vencer} \end{cases}$$

Note que após o resultado final de uma partida, temos o ponto definido pela realidade: em caso de vitória do time da casa, este ponto está em $(1, 0, 0)$; em caso de empate está em $(0, 1, 0)$; e em caso de vitória do time visitante está em $(0, 0, 1)$.

É observado então que, quanto mais "errada" estiver a previsão, maior será a distância observada entre o ponto previsto e o ponto do resultado real. Os valores possíveis da Distância de De Finetti em R^3 possuem um intervalo de $[0, 2]$, e quanto menor for, melhor é a previsão.

Para avaliar os modelos ao decorrer de um campeonato, pode-se utilizar a média da Distância de De Finetti de todos os resultados previstos por cada modelo. Os modelos com as menores distâncias médias são os que possuem maior precisão, que é um fator de alta importância para um modelo de predição.

Outra informação importante que pode ser extraída da Distância de De Finetti é que é possível estabelecer um limite mínimo de viabilidade de um modelo de previsão. Vamos analisar a seguinte situação: Propomos um modelo de previsão preguiçoso que sempre atribui chances iguais para os resultados possíveis. Isto é, para todas as partidas, a tripla de previsão será $(0, \overline{3}, 0, \overline{3}, 0, \overline{3})$. Como os três valores são iguais, a Distância de De Finetti sempre será independente do resultado:

$$(0{,}\overline{3}-1)^2+(0{,}\overline{3}-0)^2+(0{,}\overline{3}-0)^2=0{,}\overline{6}$$

Ou seja, qualquer modelo ou algoritmo que tenha uma Distância de De Finetti maior que $0{,}\overline{6}$ pode ser visto como inferior a simplesmente assumir probabilidades iguais aos três resultados possíveis.

2.2.2 Monte Carlo

Um modelo de predição é capaz de calcular as probabilidades dos resultados de cada partida ao decorrer de um campeonato. Porém, caso seja de interesse calcular a probabilidade de cada time vencer o campeonato, ser rebaixado, ou qualquer outra estatística relevante que diga respeito a um futuro mais complexo do que o simples cálculo da predição do resultado de uma partida, pode-se adotar a estratégia da geração de um grande número de dados.

O Método de Monte Carlo (METROPOLIS; ULAM, 1949) consiste na utilização da geração de um grande número de eventos aleatórios para estimar um resultado. Como foi explicado em seções anteriores, modelos de predição de partidas de futebol geram probabilidades de vitória/empate/derrota para cada confronto entre dois times. Com

isso, o Método de Monte Carlo é utilizado para realizar a previsão do resultado de um campeonato com base nos resultados anteriores e atuais forças dos times, iterativamente.

O resultado de cada partida é estimado, com base nas predições de resultados calculados pelo modelo para elas, usando um número gerado aleatoriamente. O mesmo é feito para todos os embates subsequentes, até o fim do campeonato. Isso constitui uma simulação. Para aplicarmos o Método de Monte Carlo, precisamos realizar N simulações (N sendo suficientemente grande), e calculamos em qual porcentagem dessas simulações cada time alcançou determinados resultados, como por exemplo ser campeão. Assim, geramos a estimativa da chance que cada time tem de realizar tais feitos, dada a situação atual.

Uma métrica gerada por Monte Carlo que será analisada mais a frente é a chance de vitória de cada time ao decorrer de um campeonato. Esse trabalho, por exemplo, se adapta rapidamente a mudanças nas forças de um time, porém não é possível prever tais mudanças com precisão muito antes de acontecerem. Por isso, observamos vários times que começam um campeonato muito bem, mas que suas chances de vencerem o campeonato vão caindo com o tempo, e vice-versa: times que têm um começo mais lento mas que acabam estando entre os principais candidatos ao título.

Para exemplificar uma simulação via Monte Carlo, podemos assumir que faremos 20.000 simulações, seja 20.000 um número suficientemente grande, de um campeonato que ainda não se iniciou, e que desejamos calcular as probabilidades de cada time ser campeão do torneio. O modelo, nesse exemplo, irá considerar que nenhum jogo tem resultado conhecido, uma vez que nenhuma partida ocorreu ainda.

Dessa forma, faremos simulações de todas as partidas a serem realizadas, adotando para cada partida probabilidades correspondentes ao que o modelo de predição hoje entende sobre aquela partida. Caso o Time A vá enfrentar o Time B, e essa partida possua estimativas de 50% de chance de vitória do Time A, 30% de chance de empate e 20% de chance de vitória do Time B pelos cálculos do modelo, será realizado um sorteio que defina um resultado para essa partida, considerando os pesos devidos pelas probabilidades associadas a cada evento.

Suponhamos que ao fim do sorteio de resultados de todas as partidas do campeonato, tenha-se definido que o time X somou o maior número de pontos, e portanto, é o campeão da simulação realizada. Após a repetição desse processo 20.000 vezes (valor definido mais acima como exemplo), teremos 20.000 campeões. Caso o time X tenha sido o campeão de 8.000 simulações, teremos que o evento de time X sendo campeão ocorreu 8.000 vezes, dentre as 20.000 possíveis, e portanto, esse time possui 40% de chance de ser campeão. Esse valor é estimado para todos os times e para todas as métricas que desejamos calcular.

Dessa forma, podemos mapear condições a se calcular, como se o time ficou na primeira colocação (ser campeão), se ficou entre os quatro últimos (ser rebaixado), entre outras, para acompanhar durante as 20.000 simulações quais resultados podem ser extraídos para essas métricas. Realizando a simples regra de três observada no parágrafo anterior, somos

capazes de calcular essas porcentagens e obter o resultado desejado.

2.3 CONSIDERAÇÕES FINAIS

Compreender os fundamentos dos métodos de predição e avaliação é essencial, mas a verdadeira força reside na habilidade de sintetizar esses aspectos em um processo coeso e eficaz. Ao agregarmos os conhecimentos discutidos até aqui, construímos um arcabouço conceitual que nos capacita a não apenas realizar previsões assertivas sobre resultados de partidas e campeonatos de futebol, mas também a aprimorar continuamente nosso modelo preditivo.

No próximo capítulo, ao explorar as contribuições e abordagens de outros estudos nesse campo, poderemos ampliar nossa visão e estabelecer um diálogo enriquecedor com as diversas perspectivas possíveis de se abordar o tema. A partir dessa base sólida de conhecimento, o capítulo seguinte se propõe a explorar as interseções, divergências e singularidades existentes entre os diferentes projetos.

3 TRABALHOS RELACIONADOS

Este capítulo aborda trabalhos que possuem grande relevância sobre os assuntos tratados ao longo do desenvolvimento deste trabalho. Algumas partes estudadas foram fundamentais para a concepção do modelo de predição desenvolvido, e no final deste capítulo é feita uma comparação entre estes trabalhos e o presente trabalho.

3.1 CHANCE DE GOL

Arruda (2000) introduz a ideia de utilizar sistemas de equações para realizar o cálculo das forças de cada time e, a partir dessas forças, estimar as probabilidades de uma partida de futebol, com uso do conceito do Modelo Poisson Duplo. Sua explicação envolve um grande rigor matemático sobre os possíveis métodos a serem utilizados para realizar a predição e avaliação das partidas usando apenas métodos estatísticos. Este artigo traz a ideia da utilização da Distância de De Finetti, que foi explicada a fundo no Capítulo 2, para medir a qualidade das predições obtidas.

Após a sua dissertação de mestrado, Arruda continuou a publicar trabalhos relacionados ao uso de estatística no futebol. Seu website (ARRUDA, 2023) detalha, com menos rigor matemático, mas com uma abundância de exemplos práticos e massas de dados de temporadas atuais e passadas de campeonatos de futebol, um modelo estatístico para avaliação dos melhores times dos campeonatos e predição dos resultados de partidas. O modelo do Chance de Gol realiza um cálculo das forças dos times, que vai sendo atualizado a cada rodada de um campeonato, para determinar qual será o resultado da próxima partida disputada entre dois times específicos. As forças dos times são atualizadas com base nos resultados das partidas, utilizando as técnicas desenvolvidas em sua dissertação de mestrado (ARRUDA, 2000), onde recebem um peso maior as partidas mais recentes, já que estas são melhores em avaliar a força de um time do que partidas mais antigas. Além disso, neste modelo, times com mando de campo possuem uma força maior comparada à força do mesmo time caso estivesse jogando fora de casa. Ao longo de um campeonato, Arruda calcula as tendências que cada time possui para vencer o campeonato, ser rebaixado ao final da temporada, dentre outros.

3.2 FIVETHIRTYEIGHT

Outro trabalho relacionado bastante relevante para nosso estudo é o algoritmo desenvolvido pelo Five... (2023). Devido à baixa quantidade de gols que ocorrem em cada partida, o resultado final pode não refletir a qualidade do jogo de cada time, e a natureza

de baixo placar do esporte, às vezes, leva a períodos prolongados de sorte, onde um time pode estar obtendo bons resultados apesar de jogar mal (ou vice-versa).

Para mitigar essa aleatoriedade e melhor estimar a qualidade subjacente de jogo de cada time, Five... (2023) utiliza quatro métricas para avaliar o desempenho de uma equipe após cada partida: gols, gols ajustados, gols esperados baseados em finalizações e gols esperados não baseados em finalizações.

A primeira métrica, "gols", trata simplesmente da quantidade de gols realizadas por cada time, sendo de fácil e direto entendimento.

Os gols ajustados levam em conta as condições em que cada gol foi marcado, reduzindo o valor dos gols marcados quando um time tem mais jogadores em campo e quando um time já está vencendo por uma grande margem. O valor de todos os outros gols é aumentado para que o número total de gols ajustados geralmente some ao número total de gols reais marcados ao longo do tempo.

Os gols esperados baseados em finalizações são uma estimativa de quantos gols um time "deveria" ter marcado, dadas as finalizações que foram por ela dadas naquela partida. A cada chute é atribuída uma probabilidade de gol com base em sua distância e ângulo em relação ao gol, bem como a parte do corpo com a qual a finalização foi dada, com um ajuste para o jogador que a realizou. Essas probabilidades são adicionadas para produzir o número de gols esperados baseados em finalizações para aquela partida, que pode ser maior ou menor do que o número de gols que o time realmente tenha marcado.

Os gols esperados não baseados em finalizações são uma estimativa de quantos gols um time "deveria" ter marcado com base em ações não relacionadas a finalizações que tenha sido realizadas em torno do gol da equipe adversária, como passes, interceptações, dribles e desarmes. Essas ações individuais são adicionadas em toda a partida para chegar aos gols esperados não baseados em finalizações daquele time. Assim como para gols esperados baseados em finalizações, há um ajuste para cada ação com base nas taxas de sucesso do jogador ou jogadores que realizaram a ação (tanto o passador quanto o receptor, no caso de um passe).

Com isso, a pontuação ofensiva de um time para uma partida será a média de seu desempenho em todas as quatro métricas, e sua pontuação defensiva será a média das quatro métricas para seu oponente. Esse placar gerado, então, é utilizado para qualificar a partida de ambos os times e ajustar suas forças de acordo com o que se esperava dos times na partida, e o que foi observado. Dessa forma, os valores dos times são atualizados, e podem ser estimados para partidas futuras de acordo com as novas forças de ataque e defesa calculadas.

Finalmente, é aplicado um processo de Poisson Duplo para extrair as probabilidades de vitória, empate e derrota a partir dos valores estimados de gols para cada time em determinada partida.

3.3 PROBABILIDADES NO FUTEBOL

A equipe do projeto "Probabilidades no Futebol" (LIMA et al., 2023), desenvolvido por um grupo de pesquisadores do Departamento de Matemática da Universidade Federal de Minas Gerais (UFMG), desenvolveu um algoritmo de predição de futebol. Embora haja uma carência de fontes abordando seu conteúdo, é possível observar empiricamente uma característica distintiva em relação a outros algoritmos: suas predições iniciais são imparciais. Isso significa que, na primeira rodada do campeonato, todos os times são considerados igualmente fortes. Essa abordagem pode ser interpretada de forma negativa, se comparada a modelos mais sofisticados que conseguem avaliar corretamente o desempenho de cada time antes do início do campeonato. Por outro lado, pode ser considerada positiva ao realizar uma comparação desprovida de conceitos pré-estabelecidos em relação a cada time. Nessa perspectiva, os times terão seus desempenhos avaliados de acordo com suas produtividades ao longo do campeonato, resultando em análises mais imparciais, embora o algoritmo não seja tão preciso desde o início.

Outro aspecto interessante e particular dessa abordagem é a clara distinção das forças de cada time em seus jogos como mandante e visitante. O algoritmo proposto no contexto do projeto "Probabilidades no Futebol" considera as forças de cada time jogando em seu próprio estádio e fora dele de forma independente. Isso significa, por exemplo, que um time pode ser mais forte jogando como visitante do que como mandante. Esse conceito, quando comparado aos demais trabalhos na área, é notavelmente distinto, uma vez que geralmente a força de um time é considerada única, e o critério de mando de campo é apenas aplicado a ela, ao invés de compô-la.

3.4 ESPIÃO ESTATÍSTICO

O algoritmo do Globo Esporte (ge) chama-se Espião Estatístico (ESPIÃO..., 2023). Ele avalia o rendimento dos times do campeonato brasileiro, usando como base de dados principal os últimos 60 dias, tanto como mandantes quanto como visitantes, em todas as competições, bem como nos últimos seis jogos oficiais, independente do mando de campo.

De acordo com as informações disponíveis em seu website, essa análise leva em consideração tanto a performance defensiva quanto a ofensiva dos times, tanto em jogadas aéreas como em jogadas pelo chão. Além disso, são efetuados cálculos específicos que relacionam a influência de bolas altas e trocas de passes rasteiros com gols marcados e sofridos, focando exclusivamente nas características dos gols obtidos durante as jogadas. Gols olímpicos, cobranças de pênaltis e faltas diretas não são considerados nesse contexto, pois são cobranças diretas para o gol.

As probabilidades estatísticas apresentadas são baseadas no modelo de "Gols Esperados" ou "Expectativa de Gols" (xG), uma métrica amplamente empregada na análise de dados. Esse modelo foi desenvolvido a partir de uma base de dados contendo 95.181

finalizações registradas pelo Espião Estatístico em 3.869 partidas dos Brasileirões desde a edição de 2013. Diversos parâmetros são levados em conta nessa análise, como a distância e o ângulo da finalização, a origem da jogada, a parte do corpo utilizada, o tempo de jogo, a diferença de valor de mercado entre os times e a diferença no placar no momento da finalização.

Por exemplo, a expectativa de gol (xG) para uma finalização da meia-lua é de cerca de 0,07, ou seja, em cada cem finalizações dessa região do campo, apenas sete acabam se concretizando em gols. Cada posição no campo possui uma expectativa de gol distinta, a qual pode ser influenciada pelo contexto da partida, como no caso de um contra-ataque, em que há menos adversários para evitar a conclusão da jogada. A pontuação é acumulada ao longo da partida, a fim de obter o xG total de um time em cada jogo. O modelo utilizado nas análises também segue a Distribuição de Poisson Dupla, algo comum nos algoritmos observados.

3.5 COMPARAÇÃO DOS ALGORITMOS E NOSSO DIFERENCIAL

Ao compararmos os modelos e algoritmos descritos, podemos observar diferentes abordagens utilizadas para resolver problemas semelhantes. Alguns conceitos, como a Poisson Dupla, são comumente encontrados na maioria dos modelos, enquanto outras escolhas, como a resolução de sistemas de equações ou a inicialização do campeonato com forças idênticas para todos os times, são mais específicas de cada algoritmo, diferenciando-os uns dos outros.

O projeto desenvolvido neste projeto final não foge dessa lógica, apresentando elementos similares aos compartilhados por outros algoritmos, bem como elementos diferenciados que contribuem para a singularidade de nossa abordagem. Podemos mencionar brevemente alguns conceitos comuns, como a utilização da Poisson Dupla ou a ideia de gols esperados, que também estão presentes em nossa implementação. No entanto, destacamos elementos distintivos, como a aplicação da correção Rho, a consideração de diferentes forças para diferentes mandos de campo, a inclusão de informações sobre desfalques nas partidas, entre outros. Além disso, o nosso projeto é desenvolvido com um foco mais computacional, algorítmico, do que uma abordagem mais matemática; desenvolvemos um modelo de predição iterativa, onde a cada rodada os valores vão sendo atualizados para aprimorar o programa cada vez mais.

Essas características evidenciam os diferenciais do nosso projeto e seu destaque em relação aos demais algoritmos e modelos de predição. A combinação desses elementos específicos resulta em um modelo que busca melhorar a precisão e confiabilidade na predição de resultados de jogos de futebol.

4 MODELO DE PREDIÇÃO

O objetivo deste trabalho é desenvolver um modelo de predição de resultados de futebol. Buscamos elaborar um modelo iterativo de previsão, utilizando conceitos estatísticos aplicados ao contexto do esporte, a fim de aprimorar seu desempenho a cada iteração.

4.1 DEFINIÇÃO DE ESCOPO

A partir destes conceitos e dos conceitos explicados no Capítulo 2, temos então o objetivo de desenvolver um modelo de predição para prever os resultados das partidas e das diversas ocorrências em um campeonato. Algo importante de ser considerado é que diferentes campeonatos de futebol possuem diferentes formatos e durações. Campeonatos menores, com fases eliminatórias nas quais cada jogo possui uma importância muito alta para cada time, tornam difícil a estimação de resultados usando estatística. Exemplos destes formatos mais curtos são a Copa do Brasil, Libertadores e até a Copa do Mundo: com poucas partidas temos uma amostragem pequena para um modelo iterativo se aperfeiçoar.

Por isso, o caso ideal para um modelo de predição de partidas de futebol usando conceitos estatísticos em cima de grandes massas de dados são campeonatos no formato de disputa em pontos corridos. Um exemplo deste tipo de campeonato é o Brasileirão: 20 times jogam um total de 38 rodadas, ao longo do ano, totalizando 380 partidas[1]. Não existem fases eliminatórias: cada jogo possui a mesma importância em questão de pontuação, e ao final de todas as rodadas, o time que está em 1° lugar vence. Outros exemplos de campeonatos como este são a *Premier League*, na Inglaterra; a *Bundesliga*, na Alemanha; a *La Liga*, na Espanha; dentre vários outros.

