UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I – CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE ESTATÍSTICA
CURSO DE GRADUAÇÃO EM BACHARELADO EM ESTATÍSTICA

SAMUEL SOUZA ALVES DE ARAÚJO

UMA IMPLEMENTAÇÃO EM R DE MODELOS PARA CÁLCULOS DE PROBABILIDADES EM JOGOS DE FUTEBOL

CAMPINA GRANDE – PB

2023

SAMUEL SOUZA ALVES DE ARAÚJO

UMA IMPLEMENTAÇÃO EM R DE MODELOS PARA CÁLCULOS DE PROBABILIDADES EM JOGOS DE FUTEBOL

Trabalho de Conclusão de Curso apresentada ao curso de Bacharelado em Estatística do Departamento de Estatística do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Estatística.

Orientador: Prof. Gustavo Henrique Esteves.

CAMPINA GRANDE – PB

2023

A663u Araujo, Samuel Souza Alves de.
Uma implementação em R de modelos para cálculos de probabilidades em jogos de futebol [manuscrito] / Samuel Souza Alves de Araujo. - 2023.
31 p. : il. colorido.

Digitado.
Trabalho de Conclusão de Curso (Graduação em Estatística) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2023.
"Orientação : Prof. Dr. Gustavo Henrique Esteves, Departamento de Estatística - CCT. "

1. programa R. 2. python. 3. modelo probabilístico. I. Título

21. ed. CDD 629.895

Elaborada por Talita M. A. Tavares - CRB - CRB 15/971 BC/UEPB

SAMUEL SOUZA ALVES DE ARAÚJO

UMA IMPLEMENTAÇÃO EM R DE MODELOS PARA CÁLCULOS DE PROBABILIDADES EM JOGOS DE FUTEBOL

Trabalho de Conclusão de Curso apresentada ao curso de Bacharelado em Estatística do Departamento de Estatística do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Estatística.

Aprovada em: 27/06/2023.

BANCA EXAMINADORA

Prof. Dr. Gustavo Henrique Esteves (Orientador)
Universidade Estadual da Paraíba (UEPB)

Prof. Dr. Tiago Almeida de Oliveira
Universidade Estadual da Paraíba (UEPB)

Prof. Dr. Silvio Fernando Alves Xavier Junior
Universidade Estadual da Paraíba (UEPB)

À minha família, pelo apoio, DEDICO.

AGRADECIMENTOS

Aos meu pais, José Alves e Maria da Salete, minhas tias, Tereza Alves e Luzia Alves, minha falecida avó, Maria Filomena, pelo apoio e pela compreensão por minha ausência em diversas ocasiões.

À Ana Cláudia e família pelo apoio incondicional e pelo direcionamento.

À PROEST (Pró-Reitoria estudantil) e profissionais envolvidos no Programa de Moradias Universitárias ao qual me ajudou imensamente na minha trajetória acadêmica.

Aos profissionais do RU (Restaurante universitário) pelo carinho e momentos de amizade.

Aos meus amigos da moradia universitária, em especial à Maxciel, Erinaldo, Neto, Rafinha e Otacílio pelo companheirismo e boas lembranças.

Aos professores e ex professores do Departamento de Estatística, em especial aos professores Gustavo Esteves, Tiago Almeida, Ednário Mendonça e Kleber Barros pela orientação, oportunidades e parceria.

Aos amigos que conheci na UEPB, em especial a Jefferson, Joab, Tiago, Aline, Lucas, Giullber, Débora, Damião, Jiulia, Suziane, Fagna, Wellerson, Geovane e aos demais colegas de curso, pelos momentos, experiência e conhecimentos compartilhados e pelo apoio.

Aos demais amigos que também foram importantes durante minha trajetória, em especial a Matheus, Vinícius, Palmério, Ivan, Weslley, Gleyson e aos demais companheiros pelo apoio e pela amizade.

Aos funcionários da UEPB, pela presteza e atendimento quando nos foi necessário.

Aos colegas de classe pelos momentos de amizade e apoio.