Com um grande total de partidas disputadas, foi elaborado um modelo de predição iterativo que atualiza a cada rodada, de acordo com o resultado das partidas, os coeficientes dos times, como a "Força de Ataque" e "Força de Defesa" descritos anteriormente. A partir destes coeficientes, os resultados das partidas futuras são previstos, e assim começa uma nova iteração do modelo.

Além disso, é de nosso interesse que o modelo seja capaz de estimar as chances de vários acontecimentos de um campeonato, como a probabilidade de cada time ser campeão, ou a probabilidade de cada time ficar entre os últimos colocados e ser rebaixado, dentre outros. Para isso, simulações do restante do campeonato podem ser realizadas a fim de mensurar as probabilidades de cada evento acima mencionado ocorrer.

Por fim, é de suma importância que todos os dados gerados pelo modelo de predição sejam guardados a cada rodada, para possibilitar a realização de análises de aprimora-

[1] Dados considerando a edição da série A no ano de 2023.

mento do modelo e também de comparação com outros algoritmos e modelos de predição. A precisão, por exemplo, é uma das métricas mais importantes para um modelo deste gênero, e tem seus resultados apresentados na Seção 6.4. Com isso, temos o fluxo de funcionamento do modelo, que representa cada iteração:

1. Resultados de partidas são previstas pelo modelo
2. Partidas acontecem e os resultados reais são obtidos
3. Forças de ataque e defesa dos times são atualizados conforme os resultados reais comparado aos resultados previstos
4. Dados históricos são salvos
5. Simulações do resto do campeonato são executadas para estimar as métricas desejadas, como campeão, G4, Z4, etc

4.2 ABORDAGEM PROPOSTA

O problema central que fundamenta o desenvolvimento do modelo de predição consiste em determinar a probabilidade de vitória, empate e derrota em um confronto entre dois times quaisquer. Para abordar essa questão, é necessário utilizar inicialmente a distribuição de Poisson. Essa distribuição permite calcular as probabilidades de um evento ocorrer X vezes, dado uma média de ocorrências desse evento.

Em outras palavras, por meio da distribuição de Poisson, pode-se estimar as chances de um time marcar n gols em uma partida, desde que se conheça a média de gols marcados por essa equipe. Assim, ao conhecer a média de gols para os times A e B, é possível calcular as probabilidades de cada quantidade de gols e, por meio da análise combinatória, determinar as probabilidades de vitória, empate e derrota.

É importante ressaltar que esse método parte do princípio de que a quantidade de gols marcados por cada time em uma partida é independente da quantidade de gols marcados pelo adversário. Como essa suposição não é completamente verdadeira, o modelo precisa incorporar uma pequena correção nas probabilidades, inflacionando ou deflacionando as probabilidades de empates e consequentemente ajustando as outras probabilidades (vitória e derrota) baseado nessa inflação ou deflação. Este ajuste é feito com base em um valor que pode ser determinado através de experimentações com massas de dados históricos. Essa correção visa ajustar as probabilidades de empate, considerando a diferença entre os empates observados na realidade e aqueles calculados inicialmente, sem a correção. Esse processo permite evitar análises ligeiramente defasadas, e será abordado com maiores detalhes na Seção 5.3.

Com isso, é possível determinar as chances de vitória, empate e derrota para os times em uma partida a partir da média de gols marcados pelos dois times. Porém, determinar

essa média de gols que cada time irá marcar em uma partida específica da maneira mais precisa possível é o grande desafio que constitui a elaboração de um modelo de predição desse gênero.

Para realizar o cálculo dessa média, nosso modelo define dois valores cruciais para cada time, que representam a atuação desses times nas partidas: "ataque", que representa a capacidade de um time de marcar gols, e "defesa", que, contra-intuitivamente, representa a capacidade de um time de sofrer gols. Ou seja:

Ataque Quanto maior for, maior a média de gols esperada para o time marcar (valores maiores são melhores para o time)

Defesa Quanto maior for, maior a média de gols esperada para o time sofrer (valores menores são melhores para o time)

A partir destes valores, podemos fazer cálculos a partir dos coeficientes de Ataque e de Defesa de cada time (exemplo: usar a força de ataque do Time A e a força de defesa do Time B para definir a média de gols esperada pelo Time A, e vice-versa) para estimar suas médias de gols. Uma explicação mais rigorosa de como podermos fazer esse cálculo será apresentada no Capítulo 5.

E por fim, existem vários fatores externos aos times que podem influenciar o resultado de uma partida: jogar em uma cidade com altitude muito elevada, por exemplo, pode prejudicar um time que não está acostumado com tal altitude; ou até fatores mais simples, como jogar fora de casa, possuir algum jogador importante desfalcado, dentre outros. Alguns desses fatores podem ser facilmente incorporados num modelo de predição, enquanto outros podem possuir uma implementação mais difícil ou ter uma significância baixa para a alteração dos resultados. Todos estes também serão abordados no capítulo 5.

4.3 TECNOLOGIAS UTILIZADAS NA SUA IMPLEMENTAÇÃO

Para implementarmos a solução proposta, optamos pelo desenvolvimento de um *script* utilizando a linguagem de programação *Python 3*. *Python* nos dá uma grande flexibilidade na hora de escrever o código, por ser uma linguagem interpretada, e a presença de várias bibliotecas voltadas para o processamento de dados e para estatística, como *NumPy*[2] e *pandas*[3], que foram utilizadas, facilitam a implementação do nosso modelo de predição. Utilizamos também bibliotecas como *matplotlib*[4] para realizar visualizações dos dados gerados.

Por mais que o ideal seja que um modelo desse gênero seja inteiramente automático, alguns fatores podem possuir uma implementação mais difícil, ou serem mais subjetivos,

[2] Disponível em: numpy.org. Acesso em 13/06/2023.
[3] Disponível em: pandas.pydata.org. Acesso em 13/06/2023.
[4] Disponível em: matplotlib.org. Acesso em 13/06/2023.

dificultando sua automatização. Por isso, no *script* desenvolvido constam algumas interações necessárias de serem realizadas pelo operador do modelo. Estas podem ser feitas através da linha de comando no terminal, e são explicadas com mais detalhes no capítulo 5.

Precisamos também armazenar várias informações, como os valores gerados a cada rodada, os coeficientes dos times, resultados passados e previstos, etc. Para isso, utilizamos no *script* uma combinação de arquivos JSON[5], arquivos CSV[6] e arquivos de texto simples.

Com a intenção de tornar mais fácil e dinâmica a interação com o modelo para um possível usuário, desenvolvemos uma página *Web* que disponibiliza as informações calculadas e visualizações dos dados de maneira mais fácil e intuitiva. Para realizar a integração desta página com o programa que implementa o modelo desenvolvido, utilizamos o *Web Framework Flask*[7], que realiza a comunicação direta entre o código do modelo e a página HTML. Esta "ponte" entre o modelo e o usuário também será explicada com mais detalhes no Capítulo 5.

A escolha dessas tecnologias e dessa abordagem nos permitiu um enfoque maior na idealização de funcionamento do modelo e suas ferramentas, uma vez que a facilidade de uso da linguagem e suas bibliotecas auxiliam imensamente na redução da complexidade e na capacitação de aplicações mais robustas.

[5] Disponível em: json.org. Acesso em 13/06/2023.
[6] Comma-separated values
[7] Disponível em: flask.palletsprojects.com. Acesso em 13/06/2023.

5 IMPLEMENTAÇÃO DO MODELO DE PREDIÇÃO

Este capítulo aborda todos os detalhes da implementação utilizados para realizar a predição de partidas e campeonatos de futebol. Dessa forma, conecta os conceitos explicados no capítulo 2, à proposta detalhada no capítulo 4.

5.1 ORGANIZAÇÃO DO CÓDIGO FONTE

O código do projeto está dividido em vários arquivos com diferentes funções. Para facilitar o desenvolvimento e entendimento dos *scripts*, os separamos em diferentes seções. Uma visão geral desta divisão é apresentada nas subseções a seguir.

5.1.1 Código Principal

Corresponde à maior porção do trabalho e consiste em todos os *scripts* responsáveis pela atualização, consulta e cálculo dos dados. Essa etapa é integralmente desenvolvida em Python e divide-se em diversos *scripts*, cada um com suas respectivas responsabilidades.

Os principais *scripts* do projeto são os seguintes, os quais serão detalhados de forma mais específica nas seções seguintes:

automaticUpdating.py
: Esse *script* é responsável por extrair diariamente as informações das partidas ocorridas no dia, e recalcular os valores relativos aos times que jogaram nesse dia.

updating.py
: Similar ao anterior, porém trata-se de uma atualização manual, utilizado quando os dados não são possíveis de serem extraídos automaticamente, por motivos externos ao projeto.

poissonProcess.py
: Esse *script* é responsável por calcular as probabilidades para uma partida específica. Geralmente, é um *script* auxiliar que é importado por outros *scripts* para realizar cálculos necessários sobre uma partida e alcançar os objetivos desejados.

simulacoes.py
: Esse *script* executa o processo de Monte Carlo, explicado na seção 2.2.2, para calcular probabilidades para diferentes métricas desejadas.

Além dos *scripts* mencionados acima, há também outros itens importantes que, embora não sejam cruciais como os anteriores, são extremamente relevantes para o projeto como um todo:

folib.py

Esse *script* representa uma biblioteca criada para o projeto. Nesse arquivo, estão presentes diversas funções utilizadas por vários outros scripts, com o objetivo de centralizá-las em um único local.

writeRanking.py

Esse *script* é responsável por criar um relatório de classificação com base no cenário atual dos times e armazená-lo.

Além dos listados, desenvolvemos vários *scripts* auxiliares que contribuem para a manutenção diária do projeto, que incluem tarefas de coleção de dados para serem salvos com os dados históricos, atualizações dos coeficientes mediante às partidas ocorridas no dia, dentre outros; e também para realizar a ampla gama de cálculos possíveis a partir das informações disponíveis sobre os times.

5.1.2 Armazenamento e manipulação de dados

Para armazenar as informações geradas diariamente, a fim de que estes dados possam ser consultadas por diferentes partes do código para a geração de dados, utilizamos vários arquivos em formatos JSON, CSV e TXT que armazenam diferentes tipos de informação, incluindo:

- Armazenamento das avaliações para cada possível valor de Rho, a fim de utilizarmos o mais otimizado.
- Registro de todas as partidas extraídas.
- Registro das informações calculadas para todos os times, em todas as diferentes datas.
- Registro de todas as probabilidades calculadas para diferentes métricas, em todas as diferentes datas em que o valor tenha sido calculado.
- Classificação dos times para todas as datas em que o *ranking* tenha sido calculado.
- Estruturas de conversão de nomes dos times, uma vez que esses nomes nem sempre são idênticos aos extraídos de diferentes fontes.
- Estruturas responsáveis por salvar informações sobre diferentes campeonatos, como país e nome do torneio, e quantidade de times rebaixados.

5.2 CONFIGURAÇÃO E INICIALIZAÇÃO DOS DADOS

Após a apresentação da organização geral dos *scripts* que constituem o modelo de predição, precisamos pontuar a nível técnico o funcionamento do modelo. A fim de aumentar a flexibilidade de alteração de algumas variáveis, pode-se guardá-las em um arquivo de configuração, o qual pode ser modificado pelo usuário para obter resultados diferentes do padrão. Um exemplo de configuração possível sem alterações feitas pelo usuário pode ser visto no código 1.

Código 1 – Configurações

```
from datetime import datetime

DATE = datetime.now().strftime('%Y-%m-%d'),

TOURNAMENTS = [
    'Brasileiro Serie A',
    'Barclays Premier League',
    'German Bundesliga',
    'Portuguese Liga',
    'Spanish Primera Division',
    'Italy Serie A',
    'French Ligue 1'
]

PERIODO_MME = 17.4 # esse valor e' um exemplo e deve ser
   preenchido pelo usuario
```

Cada time possui valores de ataque, defesa e relevância do mando de campo. Os valores de ataque e defesa de cada equipe são atualizados a cada partida disputada, conforme explicado na Seção 5.4. Já o valor de relevância do mando de campo é atualizado anualmente, sendo o mesmo para todos os times do mesmo país. O cálculo dessa métrica de mando de campo é realizado conforme o código 2.

Código 2 – Extração da relevância do mando de campo

```
from config import PERIODO_MME

def calcula_fator_casa():
    n = int(input("Anos a considerar: "))

    pesos = []
    fator_por_ano = []

    for i in range(n):
        peso = 1 if i == 0 else pesos[i-1] + pesos[i-1] / ((
           PERIODO_MME - 1)/2)
        pesos.append(peso)

        media_casa = float(input("Casa: "))
        media_fora = float(input("Fora: "))
        media = (media_casa + media_fora) / 2
        fator_por_ano.append(media_casa / media)

    resp = sum(peso * fator for peso, fator
               in zip(pesos, fator_por_ano)) / sum(pesos)
    return round(resp - 1, 5)
```

Devido à natureza anual dessa execução, não houve preocupação em automatizar esse processo, e portanto o mesmo é executado e tem seus valores inseridos manualmente. Para preenchê-lo, é necessário informar quantos anos anteriores serão considerados no cálculo (por padrão, utilizamos 5) e as médias de gols marcados pelos times mandantes e visitantes em cada temporada. Por exemplo, se estivermos calculando os valores de mando de campo para o Campeonato Brasileiro de 2023, usando $n = 5$, devemos inserir as médias de gols dos times mandantes e visitantes para as temporadas de 2018, 2019, 2020, 2021 e 2022. Este cálculo utiliza o conceito de média móvel exponencial, explicado na seção 2.1.4 e desenvolvido em detalhes na seção 5.4.

Como veremos também na seção 5.4, toda partida realizada por um time gerará uma modificação de seus valores, os quais serão atualizados diariamente. Porém, para podermos atualizá-los periodicamente, precisamos primeiro que os mesmos existam, e portanto é essencial descrever a criação dos valores de um time. Essa criação, entretanto, pode ser dividida em dois possíveis cenários.

O primeiro cenário é o de inicialização de um campeonato inteiro, onde nunca tivemos quaisquer informações no projeto sobre seus times. Esse cenário, naturalmente, só ocorre

uma vez, e em nosso caso está relacionado à criação do modelo em sua origem, ocorrida em 2020. Para quantificar os valores iniciais de cada time, nesse cenário, não há uma solução objetiva ou única, podendo-se utilizar de outros algoritmos ou dados de casas de apostas para buscar estimativas adequadas para cada time. É esperado que esse começo não apresente necessariamente boas previsões, mas ao longo do tempo quaisquer equívocos existentes serão minimizados, e suas previsões se tornarão cada vez menos dependentes dos valores iniciais escolhidos.

O segundo cenário que deve ser mencionado é o ocorrido anualmente, ao iniciar uma nova temporada. Os times, nesse cenário, estão há meses sem receber atualizações do modelo, uma vez que a temporada anterior terminou e os atletas desfrutaram de um período de descanso. Além disso, novos times, provenientes da segunda divisão nacional, ascenderam à primeira divisão, e devem ser ingressados ao modelo, no lugar dos times que foram rebaixados na temporada anterior. A solução adotada nesses casos envolve considerar o valor de mercado de cada time, o qual pode ser consultado em sites que realizam o acompanhamento do mercado de futebol, como o *Transfermarkt*[1].

É realizada uma regressão linear, onde os eixos são as forças e os valores de mercado de cada time, a fim de calcular a reta que melhor descreve a relação entre os eixos utilizados. Com isso, cada time poderá ter uma nova força a si associada, em função de seu valor de mercado. Para o caso dos times que já existem no modelo, utilizamos como sua nova força para iniciar o campeonato a média aritmética ponderada de sua força final para a temporada anterior e sua força implícita por seu valor de mercado, onde a primeira tem peso 2 e a segunda tem peso 1. Já para os times que ascenderam da divisão inferior, por não terem forças previamente calculadas, apenas adotam a força implícita por seu valor de mercado como sua força inicial.

5.3 CÁLCULO DAS PREDIÇÕES DE RESULTADOS PARA UMA PARTIDA

Com os valores de ataque, defesa e relevância do mando de campo de dois times, podemos calcular o resultado de uma partida entre eles. Seja a variável *"teams"* um dicionário contendo todos os times como chaves e suas respectivas métricas como seus valores, podemos, por exemplo, acessar o valor de ataque do Flamengo conforme o código 3.

Código 3 – Exemplo de acesso a uma métrica

```
teams["FLAMENGO"]["atk"]
```

Os coeficientes de defesa e mando de campo são representadas pelas chaves *"defense"* e *"fieldFactor"*, respectivamente. Para calcular as probabilidades associadas a uma partida,

[1] Disponível em https://www.transfermarkt.com.br/serie-a/startseite/wettbewerb/BRA1. Acesso em 13/08/2023

devemos identificar em qual dos quatro cenários possíveis ela se enquadra:

- Campo neutro: Quando uma partida é disputada em um estádio que não pertence a nenhum dos times envolvidos, sem a presença de uma maioria de torcedores de qualquer time, consideramos que se trata de um campo neutro. Nesse caso, o fator de mando de campo não influencia nos cálculos das probabilidades da partida.

- Clássico estadual: Quando uma partida ocorre entre dois times do mesmo estado, consideramos que o fator de mando de campo deve ser reduzido em relação a um cenário normal, embora ainda deva ser considerado. Isso ocorre porque o time visitante reside próximo ao estádio da partida, o que diminui as adversidades relacionadas a deslocamento, viagem, hospedagem, tempo de preparação técnica, entre outros.

- Mando de campo sem torcida: Algumas situações podem levar a partidas sem a presença de público nos estádios, como ocorreu durante o auge da pandemia de COVID-19 ou quando clubes são punidos por episódios de violência em seus estádios, ficando temporariamente sem torcida. Esse cenário, assim como o de clássicos estaduais, também reduz a relevância do fator de mando de campo, embora mantenha um valor positivo.

- Mando de campo tradicional: A maioria das partidas se enquadra nesse cenário. Caso os cenários específicos mencionados anteriormente não se apliquem à partida em questão, utilizamos o fator de mando de campo tradicional, representado pelo valor *"fieldFactor"* associado aos times.

Uma vez determinado o cenário, realizamos o cálculo da partida. É importante ressaltar que as penalidades atribuídas ao *"fieldFactor"* nos casos de mando de campo sem torcida e clássico estadual são semelhantes, portanto, adotamos um mesmo valor para ambos os casos. Dessa forma, podemos calcular o fator de mando de campo da partida conforme o código 4.

Código 4 – Cálculo do fator mando de campo para uma partida

```
# Mando de campo tradicional
if neutrality == 0:
    home_factor = sqrt(1 + teams[team1]['fieldFactor']) *
       sqrt(1 + teams[team2]['fieldFactor'])
    away_factor = sqrt(1 - teams[team1]['fieldFactor']) *
       sqrt(1 - teams[team2]['fieldFactor'])

# Campo neutro:
elif neutrality == 1:
    home_factor = 1
    away_factor = 1

# Classico estadual / Mando de campo sem torcida
elif neutrality == 2:
    home_factor = max(1, sqrt(1 + teams[team1]['fieldFactor'
       ]) * sqrt(1 + teams[team2]['fieldFactor']) - 0.08)
    away_factor = min(1, sqrt(1 - teams[team1]['fieldFactor'
       ]) * sqrt(1 - teams[team2]['fieldFactor']) + 0.08)
```

Como o valor calculado para o fator mando de campo no Brasileirão 2023 foi de 0,21379 por exemplo, as variáveis *"home_factor"* e *"away_factor"* nesse campeonato assumiriam os valores (1,21379; 0,78621), (1; 1) ou (1,13379; 0,86621) para os casos de *"Mando de campo tradicional"*, *"Campo neutro"* e *"Clássico estadual / Mando de campo sem torcida"* respectivamente.

Com os valores *"home_factor"* e *"away_factor"* calculados, podemos realizar o cálculo do lambda de cada time em uma partida entre *"equipeA"* e *"equipeB"*. O lambda de uma equipe representa a média de gols esperados para o time na partida com suas probabilidades calculadas. Por exemplo, se um time tem um lambda calculado de 1,5 para uma partida, significa que espera-se que esse time marque, em média, 1,5 gols no jogo. Os cálculos são realizados como no código 5.

Código 5 – Cálculo de uma partida

```
lambdaA = teams["equipeA"]["atk"] * teams["equipeB"]["defense
   "] * home_factor
lambdaB = teams["equipeB"]["atk"] * teams["equipeA"]["defense
   "] * away_factor
```

Com os valores de *"lambdaA"* e *"lambdaB"* calculados, aplicamos a Poisson Dupla, conforme introduzido na seção 2.1.2, através do código 6. Assim, obtemos as probabilidades

dos três resultados possíveis (vitória, empate e derrota) da partida. Por fim, aplicamos a Correção Rho, mencionada na seção 2.1.3, detalhada no código 7.

Código 6 – Aplicando a Poisson Dupla

```
import math

probsA = []
probsB = []
winA = draw = winB = 0.0

for i in range(26):
    result = (math.exp(-lambdaA) * lambdaA**i) / math.
       factorial(i)
    probsA.append(result)
    result = (math.exp(-lambdaB) * lambdaB**i) / math.
       factorial(i)
    probsB.append(result)

for i in range(26):
    for j in range(26):
        if i > j:
            winA += probsA[i] * probsB[j]
        if i < j:
            winB += probsA[i] * probsB[j]
        if i == j:
            draw += probsA[i] * probsB[j]
```

Código 7 – Aplicando a Correção Rho

```
rho = get_rho()

draw_increase = rho * draw
non_draw = winA + winB
draw_decrease = draw_increase / non_draw

draw += draw_increase
winA -= draw_decrease * winA
winB -= draw_decrease * winB
```

A função "`get_rho()`" utiliza o conceito de *Backtest*, também apresentado na seção

2.1.3. Nessa função, utilizamos todo o histórico de partidas extraídas até o momento para identificar o valor de Rho mais adequado a ser utilizado, ou seja, aquele valor que melhor aproxima o número de empates que ocorreram com o número médio de empates calculados com cada valor de Rho.

Por exemplo, em 28/06/2023, o valor de Rho ótimo está em 0,07, com um total de 5.120 partidas, das quais 1.302 terminaram em empate. Se calcularmos o somatório de todas as probabilidades de empate utilizando o valor $\rho = 0,07$, obtemos um total de empates esperados de 1.304,78, o que é o mais próximo de 1302 entre os valores de Rho calculados, com uma precisão de duas casas decimais. Se não estivéssemos utilizando Rho (Rho = 0), por exemplo, a quantidade média esperada de empates seria de 1.219,42, um valor que destoa da realidade. Portanto, após esse ajuste final realizado pela correção Rho, temos as probabilidades dos três resultados possíveis para a partida calculados.

5.4 ATUALIZAÇÃO DOS DADOS

Ao final de cada dia, após a realização de pelo menos uma partida, é executado o *script* denominado "automaticUpdating.py". Seu objetivo é extrair os dados necessários de todas as partidas ocorridas nesse dia e atualizar os valores de cada time que tenha jogado. Caso essa extração falhe, pode-se utilizar o *script* "updating.py", o qual tem o mesmo objetivo, porém o realiza de forma manual.

A obtenção automática dos dados é realizada utilizando a API do (FIVE..., 2023). Essa API retorna os dados desejados em formato CSV, que é transformado em um *DataFrame*[2] utilizando a função "read_csv" do pacote *pandas*. Nesse contexto, o *DataFrame* facilita a representação e o processamento dos dados obtidos, tornando-os mais acessíveis para as etapas subsequentes do projeto. A extração dos dados é feita de maneira simples seguindo o código 8.

Código 8 – Extração das partidas do dia

```
import pandas as pd
from config import DATE, TOURNAMENTS

df = pd.read_csv(
    'https://projects.fivethirtyeight.com/'
    'soccer-api/club/spi_matches_latest.csv'
)
df = df.loc[df['date'] == DATE]
df = df.loc[df['league'].isin(TOURNAMENTS)]
```

[2] Estrutura de dados organizada em linhas e colunas, permitindo a manipulação e análise dos mesmos de forma eficiente. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

No código 8, realizamos uma consulta à API para obter os dados desejados. Em seguida, filtramos apenas as partidas que ocorreram na data em questão e que pertencem aos campeonatos conhecidos pelo modelo de predição.

Após a extração dos dados das partidas, podemos prosseguir com a obtenção dos valores que serão utilizados posteriormente para avaliar o desempenho dos times. O código 9 exemplifica essa extração. Nele, vemos que cada elemento da lista "*matches*" corresponde a um dicionário que atribui a cada time um valor. Esse valor corresponde a média aritmética simples entre as 4 métricas utilizadas, como detalhado na seção 3.2: gols, gols ajustados, gols esperados baseados em finalizações e gols esperados não baseados em finalizações.

Código 9 – Extração dos valores das partidas

```
matches = []

for _, row in df.iterrows():
    matches.append(
        {
            row['team1']: round((
                row['score1'] + row['xg1'] +
                row['nsxg1'] + row['adj_score1']
            )/4.0, 4),
            row['team2']: round((
                row['score2'] + row['xg2'] +
                row['nsxg2'] + row['adj_score2']
            )/4.0, 4)
        }
    )
```

Com os valores de cada partida em mãos, é possível associá-los aos valores esperados para cada partida. Para isso, serão utilizadas as informações de ataque e defesa de cada time, bem como o peso do mando de campo, a fim de estimar o placar esperado para cada partida ocorrida. O trecho do código 10 ilustra como é calculado o placar estimado para uma partida.

Código 10 – Estimativa das partidas

```
homeFactor = (
    sqrt(1 + teams[team1]['fieldFactor']) *
    sqrt(1 + teams[team2]['fieldFactor'])
)
awayFactor = (
    sqrt(1 - teams[team1]['fieldFactor']) *
    sqrt(1 - teams[team2]['fieldFactor'])
)

lambdaA = (
    teams[team1]['atk'] *
    teams[team2]['defense'] *
    homeFactor
)
lambdaB = (
    teams[team2]['atk'] *
    teams[team1]['defense'] *
    awayFactor
)
```

Com as informações sobre o que ocorreu em cada partida e o que era previsto ocorrer, é possível realizar uma comparação que indique a direção na qual os valores de ataque e defesa de cada time devem ser atualizados. Por exemplo, se um time era esperado marcar em média 1,5 gols, mas produziu o equivalente a 1,9 gols na partida, será necessária uma atualização positiva no valor de seu ataque. No código 11, apresentamos um trecho de código que representa essa atualização. Nele, as variáveis a serem consideradas são as seguintes:

realA

Valor calculado pela média das métricas extraídas da API do (FIVE..., 2023) para o time mandante.

realB

Valor calculado pela média das métricas extraídas da API do (FIVE..., 2023) para o time visitante.

lambdaA, lambdaB

Valores calculados no trecho de código apresentado anteriormente, representando as médias estimadas de produção de gols para cada time na partida em questão.

Código 11 – Atualização dos valores

```
from config import PERIODO_MME as N

calculadoAtkA = sqrt(realA/lambdaA) * teams[team1]['atk']
calculadoDefA = sqrt(realB/lambdaB) * teams[team1]['defense']
calculadoAtkB = sqrt(realB/lambdaB) * teams[team2]['atk']
calculadoDefB = sqrt(realA/lambdaA) * teams[team2]['defense']

teams[team1]['atk'] = (
    (N - 1)/(N + 1) * teams[team1]['atk'] + 2/(N + 1) *
       calculadoAtkA)
teams[team1]['defense'] = (
    (N - 1)/(N + 1) * teams[team1]['defense'] + 2/(N + 1) *
       calculadoDefA)

teams[team2]['atk'] = (
    (N - 1)/(N + 1) * teams[team2]['atk'] + 2/(N + 1) *
       calculadoAtkB)
teams[team2]['defense'] = (
    (N - 1)/(N + 1) * teams[team2]['defense'] + 2/(N + 1) *
       calculadoDefB)
```

A atualização dos valores de ataque e defesa de cada time segue uma média aritmética ponderada, com pesos de acordo com o período da média móvel exponencial em uso, conforme visto na seção 2.1.4. Assim, utiliza-se um peso $\frac{N-1}{2}$ vezes maior para o valor atual da métrica do que para o novo valor calculado. Como esses valores também devem somar 1, por serem os pesos de uma média aritmética ponderada, temos que os valores $\frac{N-1}{N+1}$ e $\frac{2}{N+1}$ são utilizados respectivamente para representar os pesos do valor atual da métrica e para o novo valor calculado. Dessa forma, estes pesos somados resultam em 1 e o primeiro peso tem valor $\frac{N-1}{2}$ vezes maior que o segundo.

5.5 SIMULAÇÃO DO RESULTADO FINAL DE CAMPEONATOS

Neste ponto, já somos capazes de realizar o ciclo iterativo entre "simular os resultados das partidas utilizando os coeficientes calculados" e "atualizar os coeficientes baseados nos resultados reais". Além de prever os resultados das partidas, porém, temos muitas métricas que desejamos extrair e prever para cada um dos times, como explicado na motivação e na seção 4.1. São elas:

- Chance de ser campeão

- Chance de ser rebaixado[3]
- Quantidade média de pontos obtidos
- Quantidade mínima e máxima de pontos obtidos
- 5 métricas que representam a chance de ser G4 a G9[4]

Para realizar uma simulação do campeonato, utilizamos o método de Monte Carlo, explicado na seção 2.2.2. Este método consiste na geração de uma alta quantidade de valores aleatórios para a obtenção das métricas desejadas. Para aplicarmos este método em um modelo de predição de partidas de futebol, definimos o seguinte fluxo lógico:

1. Para cada partida que ainda não aconteceu no campeonato, são gerados dois valores aleatórios, com peso baseado nas probabilidades de gol de cada time. Estas serão as quantidades de gols de cada time na partida.
2. Dependendo da quantidade de gols, é determinado o resultado das partidas entre vitória, empate ou derrota.
3. A pontuação do campeonato é atualizada conforme o resultado de cada partida simulada.

Este fluxo compõe apenas uma simulação do campeonato. Por Monte Carlo, porém, precisamos de uma grande massa de dados. Por isso, este processo é aplicado ao longo de um grande número de repetições; para nossas experimentações, utilizamos 20.000 iterações a cada simulação de campeonato. Ao final de cada iteração, as métricas desejadas, explicitadas anteriormente, são coletadas, e ao final da simulação estas métricas são disponibilizadas ao usuário e salvas em um arquivo para análises seguintes.

Um detalhe importante de ser explicitado das métricas é o significado de "chance" em métricas como chance de ser campeão, chance de ser rebaixado, etc. Estas métricas não representam a probabilidade deste evento ocorrer no mundo real, mas sim a porcentagem de vezes que este evento foi observado ao decorrer de N iterações do processo de simulação. Como o modelo de Monte Carlo sugere, essa porcentagem observada, dado um grande número de simulações, tende a se aproximar da probabilidade real.

O código principal que executa cada simulação é bem simples. Primeiro, são inicializadas as variáveis que contém as opções da simulação, e são carregadas em memória os times daquele campeonato.

[3] No Brasileirão, por exemplo, os 4 times com menor colocação ao final do campeonato são rebaixados para a série inferior no campeonato seguinte. A quantidade de times rebaixados pode variar dependendo do campeonato observado.

[4] Gn significa estar entre os n times com maior colocação no campeonato

O objeto "*teams*", que contém todos os times do campeonato escolhido, é inicializado com todos os times com valores padrão. Esse objeto é um dicionário, onde as chaves são os nomes dos times e seus valores são as variáveis associadas ao time, como "*atk*", "*defense*" e "*fieldFactor*"

Na maioria dos casos, o campeonato que estamos analisando já está em andamento. Por isso, é importante que os resultados reais das partidas que já ocorreram sejam atualizados antes da simulação ocorrer, ambos para aumentar a eficiência do código, reduzindo retrabalho, e para obtermos uma melhor representação da situação atual do campeonato.

Código 12 – *Loop* principal de uma simulação

```
def main():
    updateMatches()
    for i in range(NUM_SIMULS):
        for team in teams:
            teams[team]['pontos'] = teams[team]['
               pontos_iniciais']
        playGroup()
        updateMetrics()
```

No código 12, a função "*updateMatches*" é responsável por atualizar o objeto "*teams*" com os dados mais atuais do campeonato, incluindo as partidas que já aconteceram. Após isso, em cada iteração do campeonato, os valores são reiniciados com os valores iniciais, e as partidas remanescentes são calculadas na função "*playGroup*", que roda o campeonato até o final.

Código 13 – Cálculo do placar de uma partida simulada

```
def poisson(A, B, importanceA, importanceB):
    lambdaA, lambdaB = findLambda(A, B, importanceA,
       importanceB)
    P1 = []
    P2 = []
    for i in range(26):
        result = (e**(-1*lambdaA) * lambdaA**i) / (fat(i)
           *1.0)
        P1.append(result)
        result = (e**(-1*lambdaB) * lambdaB**i) / (fat(i)
           *1.0)
        P2.append(result)
    golsA = draftGoal(P1)
    golsB = draftGoal(P2)
    return golsA, golsB
```

As probabilidades para cada placar das partidas são calculadas assim como são calculadas normalmente no resto do modelo, usando o método de Poisson Duplo, e salvas em uma lista. A partir destas probabilidades, em seguida, são determinados os números de gols que cada time fez na partida simulada de maneira aleatória, tendo as probabilidades como peso. Após determinar o resultado final da partida, a pontuação dos times é atualizada conforme se houve empate ou vitória do Time A ou Time B. Este processo constitui o cálculo de uma partida, e o mesmo é repetido para todas as partidas restantes do campeonato. Tal processo é explicitado no código 13.

Um detalhe importante é a consideração dos critérios de desempate: times podem ficar com a mesma quantidade de pontos ao final de um campeonato. Para determinar um desempate, campeonatos diferentes usam métricas diferentes, mas algumas são mais comuns que as outras, e existem casos onde mesmo após a aplicação de um critério de desempate, dois times podem continuar empatados, criando a necessidade da definição de múltiplos critérios. Um dos principais critérios de desempate é a quantidade de vitórias obtidas pelo time. Outro critério de desempate comumente utilizado, mas com menos prioridade do que o número de vitórias, é o saldo de gols, que é calculado como o total de gols marcados subtraído do total de gols sofridos. Ambos estes critérios são tratados no código através de uma pequena otimização, onde os dois são calculados de maneira embutida na pontuação de cada time, na hora de calcular a pontuação a ser adicionada no final da partida. Este processo é mostrado no código 13, no qual há vitória do Time A e derrota do time B.

Código 14 – Exemplo de pontuação na simulação com Time *A* vencedor

```
if golsA > golsB:
    teams[A.upper()]['pontos'] += 3 + 0.0001*(golsA-golsB) +
       0.00000001*golsA
    teams[B.upper()]['pontos'] += 0.01 + 0.0001*(golsB-golsA)
        + 0.00000001*golsB
```

Por fim, na função "*updateMetrics*", as métricas de interesse são atualizadas para cada time antes da próxima iteração, onde os valores são reinicializados. Ao final do processamento da simulação, após as N iterações, os resultados com as métricas são salvos em um arquivo TXT, que inclui a data atual no nome do arquivo para facilitar o arquivamento dos resultados. As informações são salvas para serem fáceis de serem lidas rapidamente, mas podem ser interpretadas ou até transformadas em um arquivo CSV para geração de análises e visualizações gráficas.

5.6 OTIMIZAÇÕES

Na seção 5.3, é detalhado todo o processo de como calcular as probabilidades de uma partida a partir dos valores armazenados de cada time. De forma resumida, podemos descrever que o processo consiste em primeiro calcular os valores *lambdaA* e *lambdaB* a partir das informações de cada time e de mando de campo, e a partir do *lambda* de cada equipe calcular as probabilidades da partida utilizando o método de Poisson Duplo.

Porém, foi observada uma oportunidade de otimização do *script* ao perceber que nem toda partida é jogada com ambos os times completamente disponíveis. Há partidas em que uma equipe pode estar, por exemplo, utilizando apenas os jogadores reservas, ou com vários jogadores titulares lesionados, ou ainda suspensos. Ou seja, não é raro observarmos cenários de partidas em que as condições normais de força não são bem fiéis à realidade daquele confronto.

É natural que, em ocasiões como essa, queiramos aplicar algum tipo de penalidade ao time com maiores problemas, e dessa forma foi idealizada a possibilidade de alterarmos ligeiramente os *lambdas* calculados, a fim de buscar um maior realismo nesses casos.

Vale destacar que a utilização dessa otimização é subjetiva e cabe ao usuário adaptar algumas variáveis a seu gosto pessoal. Isto acontece uma vez que o principal objetivo dessa etapa consiste em penalizar os times que estejam em um cenário fora do ideal para a partida, mas sem uma definição clara da "magnitude" da penalidade. Uma implementação deste cálculo de penalidade pode ser vista no código 15.