RESUMO

O presente trabalho tem o objetivo de implementar um modelo probabilístico previamente proposto nas referências bibliográficas, utilizando a ferramenta computacional R para desenvolver um pacote que automatiza a estimação dos parâmetros oriundos do modelo e realiza o cálculo de probabilidades dos desfechos em partidas de futebol. Adicionalmente, uma ferramenta que automatiza o processo da coleta de dados da *internet* foi desenvolvida utilizando a linguagem de programação Python e a biblioteca *Selenium*. Com uso dessas ferramentas, um experimento foi realizado para calcular as probabilidades do desfecho dos confrontos em três rodadas de um campeonato nacional na temporada de 2020, além das probabilidades de classificação e rebaixamento para a próxima fase, obtendo resultados satisfatórios. O Trabalho visou oferecer ferramentas que possam tornar o processo de análise e coleta de dados mais rápido e fácil, permitindo o uso de pessoas de todo o mundo para compreensão mais aprofundada do desempenho das equipes em jogos futuros, para pesquisas futuras e podendo até contribuir para a tomada de decisão de forma mais embasada no âmbito esportivo. Como conclusão, os resultados obtidos mostram que as ferramentas desenvolvidas proporcionam um cálculo das probabilidades dos desfechos das partidas de forma coerente, podendo ser útil para os profissionais da área, pesquisadores e amantes do esporte.

Palavras-Chave: programa R; python; modelo probabilístico.

ABSTRACT

The present work aims to implement a probabilistic model previously proposed in the literature, using the computational tool R to develop a package that automates the estimation of parameters from the model and calculates the probabilities of outcomes in soccer matches. Additionally, a tool that automates the process for data colecting from the internet was developed using the Python programming language and the Selenium library. Using these tools, an experiment was carried out to calculate the probabilities of the outcomes of matches in three rounds of a national championship in the 2020 season, as well as the probabilities of classification and relegation for the next phase, obtaining satisfactory results. The work aimed to offer tools that can make the process of analysis and data collection faster and easier, allowing people from all over the world to have a more in-depth understanding of team performance in future games, for future research, and even to contribute to decision-making in the sports field. In conclusion, the results obtained show that the developed tools provide a coherent calculation of the probabilities of match outcomes, which can be useful for professionals in the field, researchers, and sports enthusiasts.

Keywords: R software; python; probabilistic model.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Com o passar dos anos, a tecnologia da informação vem revolucionando muitos processos em diversas áreas do conhecimento, incluindo a modelagem estatística. Com o advento da computação, tornou-se possível criação e aplicação de diversas abordagens computacionais para todas as tarefas. Os *softwares* mais populares para modelagem estatística, como o R e o Python, possuem diversas opções de pacotes e funções capazes de automatizar tarefas importantes e analisar dados de forma mais eficiente. No entanto, existem poucos pacotes voltados para a modelagem probabilística em esportes.

Os pacotes são um conjunto de funções, que são compostas por conglomerados de códigos que visam facilitar e otimizar tarefas. Há vários pacotes criados para diversas tarefas, e o critério de escolha do pacote a ser utilizado é determinado pelos usuários. Esses pacotes são hospedados em alguma plataforma (ou repositório) externa e são instalados na máquina dos usuários por meio de artifícios da linguagem de programação utilizada. O GitHub é uma das plataformas mais utilizadas para hospedagem de projetos de código livre, utilizando a linguagem de versionamento de código chamada Git, é possível fazer vários procedimentos com o código do projeto, como transferência, manutenção, desenvolvimento e acompanhamento, possibilitando também a contribuição de pessoas autorizadas pelo autor do código. Com auxílio desses recursos aplicados à estatística, torna-se possível construir diversas aplicações oriundas da modelagem estatística, como por exemplo, voltadas para o esporte.

O esporte é um fenômeno global e possui uma legião de fãs nas mais diversas modalidades, em especial no futebol. E embora o futebol seja um esporte estratégico e emocionante, sua natureza imprevisível é uma das principais razões pelas quais ele é tão amado em todo o mundo.

Muitos fatores podem influenciar no resultado de uma partida de futebol, criando várias barreiras para o processo de modelagem probabilística, como o levantamento e coleta dos dados necessários. Devido a essas e outras complexidades, as ferramentas responsáveis por fornecer os dados necessários para a estimação dos parâmetros dos modelos (ou treinamento de modelos, nos casos de algoritmos de aprendizado de máquina) ainda são limitados.

Nesse trabalho, é proposta uma ferramenta implementada na linguagem de programação R com capacidade de fornecer dados atualizados utilizando como fonte uma página web. Além disso, uma segunda ferramenta capaz de utilizar os dados fornecidos para calcular as probabilidades dos desfechos dos jogos de futebol foi desenvolvida e implementada no *software* R. Essas ferramentas foram utilizadas em um experimento em parceria com o Globo Esporte da Paraíba ao qual foram calculadas as probabilidades do desfecho dos confrontos em três rodadas de um campeonato nacional na temporada de 2020, além das probabilidades de classificação e rebaixamento para a próxima etapa do campeonato.