Código 15 – Função de otimização

```
def optimize_lambdas(points_deficit_A, points_deficit_B,
    lambda_A, lambda_B, penalty_factor):
    if penalty_factor < 1:
        raise Exception("Penalty factor should be at least 1.
            ")
    min_points_deficit = min(points_deficit_A,
        points_deficit_B)
    points_deficit_A -= min_points_deficit
    points_deficit_B -= min_points_deficit

    if points_deficit_A:
        lambda_A *= (100 - points_deficit_A) / 100 +
            points_deficit_A / (100 * penalty_factor)
        lambda_B *= (points_deficit_A * penalty_factor -
            points_deficit_A + 100) / 100
    elif points_deficit_B:
        lambda_A *= (points_deficit_B * penalty_factor -
            points_deficit_B + 100) / 100
        lambda_B *= (100 - points_deficit_B) / 100 +
            points_deficit_B / (100 * penalty_factor)

    return lambda_A, lambda_B
```

Dois pontos de subjetividade são possíveis de se identificar: o primeiro sendo o valor de *penalty_factor*, o qual deve ser maior ou igual a 1 (o valor igual a 1 resulta em um impacto nulo), e mensura o impacto das penalidades a serem aplicadas. O segundo trata da maneira de quantificar as variáveis de *points_ deficit* de cada time. Alguns dos fatores que podem ser considerados na definição desses pontos de penalidade são ausências de jogadores relevantes na partida e diferentes graus de importância da partida para as equipes envolvidas. Entretanto, cabe mais uma vez ao usuário definir, caso seja sua vontade, suas regras de utilização dessa otimização. As regras definidas e utilizadas nas experimentações observadas no capítulo 6 são detalhadas na seção 6.2.

Esse cálculo de otimização dos *lambdas* pode ser realizado, caso desejado, após o cálculo original de cada *lambda*, servindo apenas para realizar um pequeno ajuste e fortalecer suavemente o lado menos desfalcado. Esta otimização, apesar de opcional e subjetiva, se apresenta como um dos diferenciais do projeto, e dependendo das condições que permeiem uma partida, pode se mostrar como bastante útil e poderosa.

5.7 INTERFACE WEB

Como foi mencionado no capítulo 4, facilitar a visualização dos dados e métricas geradas é de grande utilidade ao usuário do *script*. Para efetuar essa interação com os dados, foi desenvolvida uma página web utilizando o *framework web Flask*. Além do *script* "app.py", que utiliza *Flask*, há arquivos HTML correspondentes a cada página individualmente.

Portanto, o *script* em Python é responsável por coletar as informações necessárias para cada página HTML e fornecer os dados já organizados para serem exibidos. Atualmente, a página web está disponível na internet[5].

5.7.1 Página inicial

A página inicial permite aos usuários consultar as probabilidades de uma partida qualquer, e é uma das principais funções que compõe este trabalho. Através dela, é possível obter as previsões para um embate entre dois times desejados, na qual as informações são calculadas assim como explicado na seção 5.3.

Um exemplo do uso desta página inicial pode ser observado na figura 1, na qual são calculados um placar médio de 1,654 gols para o time Fluminense e 1,458 para o time Flamengo, resultando em uma probabilidade de vitória do Fluminense de 41,47%, de empate de 25,33%, e de vitória do Flamengo de 33,20%. Além disso, são disponibilizadas algumas informações para utilização em casas de aposta, como as "*odds*".

Para realizar o cálculo de uma partida usando esta página, selecione os times desejados, mantendo em mente que o time à esquerda (Time 1) é o time que possui o mando de campo. Com os times definidos, indique se a partida é uma partida normal, possui mando neutro ou é um clássico entre os times, como explicado na seção 5.3. Opcionalmente, podem ser definidos os desfalques para cada um dos times, caso existam, como explicado na seção 5.6.

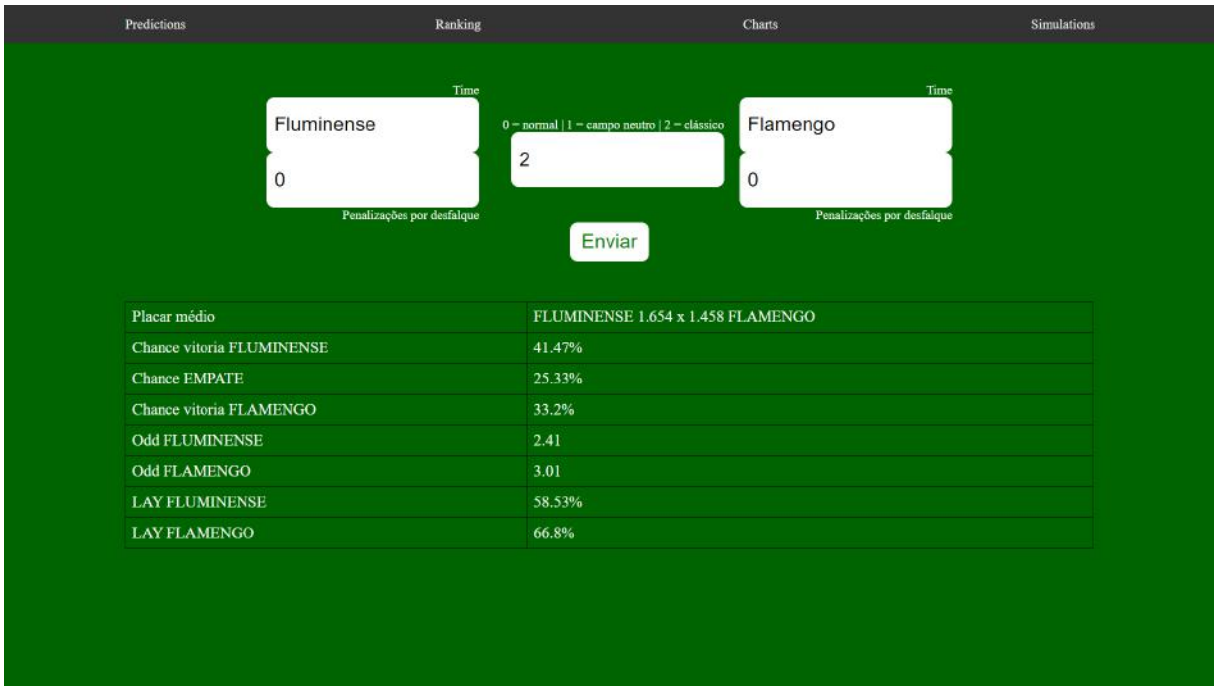
5.7.2 Ranking

A página seguinte é a página que mostra o *ranking* dos times, de acordo com o modelo. Nela, é possível visualizar os diferentes coeficientes utilizados internamente pelo modelo, como os coeficientes descritos neste capítulo. Na figura 2, podemos observar como a tabela estava no dia 16/07/2023. Caso seja desejado obter a tabela de datas mais antigas, é possível selecionar a data desejada em uma seleção na página.

Um detalhe é que no começo da página todos os times são agrupados juntos, mas os coeficientes de um time só podem ser considerados em relação aos coeficientes de outros times que disputam o mesmo campeonato. Isso também acontece com cálculos

[5] https://futebol-onisciente.com/

Figura 1 – Página inicial do website



Exemplo de dados calculados para partida entre Fluminense e Flamengo. Note que, por esta partida ser um clássico do Rio de Janeiro, aplicamos o fator de mando de campo como tipo 2.

de predições de partidas entre times que não se enfrentariam normalmente (por exemplo, Flamengo x Real Madrid, dois times de regiões diferentes), que podem não ser condizentes com a realidade. A tabela com os times separados por campeonato pode ser encontrada abaixo da tabela principal.

Na tabela, são listados parâmetros para cada time, os quais são calculados pelo nosso modelo. São eles:

SPI (*Soccer Power Index*) SPI é uma métrica criada pelo Five... (2023) para avaliar o desempenho de times de futebol. Seu valor é calculado como "a porcentagem do total de pontos que um time faria[6] caso enfrentasse todos os outros times". Adotamos duas fórmulas diferentes para essa métrica, a fim de avaliar de maneira diferente a definição de "todos os outros times". Enquanto o "SPI Default" considera uma força média, que busca generalizar a força de um time médio do mundo, o "SPI Round-Robin" considera que os times existentes são apenas as conhecidas pelo modelo, e dessa forma o cálculo pode ser realizado ao simular cada time enfrentando todas as outras, e extraindo com isso seu desempenho médio.

ATK Coeficiente de ataque do time, como calculado pelo modelo.

[6] 3 pontos para vitória, 1 ponto para empate, 0 pontos para derrota

Figura 2 – *Power Ranking* dos times

#	Club	SPI Default	SPI Round-Robin	Power	ATK	DEF	OFF	Power Ranking
1	MANCHESTER CITY	92.873	80.288	7.329	3.047	0.416	2.533	100.00
2	BARCELONA	90.824	77.140	6.467	2.776	0.429	2.384	95.02
3	BAYERN MUNCHEN	91.650	76.390	5.427	3.236	0.596	3.859	88.06
4	ARSENAL	89.731	73.643	4.975	2.975	0.598	3.558	84.60
5	LIVERPOOL	89.093	72.884	4.889	2.880	0.589	3.394	83.91
6	REAL MADRID	88.285	72.061	4.835	2.750	0.569	3.129	83.46
7	NEWCASTLE	87.348	71.003	4.716	2.631	0.558	2.936	82.47
8	DORTMUND	90.160	73.238	4.651	3.206	0.689	4.420	81.92
9	ATL MADRID	87.246	70.571	4.572	2.668	0.584	3.114	81.24
10	REAL SOCIEDAD	83.255	67.030	4.389	2.154	0.491	2.114	79.61
11	BENFICA	84.817	68.097	4.347	2.376	0.547	2.598	79.23
12	NAPOLI	82.752	66.077	4.158	2.168	0.522	2.262	77.46
13	MANCHESTER UNITED	85.134	67.749	4.136	2.508	0.606	3.041	77.26
14	BRIGHTON	86.687	68.881	4.128	2.768	0.671	3.713	77.18
15	LEIPZIG	83.706	66.571	4.089	2.320	0.567	2.633	76.80
16	INTER	83.991	65.933	3.827	2.481	0.648	3.218	74.17
17	ASTON VILLA	80.923	63.760	3.797	2.082	0.548	2.283	73.85
18	SPORTING	80.124	62.404	3.550	2.093	0.590	2.469	71.19
19	VILLARREAL	82.860	64.116	3.545	2.467	0.696	3.434	71.13
20	PORTO	78.612	60.922	3.404	1.985	0.583	2.314	69.51
21	BRENTFORD	79.329	61.088	3.339	2.104	0.630	2.652	68.74
22	TOTTENHAM	81.911	62.224	3.255	2.516	0.773	3.889	67.73
23	MILAN	77.815	59.568	3.190	2.004	0.628	2.517	66.93

Tabela de *ranking* dos times gerada pelo modelo desenvolvido, atualizada em 16/07/2023.

DEF Coeficiente de defesa do time, como calculado pelo modelo.

Power Uma métrica usada para rapidamente avaliar o poder de um time. Power $= \frac{\text{ATK}}{\text{DEF}}$. Geralmente, quanto maior o valor de "*Power*" de um time, mais forte será esse time.

OFF Métrica de "ofensividade" do time. $\text{OFF} = 2 \cdot \text{ATK} \cdot \text{DEF}$. Usado para avaliar o número de gols que ocorrem em partidas deste time, sejam contra ou a favor: em jogos com times que possuem maior ofensividade, ocorrem mais gols totais em média.

Power Ranking Essa métrica normaliza, para o campeonato em questão, os valores de "power" das equipes, de modo a classificar o time com maior "power" com o valor 100, enquanto o time com o menor valor recebe a avaliação 0. Dessa forma, podemos facilmente mensurar a força de um time em relação aos demais daquele campeonato.

Performance Após realizar os cálculos dos valores de força normalizados, explicitados na métrica "*power ranking*", utilizamos a fórmula de normalização adotada para expressar um valor que represente a performance que cada time está apresentando até então na temporada atual. Dessa forma, podemos ter performances acima do valor 100, ou seja, indicando que o time está até o momento performando acima do esperado para o atual melhor time, assim como podemos obter valores abaixo de 0, indicando uma performance abaixo do esperado para o pior time atualmente observada no torneio.

Expected Points Assim como xG representa a "quantidade esperada de gols" para uma situação, "*Expected Points*" representa a "quantidade esperada de pontos" que um time fez no campeonato dadas as probabilidades de vitória, empate e derrota, ou seja: $\text{xP} = 3 \cdot \text{chance de vitória} + 1 \cdot \text{chance de empate} + 0 \cdot \text{chance de derrota}$. Essas probabilidades, porém, são calculadas a partir dos desempenhos obtidos nas partidas até então realizadas pelo time no campeonato, e não possuem qualquer relação com as probabilidades que podemos calcular previamente para as partidas.

Desses parâmetros, "SPI" e "Expected Points" são os únicos conceitualmente existentes em outros projetos e ideias, ainda que sejam calculados por nosso projeto. Enquanto o "SPI" busca equivaler à métrica adotada pelo Five... (2023), o conceito expresso pelo "Expected Points" é análogo à ideia de "Expected Goals", ou seja, se propõe a quantificar a pontuação justa que o time em questão deveria ter no campeonato até o momento de seu cálculo.

5.7.3 *Charts*

Esta página, chamada de *Team Comparison*, permite aos usuários selecionar times e visualizar gráficos relacionados às suas forças estimadas pelo modelo ao longo do tempo, além do desempenho de cada time no campeonato atual, de forma comparativa e individual. Os gráficos são gerados usando a biblioteca "*matplotlib*", mencionada na seção 4.3, e são servidos ao usuário em forma de imagem.

Para usar esta página, deve-se selecionar um ou mais times na lista, que estão agrupados por país. Opcionalmente, é escolhido se os gráficos serão gerados de maneira normalizada (opção "*normalized*"), para realizar comparações mais relativas entre os times ao longo da temporada, ou se serão gerados com os valores absolutos (opção "*real*") de cada time. Além disso, é possível escolher a rodada inicial e final dos gráficos gerados. O eixo Y representa o "*Power*" dos times, mesma métrica explicada na seção 5.7.2, e o eixo X representa a rodada do campeonato.

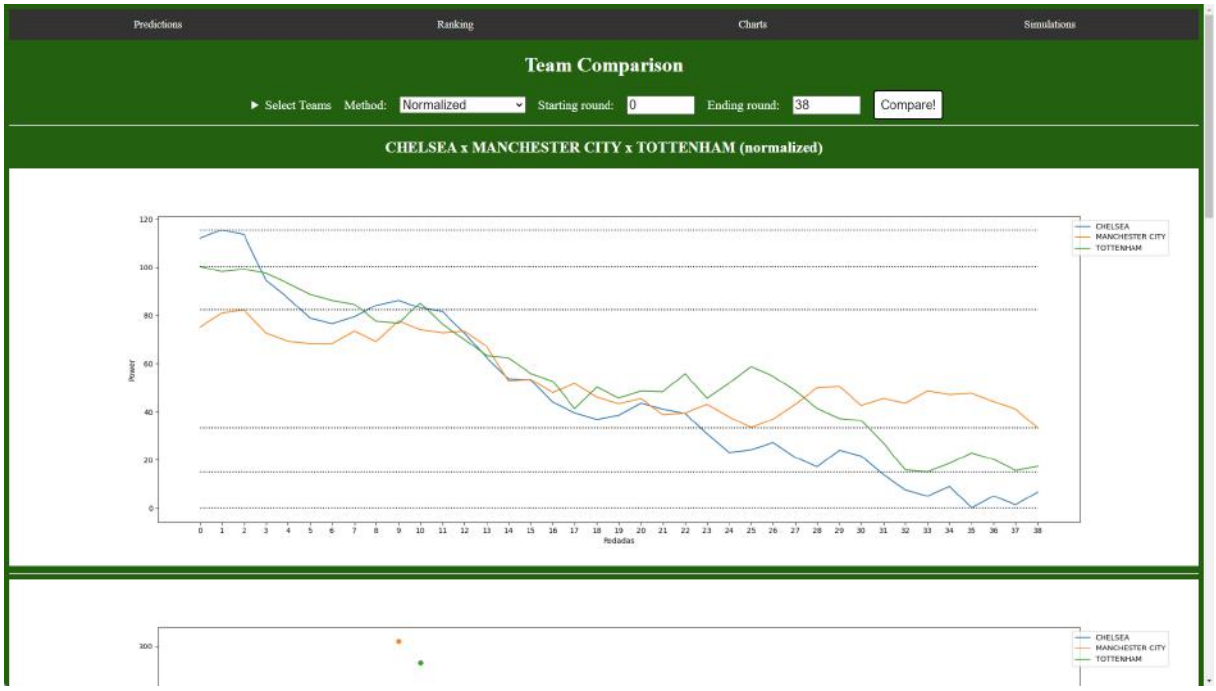
Note que na visualização normalizada o eixo Y deixa de representar os valores reais de "*Power*", e passa a ser um valor relativo, onde o valor mais baixo de todos se torna zero. Dessa forma, podemos ver com maior clareza a variação total da métrica. Caso o valor máximo seja X, saberemos que esse valor, na escala real, é $X\%$ maior que o valor mínimo alcançado em outro momento.

Um exemplo de gráficos gerados pela comparação normalizada entre três times do campeonato inglês *Premier League* pode ser visto na figura 3. A mesma comparação, mas com valores absolutos, pode ser observada na figura 4. Enquanto o gráfico com valores normalizados facilita a visualização do quanto a força de cada time varia, o gráfico com valores absolutos passa com maior clareza a verdadeira força de cada time ao longo do

tempo, demonstrando que a escolha do tipo de gráfico a ser gerado está diretamente relacionado com o tipo de informação que o usuário deseja extrair ao visualizá-lo.

Após o gráfico mostrado nessas figuras, é mostrado o mesmo gráfico mas com vários pontos ao longo do eixo das rodadas, que representam a atuação do time naquela rodada. Com isso, fica fácil visualizar situações nas quais um time performou melhor ou pior que o esperado. Um exemplo desta visualização pode ser visto na figura 5. Ao longo do resto da página, são mostrados gráficos iguais aos dois primeiros, mas separando cada time individualmente.

Figura 3 – Exemplo de página de comparação relativa entre times

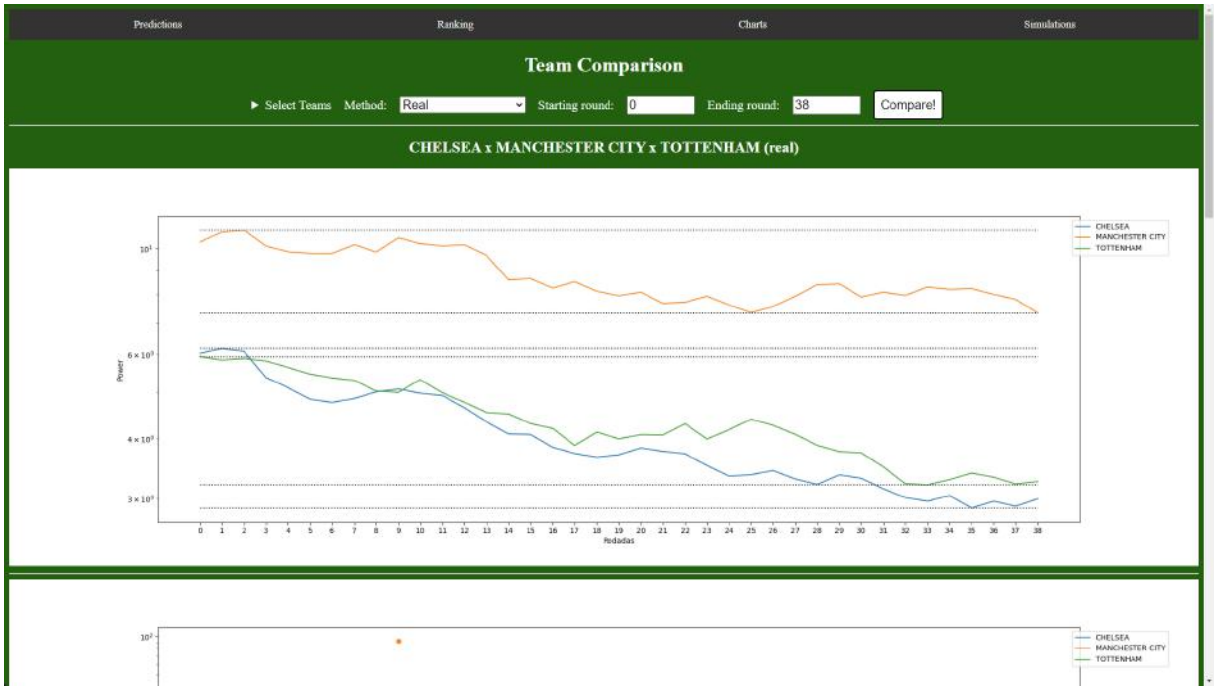


Comparação normalizada entre os times *Chelsea*, *Manchester City* e *Tottenham*, em relação ao valor de *Power* de cada time ao longo da temporada.

5.7.4 Simulations

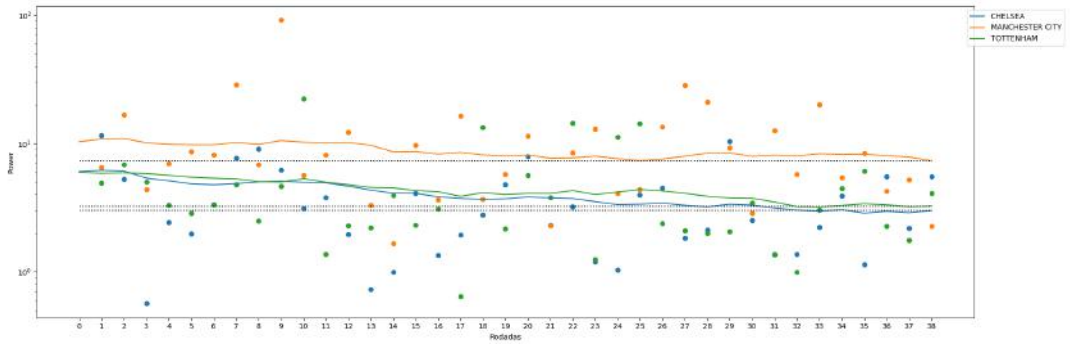
Por último, a página *Simulations* contém um agregado das informações calculadas pela parte do código de simulações, detalhado na seção 5.5, distribuídas em várias tabelas. Nela, é possível visualizar os dados de simulação de vários campeonatos; por padrão, são mostrados os dados da rodada mais recente do campeonato escolhido, seja este em andamento ou finalizado, mas também é possível visualizar os dados de simulação de datas específicas, escolhidas através de um campo de seleção de data, assim como na página que mostra o *ranking* dos times. Para utilizar esta página, basta escolher um campeonato dentre as opções disponíveis. Um exemplo dela pode ser observado na figura 6.

Figura 4 – Exemplo de página de comparação absoluta entre times



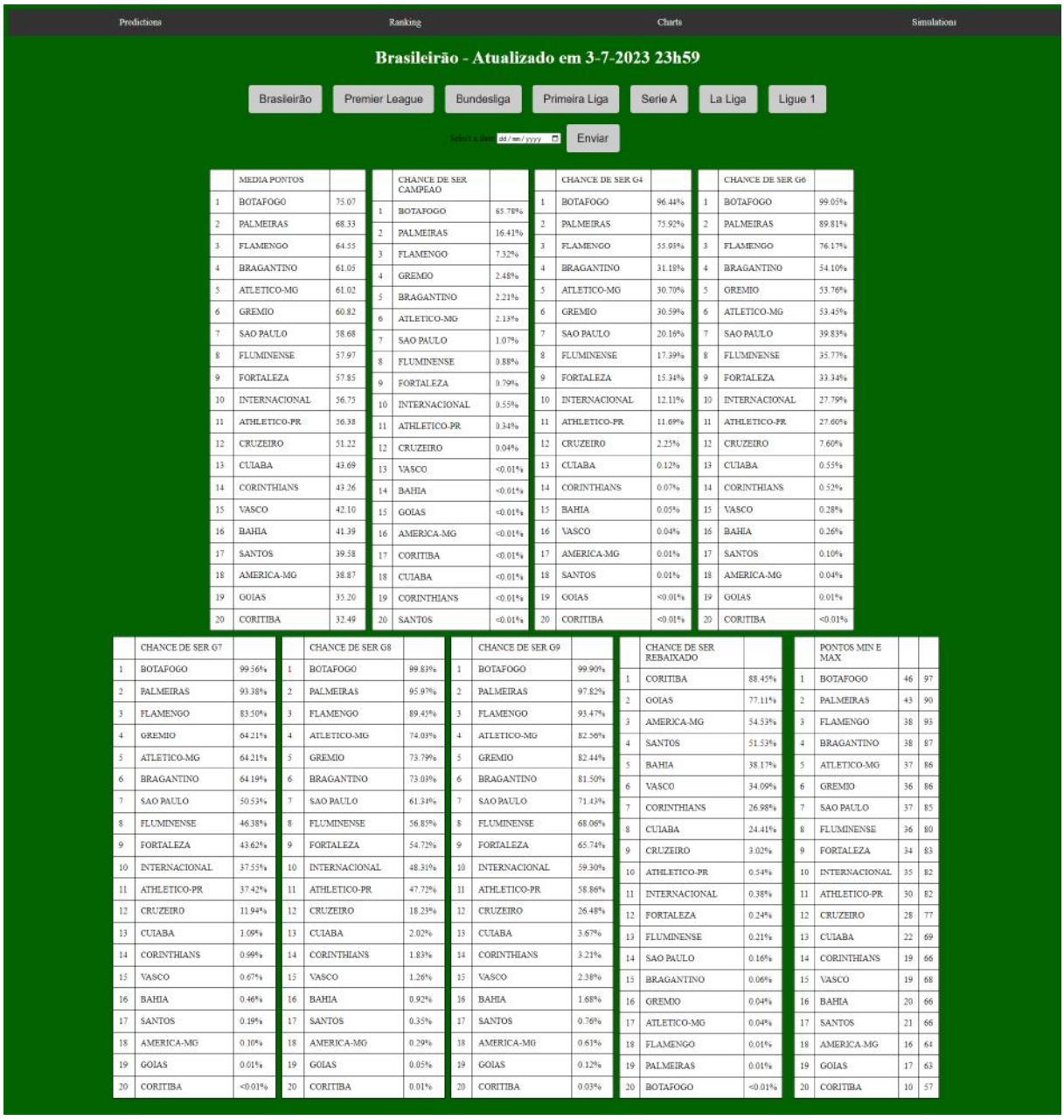
A mesma comparação da figura 3 entre os times *Chelsea*, *Manchester City* e *Tottenham*, mas feita em relação ao valor absoluto de *Power* de cada time. Como é possível observar, *Manchester City* é um time muito acima dos outros dois times dessa comparação.

Figura 5 – Exemplo do gráfico de comparação absoluta entre os times com os dados das partidas



A mesma comparação da figura 4, feita com os valores absolutos, mas desta vez com os pontos que representam a performance dos times nas partidas ao longo das rodadas do campeonato.

Figura 6 – Exemplo de página de simulações



	MEDIA PONTOS	
1	BOTAFOGO	75.07
2	PALMEIRAS	68.33
3	FLAMENGO	64.55
4	BRAGANTINO	61.05
5	ATLETICO-MG	61.02
6	GREMIO	60.82
7	SAO PAULO	58.68
8	FLUMINENSE	57.97
9	FORTALEZA	57.85
10	INTERNACIONAL	56.75
11	ATHLETICO-PR	56.38
12	CRUZEIRO	51.22
13	CUIABA	43.69
14	CORINTHIANS	43.26
15	VASCO	42.10
16	BAHIA	41.39
17	SANTOS	39.58
18	AMERICA-MG	38.87
19	GOIAS	35.20
20	CORITIBA	32.49

	CHANCE DE SER CAMPEAO	
1	BOTAFOGO	65.78%
2	PALMEIRAS	16.41%
3	FLAMENGO	7.32%
4	GREMIO	2.48%
5	BRAGANTINO	2.21%
6	ATLETICO-MG	2.13%
7	SAO PAULO	1.07%
8	FLUMINENSE	0.88%
9	FORTALEZA	0.79%
10	INTERNACIONAL	0.55%
11	ATHLETICO-PR	0.34%
12	CRUZEIRO	0.04%
13	VASCO	<0.01%
14	BAHIA	<0.01%
15	GOIAS	<0.01%
16	AMERICA-MG	<0.01%
17	CORITIBA	<0.01%
18	CUIABA	<0.01%
19	CORINTHIANS	<0.01%
20	SANTOS	<0.01%

	CHANCE DE SER G4	
1	BOTAFOGO	96.44%
2	PALMEIRAS	75.92%
3	FLAMENGO	55.93%
4	BRAGANTINO	31.18%
5	ATLETICO-MG	30.70%
6	GREMIO	30.59%
7	SAO PAULO	20.16%
8	FLUMINENSE	17.39%
9	FORTALEZA	15.34%
10	INTERNACIONAL	12.11%
11	ATHLETICO-PR	11.69%
12	CRUZEIRO	2.25%
13	CUIABA	0.12%
14	CORINTHIANS	0.07%
15	BAHIA	0.05%
16	VASCO	0.04%
17	AMERICA-MG	0.01%
18	SANTOS	0.01%
19	GOIAS	<0.01%
20	CORITIBA	<0.01%

	CHANCE DE SER G6	
1	BOTAFOGO	99.05%
2	PALMEIRAS	89.81%
3	FLAMENGO	76.17%
4	BRAGANTINO	54.10%
5	GREMIO	53.76%
6	ATLETICO-MG	53.45%
7	SAO PAULO	39.83%
8	FLUMINENSE	35.77%
9	FORTALEZA	33.34%
10	INTERNACIONAL	27.79%
11	ATHLETICO-PR	27.60%
12	CRUZEIRO	7.60%
13	CUIABA	0.55%
14	CORINTHIANS	0.52%
15	VASCO	0.28%
16	BAHIA	0.26%
17	SANTOS	0.10%
18	AMERICA-MG	0.04%
19	GOIAS	0.01%
20	CORITIBA	<0.01%

	CHANCE DE SER G7	
1	BOTAFOGO	99.56%
2	PALMEIRAS	93.38%
3	FLAMENGO	83.50%
4	GREMIO	64.21%
5	ATLETICO-MG	64.21%
6	BRAGANTINO	64.19%
7	SAO PAULO	50.53%
8	FLUMINENSE	46.38%
9	FORTALEZA	43.62%
10	INTERNACIONAL	37.55%
11	ATHLETICO-PR	37.42%
12	CRUZEIRO	11.94%
13	CUIABA	1.09%
14	CORINTHIANS	0.99%
15	VASCO	0.67%
16	BAHIA	0.46%
17	SANTOS	0.19%
18	AMERICA-MG	0.10%
19	GOIAS	0.01%
20	CORITIBA	<0.01%

	CHANCE DE SER G8	
1	BOTAFOGO	99.83%
2	PALMEIRAS	95.97%
3	FLAMENGO	89.45%
4	ATLETICO-MG	74.03%
5	GREMIO	73.79%
6	BRAGANTINO	73.03%
7	SAO PAULO	61.34%
8	FLUMINENSE	56.85%
9	FORTALEZA	54.72%
10	INTERNACIONAL	48.31%
11	ATHLETICO-PR	47.72%
12	CRUZEIRO	18.23%
13	CUIABA	2.02%
14	CORINTHIANS	1.83%
15	VASCO	1.26%
16	BAHIA	0.92%
17	SANTOS	0.35%
18	AMERICA-MG	0.29%
19	GOIAS	0.05%
20	CORITIBA	0.01%

	CHANCE DE SER G9	
1	BOTAFOGO	99.90%
2	PALMEIRAS	97.82%
3	FLAMENGO	93.47%
4	ATLETICO-MG	82.56%
5	GREMIO	82.44%
6	BRAGANTINO	81.50%
7	SAO PAULO	71.43%
8	FLUMINENSE	68.06%
9	FORTALEZA	65.74%
10	INTERNACIONAL	59.30%
11	ATHLETICO-PR	58.86%
12	CRUZEIRO	26.48%
13	CUIABA	3.67%
14	CORINTHIANS	3.21%
15	VASCO	2.38%
16	BAHIA	1.68%
17	SANTOS	0.76%
18	AMERICA-MG	0.61%
19	GOIAS	0.12%
20	CORITIBA	0.03%

	CHANCE DE SER REBAIXADO	
1	CORITIBA	88.45%
2	GOIAS	77.11%
3	AMERICA-MG	54.53%
4	SANTOS	51.53%
5	BAHIA	38.17%
6	VASCO	34.09%
7	CORINTHIANS	26.98%
8	CUIABA	24.41%
9	CRUZEIRO	3.02%
10	ATHLETICO-PR	0.54%
11	INTERNACIONAL	0.38%
12	FORTALEZA	0.24%
13	FLUMINENSE	0.21%
14	SAO PAULO	0.16%
15	BRAGANTINO	0.06%
16	GREMIO	0.04%
17	ATLETICO-MG	0.04%
18	FLAMENGO	0.01%
19	PALMEIRAS	0.01%
20	BOTAFOGO	<0.01%

	PONTOS MIN E MAX		
1	BOTAFOGO	46	97
2	PALMEIRAS	43	90
3	FLAMENGO	38	93
4	BRAGANTINO	38	87
5	ATLETICO-MG	37	86
6	GREMIO	36	86
7	SAO PAULO	37	85
8	FLUMINENSE	36	80
9	FORTALEZA	34	83
10	INTERNACIONAL	35	82
11	ATHLETICO-PR	30	82
12	CRUZEIRO	28	77
13	CUIABA	22	69
14	CORINTHIANS	19	66
15	VASCO	19	68
16	BAHIA	20	66
17	SANTOS	21	66
18	AMERICA-MG	16	64
19	GOIAS	17	63
20	CORITIBA	10	57

Dados gerados pelas simulações do modelo para o campeonato Brasileiro Série A, atualizado em 03/07/2023.

6 EXPERIMENTAÇÕES REALIZADAS

Neste capítulo, serão mostradas análises e comparações práticas realizadas no contexto deste trabalho. Vários campeonatos de futebol foram analisados de início a fim, e para avaliar o funcionamento do projeto desenvolvido foram feitas comparações empíricas com outros modelos de predição de partidas de futebol, como os mencionados no capítulo 3.

6.1 METODOLOGIA

Antes de mostrar os experimentos realizados, precisamos antes definir a metodologia adotada para a realização dos experimentos, garantindo que estes sejam feitos de maneira consistente e reprodutível. Considerando a operacionalidade do trabalho, detalhado ao longo do capítulo 5, foi escolhido o Campeonato Brasileiro de Futebol, Série A, para realizar as principais comparações, por mais que o programa desenvolvido seja compatível com outros campeonatos. Isto se dá por vários fatores, como maior familiaridade por nossa parte, maior facilidade de achar recursos e trabalhos sobre o mesmo assunto, dentre outros.

Para realizarmos a experimentação, então, desenvolvemos a seguinte metodologia, a ser executada para cada um dos campeonatos desejados:

1. A cada rodada, os resultados de todas as partidas desta rodada são previstos pelo modelo e salvos como "vitória do time A", "empate" ou "vitória do time B". É importante que estes valores sejam salvos para que sejamos capazes de realizar comparações e análises.

2. Periodicamente, ao final dos dias em que ocorreu pelo menos um jogo, os coeficientes dos times são atualizados após o final das partidas, como detalhado na seção 5.4. Além disso, é recalculado o valor de correção de empates, Rho (seção 2.1.3), que nem sempre sofre mudanças.

3. Todos os resultados (probabilidades de vitória do time A, de empate e de vitória do time B) previstos pelo modelo são salvos em uma planilha eletrônica, junto às previsões de todos os algoritmos e modelos de predição de interesse para comparação. Todos são comparados usando a Distância de De Finetti, detalhado na seção 2.2.1, a fim de determinar qual modelo de predição obteve melhor resultado.

Por fim, para determinar se os resultados obtidos foram estatisticamente significativos, utilizamos testes estatísticos para determinar o nível de confiança de um modelo de predição ser superior a outro. As previsões de cada partida por cada algoritmo ou modelo foram usadas como amostras, e a partir disso são geradas análises que são usadas para

determinar a probabilidade de determinar a probabilidade do algoritmo ou modelo ser realmente melhor que o outro.

6.2 VARIÁVEIS DO MODELO: ESCOLHAS E MOTIVAÇÕES

Foram citadas, durante o capítulo 5, variáveis cujo usuário deveria definir, uma vez que tratam-se de valores subjetivos e sem comprovação de um valor ótimo. Nesse capítulo, porém, iremos abordar resultados obtidos pelo uso do modelo preditivo ao longo dos últimos anos, e portanto devemos explicitar quais valores foram utilizados para alcançar os resultados desejados.

A variável *PERIODO_MME*, definida no arquivo de configuração *config.py*, foi utilizada com valor 17,4. Este valor, aplicado à equação da média móvel exponencial introduzida na seção 2.1.4, nos mostra que ao atualizar os valores de um time após uma partida da mesma, damos um peso 8,2 vezes maior ao valor da variável até então em relação ao novo valor recebido que busca atualizar a informação, uma vez que $\frac{17,4-1}{2} = 8,2$. Ou seja, caso tenhamos, por exemplo, uma variável de força do time A como 2 e queremos atualizá-la após extrair uma partida em que a mesma obteve avaliação de força 3, teremos o valor para a nova força calculada como:

$$\frac{2 \cdot 8,2 + 3 \cdot 1}{8,2 + 1} \approx 2,1087$$

A utilização do valor 17,4 não é matematicamente justificável, e foi escolhida ao buscar um valor que não atualize as métricas de maneira muito brusca nem muito suave, dando uma relevância acima ou abaixo da adequada a cada partida nova realizada.

Outras duas variáveis subjetivas são *penalty_factor* e *points_deficit*, introduzidas na seção 5.6. A primeira busca medir o impacto das penalidades a serem aplicadas na otimização, e foi utilizado o valor 1,3375. Essa decisão foi tomada por parecer um bom valor de quantificação das penalidades, sem influenciar em muito os cálculos realizados antes da otimização, porém se mostrando influente conforme os pontos de penalidade são maiores.

Já a maneira de gerar os valores de *points_deficit* é mais complexa, uma vez que não trata-se de um valor fixo utilizado na função de otimização, mas sim de uma variável que deve crescer conforme o grau de penalidade que um time deva ter em uma partida cresça. Ou seja, caso estejamos calculando as probabilidades de uma partida em que ambos os times não devam receber penalidades, ambos teriam essas variáveis como 0. Porém, se a equipe A tem alguma penalidade a receber, seu valor de penalidade poderia ser por exemplo 10, mostrando-se um valor positivo que alteraria negativamente os cálculos a favor desse time.

Com isso, foi utilizada a seguinte regra para punir times em situações adversas para a partida com probabilidades calculadas: consideramos que cada jogador que desfalque

um time em uma partida gera uma penalidade de zero a sete pontos, dependendo de sua frequência no time titular. Ou seja, caso um time jogue uma partida com três desfalques, sendo um jogador que nunca é titular e outros dois que sempre são titulares, teremos uma penalidade aplicada de $0 + 7 + 7 = 14$ pontos. Por último, alguns cenários nas etapas finais de um campeonato podem desincentivar um dos lados em uma partida. Supondo um cenário de última rodada do Campeonato Brasileiro em que o Flamengo irá enfrentar o Coritiba, e o Coritiba já esteja matematicamente rebaixado: é de se esperar que o Coritiba não esteja tão focado na partida, ou não se importe tanto com seu resultado, e dessa maneira tenha um incentivo menor para com o jogo. Nesses casos, por padrão, penalizamos o time desinteressado em quarenta pontos.

O projeto, portanto, foi usado, utilizando os critérios nessa seção definidos. Seus resultados foram devidamente armazenados, e podem ser observados ao longo desse capítulo.

6.3 RESULTADOS DO CAMPEONATO BRASILEIRO SÉRIE A

Com a metodologia definida e escolhas subjetivas feitas, podemos começar a demonstrar os resultados obtidos na experimentação. Como mencionado anteriormente, o principal campeonato escolhido para a realização das análises foi o Campeonato Brasileiro de Futebol, Série A. O Campeonato Brasileiro, também conhecido como "Brasileirão", possui 20 times que disputam um total de 38 rodadas, no qual cada time joga contra cada outro time duas vezes, alternando o mando de campo, totalizando 380 partidas. Este formato está presente desde 2006, e portanto corresponde ao formato adotado em todos os anos observados nas experimentações realizadas.

Para realizar uma breve análise dos resultados nesta seção, foram escolhidas duas métricas de alta importância para os times do campeonato: a probabilidade de um time ser campeão, e a probabilidade de um time ser rebaixado. Para gerar gráficos pouco poluídos e fáceis de entender, foi escolhido apenas gerar os gráficos de campeão para os times do G4, e de rebaixamento para os times do Z4[1], mas note que é possível gerar gráficos com todos os times do campeonato. Além disso, caso seja de interesse, várias outras métricas estão disponíveis para a geração de gráficos: todas que foram mencionadas na seção 5.5.

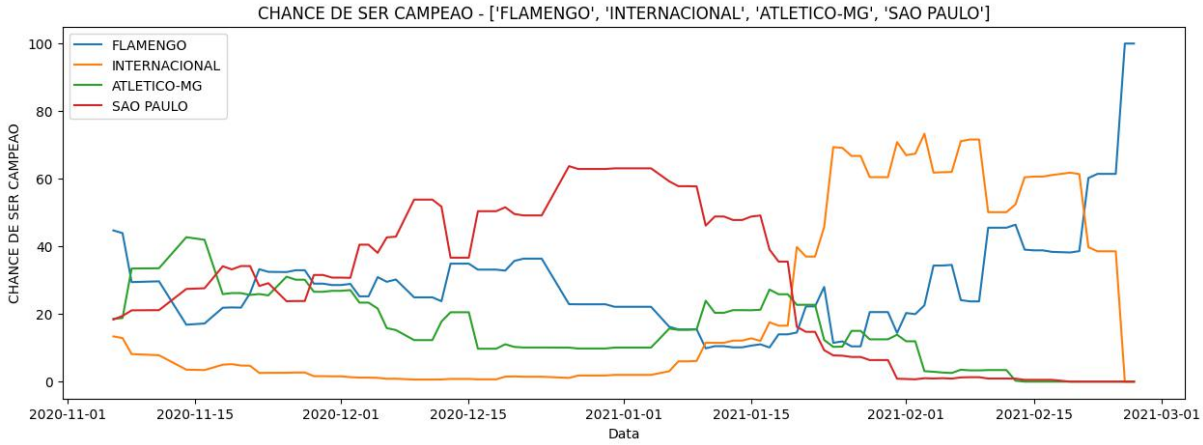
Algo importante de destacar é que a concepção desses cálculos ocorreu durante a edição de 2020 do Brasileirão. São consideradas apenas probabilidades deste campeonato a partir de 06 de Novembro de 2020, na qual ocorria a vigésima rodada, uma vez que a coleta de dados de modelos externos nem sempre é possível após a data de ocorrência das partidas. Outro detalhe é que a edição de 2023, por não estar encerrada durante a concepção deste artigo, está incompleta, e as análises foram feitas até a décima quarta rodada, que era a rodada mais atual quando este texto foi escrito.

[1] Z4 significa que o time está entre os 4 times com menor colocação no campeonato. No Brasileirão, todos os times do Z4 são rebaixados para a série B no ano seguinte.

6.3.1 Brasileirão 2020

Neste ano, ao final do campeonato, o Flamengo foi campeão, com 71 pontos, seguido pelo Internacional, que ficou em 2° lugar com 70 pontos, Atlético Mineiro, que ficou em 3° lugar com 68 pontos, e São Paulo, que ficou em 4° lugar com 66 pontos. O gráfico com a probabilidade de cada um destes times serem campeões, calculadas pelo modelo, pode ser visto na figura 7. É possível observar no gráfico que o São Paulo era o favorito por muitas rodadas, mas teve uma queda repentina e acabou terminando o campeonato em 4° lugar. O Flamengo, porém, teve um crescimento repentino nas últimas rodadas, e acabou ganhando o campeonato por muito pouco, com uma diferença de apenas um ponto.

Figura 7 – Brasileirão 2020: Probabilidades por rodada de times que terminaram no G4 serem campeões



Os quatro times com a menor colocação neste campeonato foram o Vasco da Gama com 41 pontos, Goiás com 37 pontos, Coritiba com 31 pontos e, por último, Botafogo com 27 pontos. Estes times foram rebaixados para a série B do Brasileirão, e suas probabilidades de serem rebaixados ao longo das rodadas podem ser acompanhadas na figura 8.

6.3.2 Brasileirão 2021

Nesta edição do Brasileirão, os quatro times com maior colocação foram: Atlético Mineiro, campeão com 84 pontos; Flamengo, 2° lugar com 71 pontos; Palmeiras, 3° lugar com 66 pontos; e Fortaleza, 4° lugar com 58 pontos. As probabilidades a cada rodada de cada time que terminou no G4 ser campeão pode ser visto na figura 9. Diferente da edição de 2020, neste campeonato o time vencedor era favorito para ser campeão de acordo com o modelo desenvolvido por muitas rodadas, e concretizou-se como tal.

Os quatro times com a menor colocação neste campeonato foram: Grêmio, com 43 pontos; Bahia, também com 43 pontos mas com um total de vitórias de 11, comparado ao Grêmio, com 12; Sport Recife, com 38 pontos e Chapecoense, com 15 pontos. O

Figura 8 – Brasileirão 2020: Probabilidades por rodada de times que terminaram no Z4 serem rebaixados

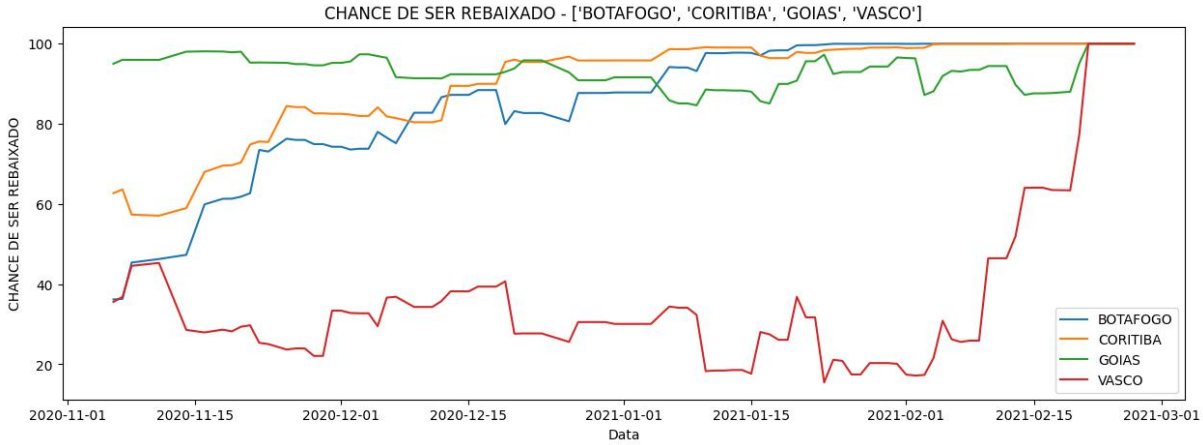


Figura 9 – Brasileirão 2021: Probabilidades por rodada de times que terminaram no G4 serem campeões

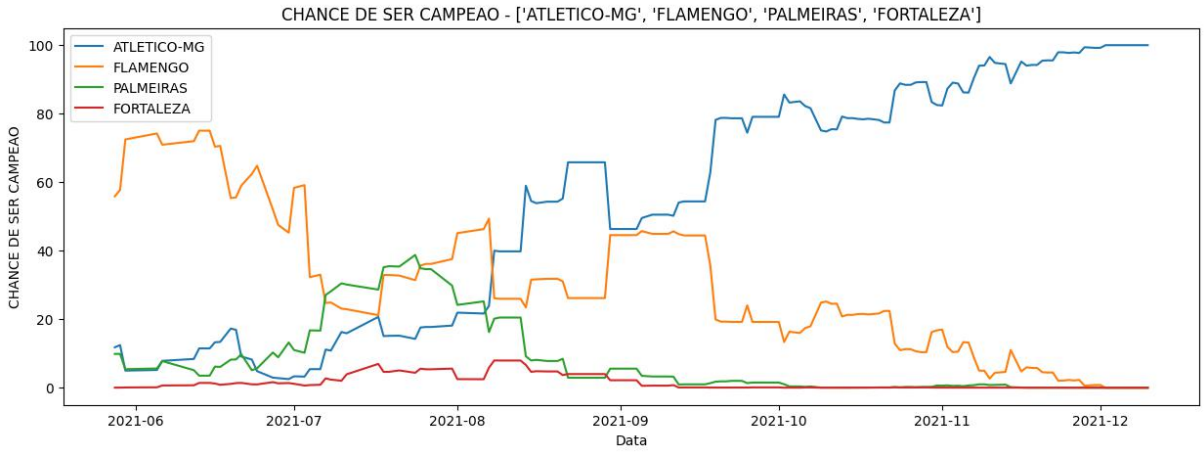


gráfico com a probabilidade de cada um destes times ser rebaixado ao longo das rodadas encontra-se na figura 10.

6.3.3 Brasileirão 2022

Já nesta edição do Brasileirão, os quatro times com maior colocação foram: Palmeiras, campeão com 81 pontos; Internacional, 2° lugar com 73 pontos; Fluminense, 3° lugar com 70 pontos; e Corinthians, 4° lugar com 65 pontos. As probabilidades a cada rodada de cada time que terminou no G4 ser campeão pode ser visto na figura 11. Como é possível observar, nesta edição do campeonato, o Palmeiras realizou uma campanha extremamente dominante, e o modelo o definia como favorito para ser campeão por muitos meses antes do campeonato acabar.

Um adendo à figura 11 que pode ser feito é que por mais que estes times tenham

Figura 10 – Brasileirão 2021: Probabilidades por rodada de times que terminaram no Z4 serem rebaixados

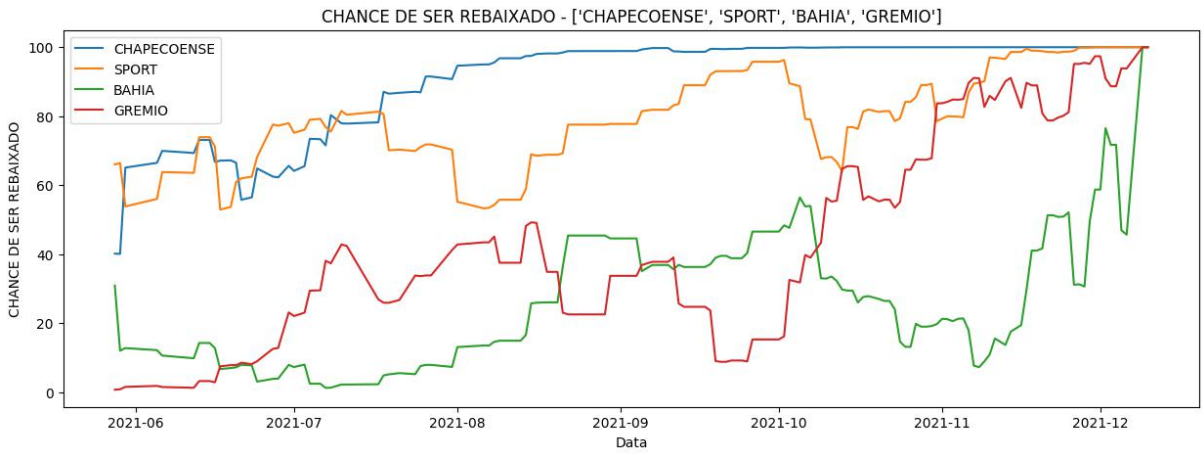
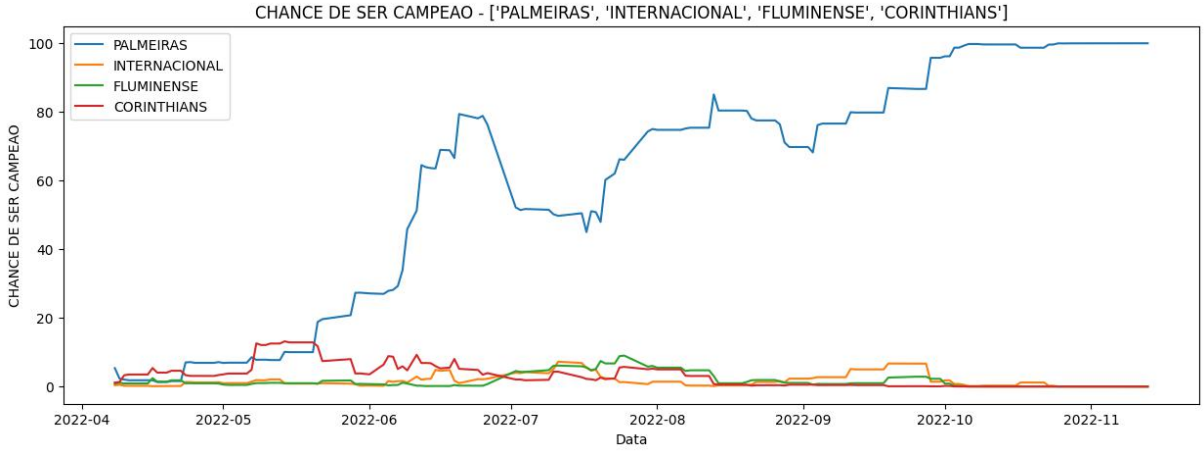


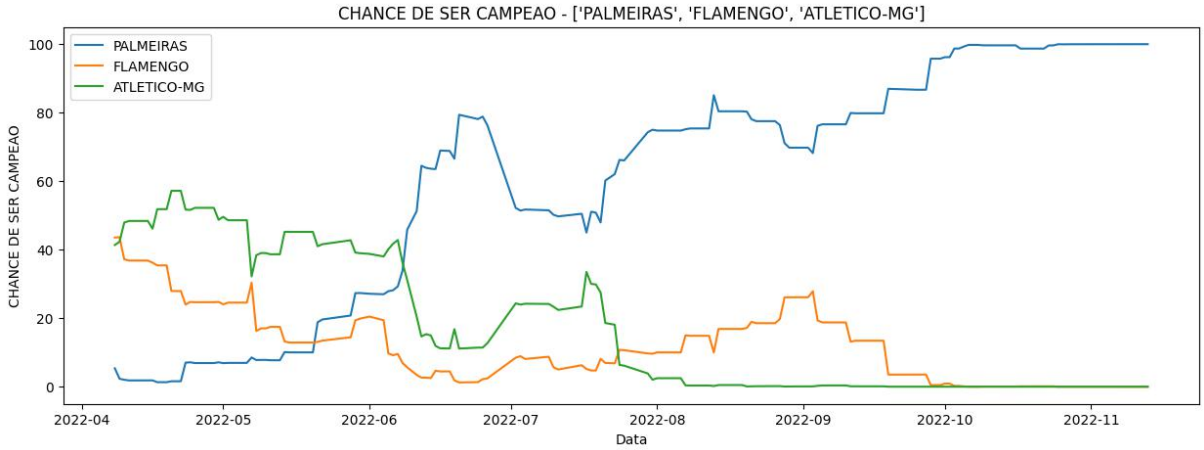
Figura 11 – Brasileirão 2022: Probabilidades por rodada de times que terminaram no G4 serem campeões



terminado o campeonato no G4, nenhum deles, exceto o Palmeiras, era um forte concorrente ao título. Um gráfico diferente foi gerado, na figura 12, mostrando as chances de ser campeão entre os times Palmeiras, Flamengo e Atlético Mineiro, que eram os times que tiveram uma disputa maior pela taça, mas tiveram uma queda ao final do campeonato. A omissão destes times no gráfico, como explicado anteriormente, é feita apenas para simplificar a visualização dos gráficos.

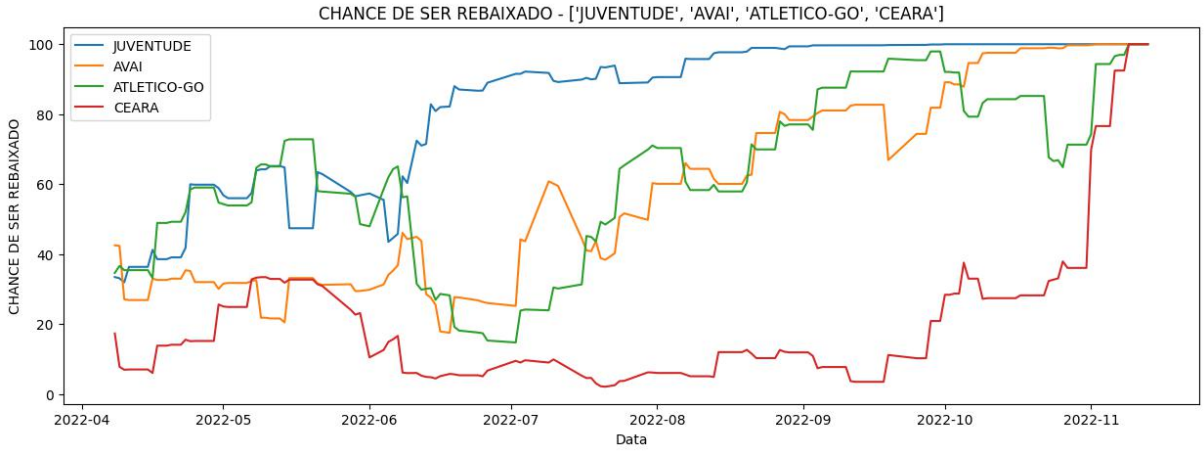
Os quatro times com a menor colocação neste campeonato foram: Ceará, com 37 pontos; Atlético Goianiense, com 36 pontos; Avaí, com 35 pontos; e Juventude, com 22 pontos. O gráfico com a probabilidade de cada um destes times ser rebaixado ao longo das rodadas encontra-se na figura 13.

Figura 12 – Brasileirão 2022: Probabilidades por rodada de Palmeiras, Atlético-MG e Flamengo serem campeões



Entre estes times, houve uma disputa mais acirrada nas previsões do modelo preditivo, mas dois deles ficaram fora do G4 ao final do campeonato

Figura 13 – Brasileirão 2022: Probabilidades por rodada de times que terminaram no Z4 serem rebaixados

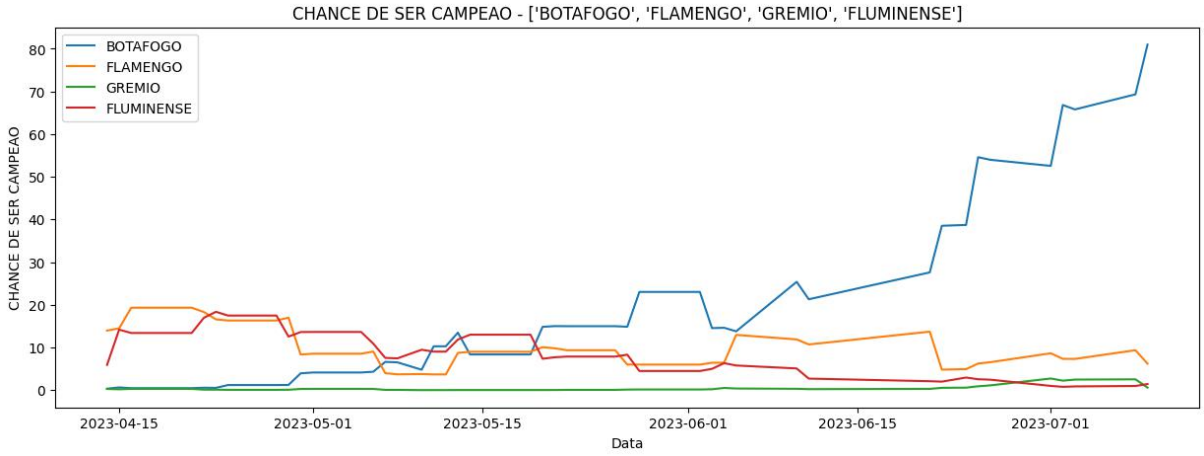


6.3.4 Brasileirão 2023

Na edição de 2023 do Brasileirão, ainda em andamento enquanto este trabalho foi escrito, os 4 times que possuíam a maior colocação na rodada mais atual eram o Botafogo em 1°, com 36 pontos; Flamengo em 2°, com 26 pontos; Grêmio em 3°, também com 26 pontos mas com um saldo de gols igual a 4, menor que 9 do Flamengo; e Fluminense em 4°, com 24 pontos. As probabilidades a cada rodada destes times ser campeão pode ser visto na figura 14. Como pode ser observado no gráfico, o Botafogo está fazendo uma campanha muito promissora, chegando a bater 80% de chance de ser campeão de acordo com o modelo, caso mantenha seu nível de atuação nos jogos. O Palmeiras, por mais que tenha tido um começo de campeonato muito bom, ficou fora do G4 na 14ª rodada devido

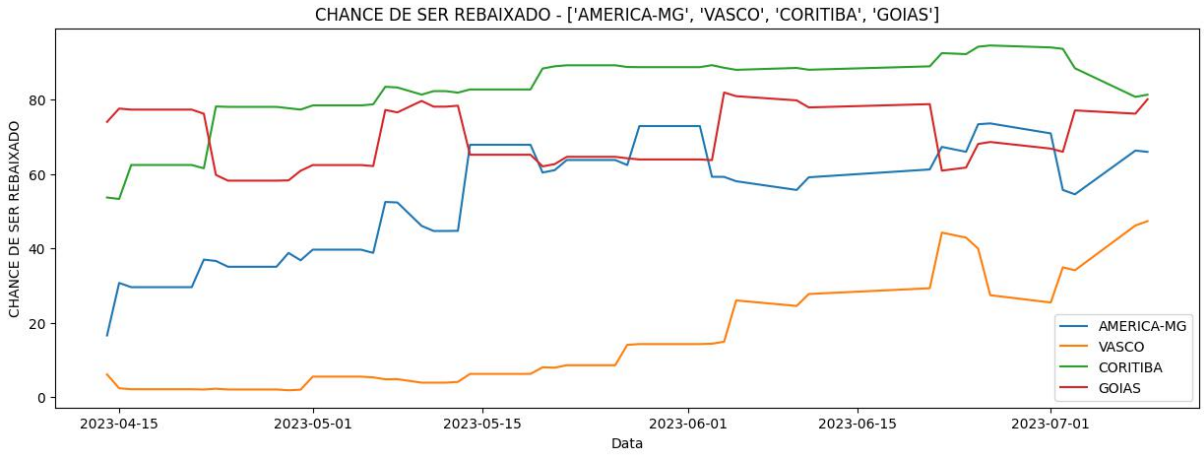
a uma série de empates e derrotas, e por isso não foi adicionado ao gráfico.

Figura 14 – Brasileirão 2023: Probabilidades por rodada de times que estão atualmente no G4 serem campeões



Já os quatro times com a menor colocação atual neste campeonato são: Goiás, com 11 pontos; Coritiba, com 10 pontos; Vasco da Gama, com 9 pontos; e América-MG, também com 9 pontos mas com saldo de gols baixíssimo de -16, comparado com -12 do Vasco da Gama. O gráfico com a probabilidade de cada um destes times ser rebaixado ao longo das rodadas encontra-se na figura 15.

Figura 15 – Brasileirão 2023: Probabilidades por rodada de times que terminaram no Z4 serem rebaixados



6.4 COMPARAÇÃO COM OUTROS ALGORITMOS

Para comparar performances de diferentes algoritmos podemos utilizar o conceito apresentado na seção 2.2.1: a Distância de De Finetti. Foi criada, para cada temporada do

campeonato brasileiro, uma planilha eletrônica, com o objetivo de armazenar as probabilidades de cada partida realizada naquele campeonato, para cada algoritmo observado. Dessa forma, foram criadas diferentes abas para cada algoritmo, além das abas *Ranking* e *Resultados*.

Foram utilizadas três métricas associadas ao cálculo da distância de De Finetti. Ainda que todas elas sejam calculadas com o mesmo princípio, e portanto tenham resultados que as classifiquem de uma mesma forma, cada métrica contribui para uma fácil visualização ou comparação com um conceito específico, e portanto foram todas evidenciadas no resultado final, apresentado na aba *Ranking*. As métricas são:

- Distância: essa métrica é exatamente o valor calculado pela distância de De Finetti. Assume valores entre 0 e 2, e quanto menor, melhor.

- Precisão: métrica criada para ser mais intuitiva que a *distância*. Assume valores entre 0% e 100%, e quanto maior, melhor. É calculada diretamente a partir do valor obtido para a *distância*, com a equação A.1, onde x representa a Distância de De Finetti e $P(x)$ nos diz a precisão. Dessa forma, a precisão recebe o valor 100% para o caso da *distância* ser 0; valor 0% para o caso da *distância* ser 2, e valores proporcionais para qualquer valor entre os extremos explicitados, trazendo maior naturalidade em sua interpretação.

- Pontuação: uma característica relevante para o valor calculado na métrica *distância* é sua comparação com o valor $0,\overline{6}$, como destacado na seção 2.2.1. Algoritmos com resultado médio da *distância* acima desse valor são vistos como inferiores a um algoritmo que simplesmente atribuísse valores iguais a cada probabilidade todas as vezes, e portanto é um cenário que devemos observar caso ocorra. Pensando nisso, foi desenvolvida a métrica *Pontuação*, a qual busca ter como valor mais negativo -100; como mais positivo 100; e ter no valor 0 a distinção entre a superação do algoritmo com probabilidades sempre iguais ou não. Como essa função não é linear, foi necessária a aplicação de uma equação de segundo grau. Essa equação é a equação 6.2, na qual p é a precisão obtida, e $A(p)$ nos dá a pontuação.

$$P(p) = 1 - \frac{p}{2} \tag{6.1}$$

$$A(p) = 150p^2 + 50p - 100 \tag{6.2}$$

Nas abas de cada algoritmo temos cada partida realizada no campeonato, as probabilidades previamente calculadas pelo algoritmo, o resultado da partida, e a *precisão* calculada. O resultado é exposto como *1* em caso de vitória do time mandante; *X* em caso de empate; e *2* em caso de vitória do time visitante. É possível visualizar também nessas abas um resumo à direita, com a precisão média geral obtida, e agrupada por cada time.

Na página *Resultados* podemos ver cada partida, seu resultado, e a *precisão* obtida por cada algoritmo, permitindo assim uma rápida comparação dos resultados de cada embate. Na página *Ranking*, finalmente, é possível observar o resultado final da comparação de cada algoritmo, classificando-os de acordo com a melhor pontuação para a pior, naquela temporada.

Ao longo dos últimos anos, os dados de diversos projetos foram coletados e armazenados nas planilhas. Como muitas dessas informações apenas ficam públicas por um certo período de tempo, como o período em que a rodada atual do campeonato está ocorrendo, esses dados são hoje de difícil acesso, e portanto diferentes anos possuem diferentes algoritmos na comparação.

Os projetos do Five... (2023) e Arruda (2023) estão presentes nas comparações de todos os anos, os quais vão de 2020 a 2023. O algoritmo do Espião... (2023) consta apenas a partir do ano de 2022, data de sua divulgação. Nesse ano, porém, seis partidas ao longo da temporada não tiveram suas probabilidades divulgadas, e os valores utilizados na comparação foram os da nossa implementação, evitando resultados díspares em tais partidas. O algoritmo desenvolvido por Lima et al. (2023), conhecido como "Probabilidades no Futebol", apenas consta em nossa avaliação do ano de 2022, e não apresentou resultado significativo. O difícil acesso a seus dados passados, combinado com o baixo desempenho, resultou na não observação do mesmo na comparação atual de 2023. Os projetos citados nesse parágrafo são descritos com maiores detalhes na seção 3.

As casas de aposta, apesar de apresentarem resultados regidos de acordo com a movimentação do mercado, foram incluídas na comparação. Trata-se de um mercado que é regulado pela oferta e demanda dos pagamentos oferecidos e da vontade de seus usuários de os consumir. Enquanto um pagamento está sendo consumido em grande volume, seu valor é diminuído, ao passo que os pagamentos com menor interesse sobem, regulando-se dinamicamente ao longo do tempo. Ainda que não seja um algoritmo, podemos obter as probabilidades implícitas de cada partida nas casas de aposta ao analisar esses valores de pagamento oferecidos. A conversão desses valores para probabilidades é simples, e é descrita no apêndice A. Para obter tais informações é utilizado o site https://www.football-data.co.uk/brazil.php, o qual disponibiliza os dados de casas de aposta em formato CSV.

As planilhas disponíveis são as dos anos 2020, 2021, 2022 e 2023[2]. Os resultados, porém, para maior facilidade, podem ser vistos nas Tabelas 2, 3, 4 e 6, respectivamente.

Como se pode ver, a implementação do modelo proposto no presente trabalho aparenta obter desempenho melhor que os demais algoritmos nos anos de 2020, 2021 e 2022, ficando atrás apenas das casas de aposta, as quais, como descritas anteriormente, são reguladas pelo mercado, e não por um algoritmo. Esse resultado é visto como extremamente positivo,

[2] Disponíveis em: https://drive.google.com/drive/folders/138bsRYTPOcz795vvFaOmP8L_e8sMMkUM

Tabela 2 – Comparação dos modelos e algoritmos para o Campeonato Brasileiro de 2020

Posição	Algoritmo	Distância	Precisão	Pontuação
1	Casas de Aposta	0,606	69,69%	7,69
2	Nossa Implementação	0,608	69,59%	7,42
3	*FiveThirtyEight*	0,618	69,12%	6,23
4	Chance de Gol	0,625	68,76%	5,29

Tabela 3 – Comparação dos modelos e algoritmos para o Campeonato Brasileiro de 2021

Posição	Algoritmo	Distância	Precisão	Pontuação
1	Casas de Aposta	0,608	69,59%	7,44
2	Nossa Implementação	0,616	69,22%	6,48
3	*FiveThirtyEight*	0,619	69,07%	6,08
4	Chance de Gol	0,630	68,49%	4,60

Tabela 4 – Comparação dos modelos e algoritmos para o Campeonato Brasileiro de 2022

Posição	Algoritmo	Distância	Precisão	Pontuação
1	Casas de Aposta	0,608	69,62%	7,53
2	Nossa Implementação	0,609	69,56%	7,35
3	*Chance de Gol*	0,612	69,40%	6,93
4	*FiveThirtyEight*	0,614	69,28%	6,64
5	Espião Estatístico	0,617	69,14%	6,27
6	Probabilidades no Futebol	0,660	66,99%	0,82

mostrando a força do projeto ao obter resultados superiores em termos das métricas de avaliação empregadas, em comparação a algoritmos mundialmente conhecidos, como o do Five... (2023), nacionalmente aclamados como o Arruda (2023) e Lima et al. (2023), e de grandes empresas como o Espião... (2023), o qual pertence à Globo. O desempenho de 2023 está abaixo do *Chance de Gol* até o momento, mas o campeonato ainda está em andamento, e mesmo assim o resultado é muito satisfatório, ao se posicionar acima do *FiveThirtyEight* e *Espião Estatístico*.

Outro fator importante é que as casas de aposta obtiveram a maior precisão dentre todos os modelos e algoritmos de predição. Inicialmente, este resultado pode parecer estranho, já que as "previsões" geradas pelas casas de apostas são derivadas diretamente dos *odds* dos possíveis resultados, valores que são determinados pelas apostas dos usuários.

Tabela 5 – Comparação dos modelos e algoritmos para o Campeonato Brasileiro de 2023

Posição	Algoritmo	Distância	Precisão	Pontuação
1	Chance de Gol	0,591	70,44%	9,65
2	Casas de Aposta	0,593	70,35%	9,42
3	Nossa Implementação	0,598	70,10%	8,75
4	*FiveThirtyEight*	0,605	69,74%	7,81
5	Espião Estatístico	0,613	69,36%	6,85

Surowiecki (2004) e Ugander, Drapeau e Guestrin (2015) oferecem uma possível explicação para tamanha precisão, detalhando que a agregação de informações por grandes grupos de pessoas resultam em decisões melhores do que poderiam ser determinadas por membros individuais do grupo.

Pode-se perceber, também, que todas as avaliações de todos os anos resultaram em algoritmos com avaliações estritamente positivas. O mais próximo de uma pontuação negativa foi o resultado do projeto "Probabilidades no Futebol", na temporada de 2022, com pontuação 0,82, mas ainda assim positiva.

6.5 NÍVEL DE CONFIANÇA

Deseja-se avaliar se um modelo de previsão é estatisticamente superior ao outro em termos de suas previsões. Para isso, considera-se uma hipótese nula (H_0) que afirma que não há diferença significativa entre eles, e uma hipótese alternativa (H_1) que afirma que um modelo é de fato superior ao outro.

Para avaliar a hipótese nula, realizamos um teste de t-student pareado, utilizando a biblioteca scipy... (2023) para a diferença entre as métricas de predição dos dois modelos preditivos. Denotamos a média da métrica de predição do modelo 1 como Z e a média da métrica de predição do modelo 2 como W. O teste estatístico compara as médias das métricas, levando em consideração as variabilidades amostrais.

Calculamos o p-valor associado a esse teste estatístico. O p-valor representa a probabilidade de observar uma diferença tão extrema quanto a observada, assumindo que a hipótese nula seja verdadeira. Um p-valor menor que um nível de significância pré-definido (α) indica evidência suficiente para rejeitar a hipótese nula em favor da hipótese alternativa.

Os resultados do teste de hipótese são apresentados na Tabela 6, mostrando os níveis de confiança calculados para o modelo desenvolvido neste trabalho ser estatisticamente superior aos outros algoritmos e modelos de predição do mercado. O nível de confiança é calculado a partir do p-valor, onde um nível de confiança de 95% corresponde a um p-valor de 0.05.

Tabela 6 – Nível de confiança de a nossa implementação ser superior a outros algoritmos

Algoritmo	**Nível de Confiança**
Probabilidades no Futebol	99,99%
Chance de Gol	98,41%
FiveThirtyEight	95,85%
Espião Estatístico	89,41%

Note que um fator que influencia muito o nível de confiança é a quantidade de amostras. O Espião Estatístico possui um número muito menor de amostras em relação aos outros modelos de predição, que pode ser um dos motivos para sua pontuação mais baixa.

Por questões comparativas, também fizemos o cálculo do nível de confiança para as Casas de Aposta serem superiores ao modelo preditivo desenvolvido neste trabalho. Neste caso, obtivemos um nível de confiança de 83,25%, sendo portanto um valor menor do que os apresentados na tabela 6.

Com essas considerações, estes experimentos evidenciam o sucesso do projeto desenvolvido. Seu uso pode ser expandido para diferentes áreas, um exemplo é o de apostas esportivas, abordado no apêndice A.

7 CONCLUSÃO

Neste trabalho, conclui-se que a aplicação conjunta de métodos estatísticos e paradigmas de programação pode trazer vários ganhos para a realização de previsões de resultados de esportes, com enfoque no futebol. Assim como feito por Arruda (2000), é utilizado o Modelo Poisson Duplo para a determinação de placares das partidas; porém, o desafio da utilização de tal modelo consiste na determinação dos valores corretos de λ. Foi demonstrado e detalhado o desenvolvimento de um modelo iterativo de predição de partidas de futebol, de como determinar valores para λ usando tal modelo preditivo, e de como o uso deste em campeonatos pode gerar uma variada gama de aplicações.

O projeto desenvolvido neste trabalho foi o que obteve a melhor precisão, avaliada usando a Distância de De Finetti, quando comparado com outros modelos de previsão mais conhecidos, como o Five... (2023), Arruda (2023), Lima et al. (2023), e Espião... (2023), exceto na edição de 2023 do Brasileirão Série A, onde Arruda (2023) esteve à frente, mas com o campeonato ainda em andamento durante a escrita deste artigo.

Além disso, mostra-se que as casas de aposta possuem a maior precisão dentre todos os modelos e algoritmos de previsão. Por isso, para a aplicação deste trabalho em apostas desportivas, seria necessário o desenvolvimento de estratégias para minimizar o risco presente em realizar apostas. Este assunto é mais detalhado no apêndice A.

Por fim, vimos que a implementação do projeto desenvolvido, apesar de demonstrar grande capacidade de realizar o que se propõe, é um tanto mecânica e imprática para um usuário não especializado. Sendo assim, alguns pontos que poderiam ser melhorados ou aprofundados futuramente incluem:

Maior facilidade de uso e mais possibilidades para o usuário final

A página Web foi desenvolvida para ser uma ferramenta fácil e intuitiva para que o usuário possa acessar, consultar os dados e com eles interagir. Porém, sua utilização sem conhecimento prévio do funcionamento da mesma pode ser muitas vezes prejudicada pela falta de informações presentes na própria interface para guiar o usuário. Além disso, pode-se incluir futuramente mais possibilidades de consulta à página Web, como a visualização de gráficos com probabilidades dos times serem campeões, rebaixados, entre outras. As possibilidades são enormes, e essas interações são um grande diferencial do projeto, permitindo que não só os dados sejam acessados e visualizados, mas também sejam gerados em tempo real, conforme solicitação do utilizador.

Automatização do Projeto

Outro ponto relevante de melhoria possível para o projeto é a possibilidade de sua preservação e atualização de maneira automática. O processo realizado atualmente

pode ser considerado semi-automático, onde os códigos de atualização e criação de dados são rodados manualmente, mas realizam seus processos de maneira automática a partir do *input* inserido.

Além disso, para o início de uma nova temporada, temos novos times promovidos às primeiras divisões nacionais, que precisam ser manualmente inseridos pelos mantenedores do projeto, além da atualização dos fatores de mando de campo de cada campeonato, entre outros. Com isso, pode-se perceber que uma automatização desses processos poderia facilitar e evitar a necessidade de intervenções humanas para a continuidade do projeto.

Disponibilização da implementação do modelo preditivo através de uma API

Um ponto que tornaria fácil a utilização do projeto para outros usuários, assim como a integração do mesmo com sistemas diversos, seria o desenvolvimento de uma API pública para o código. Com uma API bem definida, fazer requisições sobre os dados gerados pelo modelo tornaria-se mais simples e acessível, e facilitaria muito um aprofundamento no desenvolvimento do *website*, por exemplo.

Inclusão de novos campeonatos

Por fim, podemos incluir a possibilidade de adição de novos campeonatos. Na implementação do projeto, foram incluídos sete dentre os mais importantes campeonatos nacionais do mundo: Brasil, Inglaterra, Alemanha, Espanha, Itália, França e Portugal. Uma futura implementação poderia viabilizar adições de mais campeonatos, sejam eles de outros países ou ainda de divisões inferiores de nações já atualmente contempladas.

REFERÊNCIAS

ARRUDA, M. L. d. **Poisson, Bayes, Futebol e DeFinetti**. Dissertação (Mestrado em Estatística) — Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2000.

ARRUDA, M. L. d. **Chance de Gol**. 2023. Disponível em: https://www.chancedegol.com.br/. Acesso em: 12/04/2023.

CHEN, J. **Backtesting: Definition, How It Works, and Downsides**. [S.l.], 2021. Disponível em: https://www.investopedia.com/terms/b/backtesting.asp. Acesso em: 14/08/2023.

CORTIS, D. **Inflating or deflating the chance of a draw in soccer**. 2018. Disponível em: https://www.pinnacle.com/en/betting-articles/soccer/inflating-or-deflating-the-chance-of-a-draw-in-soccer. Acesso em: 03/07/2023.

DE FINETTI, B. **Probability, Induction and Statistics: The Art of Guessing**. [S.l.]: New York: John Wiley, 1972.

EMPACHER, C.; KAMPS, U.; VOLOVSKIY, G. Statistical prediction of future sports records based on record values. **Stats**, v. 6, n. 1, p. 131–147, 2023. ISSN 2571-905X. Disponível em: https://www.mdpi.com/2571-905X/6/1/8.

ESPIÃO Estatístico. 2023. Disponível em: https://ge.globo.com/espiao-estatistico/. Acesso em: 28/05/2023.

FIVE Thirty Eight: Club soccer predictions. 2023. Disponível em: https://projects.fivethirtyeight.com/soccer-predictions/. Acesso em: 12/04/2023.

HAMILTON, H. **Moneyball and soccer**. [S.l.], 2009. Soccermetrics Research, LLC. Disponível em: https://www.soccermetrics.net/high-level-discussions/moneyball-and-soccer-2. Acesso em: 03/07/2023.

HUNTER, J. S. The exponentially weighted moving average. **Journal of Quality Technology**, Taylor Francis, v. 18, n. 4, p. 203–210, 1986. Disponível em: https://doi.org/10.1080/00224065.1986.11979014.

KELLY, J. L. A new interpretation of information rate. **The Bell System Technical Journal**, Nokia Bell Labs, v. 35, n. 4, p. 917–926, 1956.

LIMA, B. N. B. d. et al. **Probabilidades no Futebol**: Projeto de um grupo do Departamento de Matemática da UFMG. 2023. Disponível em: https://www.mat.ufmg.br/futebol/. Acesso em: 28/05/2023.

METROPOLIS, N.; ULAM, S. The Monte Carlo Method. **Journal of the American Statistical Association**, v. 44, p. 335–341, 1949.

SCIPY.STATS.T. 2023. Disponível em: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.t.html. Acesso em: 31/08/2023.

SILVA, W. B. d. **Distribuição de Poisson bivariada aplicada à previsão de resultados esportivos**. Dissertação (Mestrado em Ciências Exatas e da Terra) — Universidade Federal de São Carlos, 2014.

SUROWIECKI, J. **The Wisdom of Crowds**. New York: Doubleday Co, 2004. 336 p.

UGANDER, J.; DRAPEAU, R.; GUESTRIN, C. The wisdom of multiple guesses. In: . New York, NY, USA: Association for Computing Machinery, 2015. (EC '15), p. 643–660. ISBN 9781450334105. Disponível em: https://doi.org/10.1145/2764468.2764529.

APÊNDICES

APÊNDICE A – UTILIZAÇÃO DO MODELO DESENVOLVIDO PARA APOSTAS ESPORTIVAS.

A possibilidade de calcular as probabilidades de partidas de futebol possibilita a utilização desses cálculos para apostas esportivas, partindo de um ponto de vista estatístico. A abordagem usa o princípio de maximizar nossos ganhos a partir de cálculos matemáticos. Com isso, utilizando o histórico de resultados de casas de aposta de 28 de agosto de 2021 a 08 de julho de 2023, simulamos a utilização da nossa implementação do modelo de predição com dados reais para determinar sua *performance*.

Para utilizarmos nossa implementação do modelo proposto com apostas esportivas, precisamos antes definir alguns conceitos importantes do mundo das apostas. Primeiro, devemos entender o que são *odds*. *Odds* são valores disponibilizados pelas casas de aposta que indicam o multiplicador de pagamento oferecido para a ocorrência de um evento. Uma *odd* de 2,1, por exemplo, significa que ao apostar X reais naquele evento, caso o mesmo ocorra, será recebido pelo apostador 2,1 vezes o valor apostado, saindo portanto com um lucro de $1{,}1X$ reais.

Podemos avaliar, para fins ilustrativos, uma disputa justa de cara ou coroa. Como ambos os eventos (sair cara ou sair coroa) possuem 50% de chance de ocorrer, é concluído que a chance de um dos eventos ocorrer é tão provável quanto a chance do mesmo não ocorrer, e portanto é justo que o pagamento do valor apostado seja tão grande quanto seu risco. Em outras palavras, a *odd* justa para o evento é 2, visto que com essa *odd* o ganho potencial é igual ao risco da aposta. Em termos gerais, podemos concluir que um evento com probabilidade p de ocorrência tem sua *odd* justa como $\frac{1}{p}$.

Dessa forma, poderíamos utilizar os valores calculados pelo modelo de predição para quantificar a probabilidade de certos eventos ocorrerem em uma partida, e com isso calcular qual seria a *odd* justa para tal evento. Caso uma partida entre Flamengo e Fluminense tenha a probabilidade de vitória do Flamengo calculada como 55%, por exemplo, saberemos que a *odd* esperada para esse evento é $\frac{1}{0{,}55} = 1{,}82$. Portanto, caso haja uma *odd* acima desse valor disponível para apostar, é entendido que o valor esperado da aposta é positivo, ou seja, caso essa aposta fosse realizada infinitas vezes, ganharia-se mais dinheiro do que se perderia, partindo do princípio que a aposta seria acertada em 55% dos casos.

Esse é um dos princípios fundamentais para que o saldo final tenda a ser positivo em apostas. Caso uma aposta não apresente valor esperado positivo, não há qualquer sentido em realizá-la, mesmo que o evento tenha alta probabilidade de ocorrer. Entretanto, é importante destacar que a realização de apostas exclusivamente de valor esperado positivo não resulta, necessariamente, em um ganho final positivo. O motivo dessa relação não ser sempre verdadeira está no fato de que o apostador possui recursos limitados, e portanto pode, antes de alcançar ganhos consideráveis, perder todo seu patrimônio numa possível

sequência de falhas.

Com esse conceito estabelecido, é essencial entender qual a forma matemática de minimizar as probabilidades de apostas de valor esperado positivo, por má gestão das quantias apostadas, resultarem em prejuízos irreparáveis. Para isso, é utilizado o Critério de Kelly (KELLY, 1956), o qual estabelece que a quantidade ótima a ser apostada ou investida em uma determinada oportunidade é diretamente proporcional à expectativa de retorno dessa oportunidade em relação ao risco envolvido. Em termos matemáticos, a fórmula do Critério de Kelly é apresentada na equação A.1.

$$f* = \frac{(p \cdot b - q)}{b} \tag{A.1}$$

Onde:

- $f*$ representa a fração da banca ou do capital disponível a ser apostada ou investida;
- p é a probabilidade de ganho da oportunidade;
- q é a probabilidade de perda da oportunidade: $1 - p$;
- b é a relação de pagamento oferecida pela oportunidade: $odd - 1$.

Uma vez que a probabilidade do evento ocorrer é algo calculado pela nossa implementação do modelo, e não uma ciência exata, como na probabilidade de obter cara no lançamento de uma moeda justa, é altamente recomendada a utilização de uma fração do valor calculado pelo Critério de Kelly. Foi utilizada em nossas simulações, portanto, o valor de 30% dos montantes retornados pela equação de Kelly. Além disso, a fim de ser ainda mais cauteloso, houve uma limitação fixa nas simulações para que nunca fosse apostado mais do que 5% do patrimônio total, a fim de evitar grandes derrotas na não ocorrência de um único evento desejado.

Por fim, vale ressaltar a escolha de não considerar apostas em alguns casos, devido a análises realizadas utilizando os gráficos disponíveis em nossa *webpage*, detalhada na subseção 5.7.3. Caso a potencial aposta seja a favor de equipes que mostrem um desempenho recente bem abaixo do esperado, ou contra equipes com um desempenho recente bem acima do esperado, optamos por não realizá-la, por entender que o modelo de predição tem apresentado dificuldades em calcular probabilidades realistas para a equipe em questão.

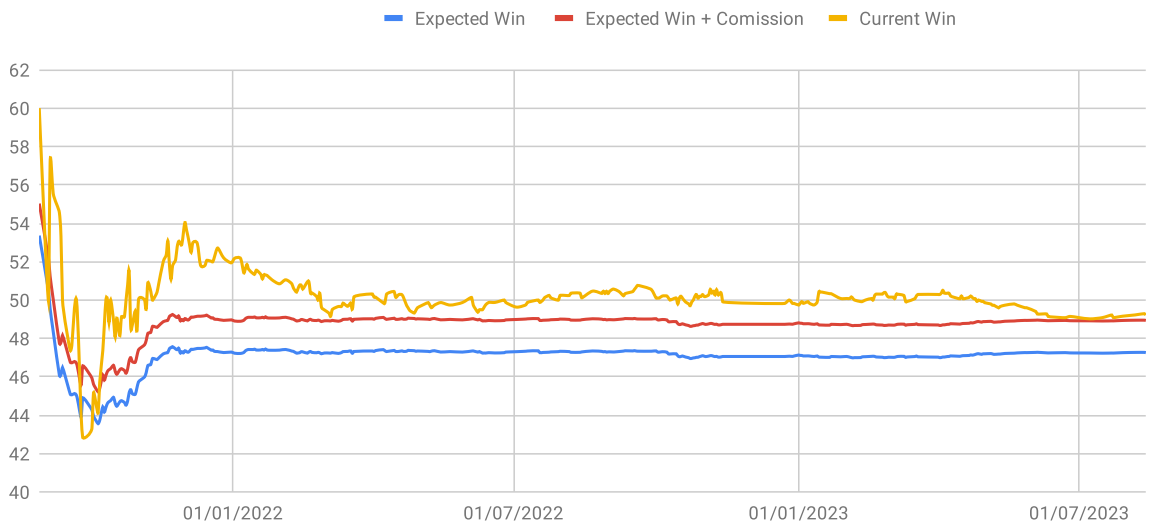
Com todos esses passos bem explicitados, sabemos como identificar uma aposta de valor esperado positivo, como definir o valor a apostar e como descartar potenciais apostas que só existam por cálculo impreciso por parte do modelo. Utilizando essa metodologia, foram identificadas, para o período de 28 de agosto de 2021 a 08 de julho de 2023, 765 apostas favoráveis. A *odd* média desse total de apostas ficou em 2,117, e portanto o total de apostas que se espera ganhar com essa *odd* ficou em $\frac{1}{2{,}117} = 47{,}24\%$.

Vale destacar que o experimento realizado utilizou uma casa de apostas online, a Betfair Exchange[1], a qual tem suas apostas realizadas entre os próprios usuários, cobrando uma comissão de 6,5% de todo lucro obtido. Dessa forma, apesar do resultado esperado implicitamente pelas *odds* apresentadas ficar em 47,24%, precisaríamos considerar o valor médio das *odds* ajustadas com a comissão cobrada, para sabermos a porcentagem necessária de se superar não só para obter resultados acima do esperado, mas também obter lucro. A equação A.2 transforma a *odd* apostada em uma *odd* com sua comissão descontada.

$$\text{Odd descontada} = (\text{odd} - 1) \cdot 0,935 + 1 \tag{A.2}$$

Utilizando essa equação, vemos que é a média da *odd* descontada média vale 2,044, trazendo uma probabilidade média implícita de 48,92%. O resultado final do experimento, porém, foi de 375 acertos, resultando em 49,02% das apostas simuladas retornando lucro. Podemos ver esse resultado ao longo do tempo na figura 16.

Figura 16 – Apostas Esportivas



Podemos ver as porcentagens (eixo Y) ao longo do tempo (eixo X). O traçado azul corresponde às porcentagens implícitas pelas *odds*. O traçado vermelho representa o valor das *odds* descontadas das comissões. O traçado amarelo representa a porcentagem de acertos simulados.

[1] Disponível em: https://www.betfair.com/br. Acesso em: 15/08/2023.