2 MATERIAL E MÉTODOS

Neste capítulo, será apresentada a base teórica utilizada para construção do modelo estatístico utilizado para a estimação das probabilidades dos desfechos em jogos de futebol, bem como as ferramentas utilizadas para desenvolvimento do processo adotado para coleta e análise dos dados.

2.1 Modelagem aplicada ao futebol

O futebol é um esporte coletivo e tem suas partidas compostas por duas equipes com onze jogadores cada, onde a equipe a qual tiver o maior número de gols realizados na partida é o vencedor, tornando os gols intrinsecamente relacionados com o resultado da partida.

O placar das equipes em uma partida de futebol pode ser visto como um vetor aleatório bivariado ao qual, apoiando-se na literatura existente, o número de gols marcados por cada equipe em uma dada partida segue uma distribuição (univariada) de Poisson. A partir de métodos que permitam estimar as probabilidades da ocorrência de determinados placares, torna-se possível utilizar essas estimativas para calcular diferentes situações, como a probabilidade de vitória do time mandante sobre o time visitante, a chance de uma equipe ser campeã ou a quantidade de pontos necessários para um clube avançar para a próxima fase de um campeonato.

Nesse contexto, diversos trabalhos se propuseram a realizar modelagens probabilísticas com base nas distribuições bivariadas de Poisson, como pode ser visto em (DYTE e CLARKE, 2000) e (SUZUKI, 2007), por exemplo.

2.1.1 Distribuições de Poisson bivariadas

De acordo com ARRUDA (2000), diante das diversas classes de distribuições de Poisson bivariadas, não são todas que são aderentes a modelagem de resultados de partidas de futebol. Para que uma distribuição bivariada seja considerada adequada, algumas premissas devem ser satisfeitas:

- As distribuições marginais (gols marcados por cada equipe) devem ser Poisson.

- A distribuição conjunta deve possuir pleno suporte, ao menos perto da origem.
- A distribuição conjunta (e as marginais) devem ser infinitamente divisíveis. De acordo com Dwass e Teicher (1957), um vetor aleatório $\boldsymbol{X}$ é dito infinitamente divisível se, para qualquer inteiro positivo $\boldsymbol{n}$, $\boldsymbol{X}$ tem a mesma distribuição de uma soma de vetores $\boldsymbol{n}$ aleatórios independentes e identicamente distribuídos.

A classe de distribuição de Poisson bivariada “de Holgate” destaca-se por ser bastante adequada a modelagem probabilística dos resultados de partidas de futebol. Além de satisfazer todas as premissas citadas, ela é a única distribuição de Poisson bivariada infinitamente divisível (DWASS e TEICHER, 1957).

Neste trabalho utilizaremos os métodos propostos por ARRUDA (2000) e mais tarde adaptados por SUZUKI (2007), com base na distribuição bivariada de Poisson “de Holgate” para implementação de um pacote capaz de realizar os cálculos das probabilidades dos desfechos de partidas de futebol na ferramenta computacional R. Além disso, durante o trabalho, a nomenclatura utilizada para se referenciar aos jogos será sempre na forma “**Mandante (X) vs Visitante (Y)**”, onde o Mandante se refere a equipe que possui o mando de campo e o Visitante se refere a equipe que viajará para disputar a partida, e X e Y se referem a suas respectivas variáveis aleatórias que representam o número de gols marcados pelas duas equipes, respectivamente.

2.1.2 A distribuição de Poisson bivariada “de Holgate”

O modelo de Poisson bivariado proposto por Holgate (1964) tem como objetivo criar um vetor de variáveis aleatórias dependentes a partir de três ou mais variáveis aleatórias independentes (CHIRE, 2013). Sendo P_1, P_2, P_{12} variáveis aleatórias independentes com distribuição de Poisson univariadas com médias dadas respectivamente por: $\lambda_1 > 0,\ \lambda_2 > 0,\ \lambda_{12} > 0$, define-se então o vetor (X, Y) como:

$$X = P_1 + P_{12},$$
$$Y = P_2 + P_{12}$$

Possuindo distribuição bivariada “de Holgate”, onde $X \sim Poisson(\lambda_1 + \lambda_{12})$ e $Y \sim Poisson(\lambda_2 + \lambda_{12})$, e portanto sua função de probabilidade é dada por:

$$p(x,y) = P(X = x, Y = y)$$
$$= P(P_1 + P_{12} = x, P_2 + P_{12} = y)$$
$$= \sum_{i=0}^{\infty} P(P_1 + P_{12} = x, P_2 + P_{12} | P_{12} = i) \times P(P_{12} = i)$$
$$= \sum_{i=0}^{\infty} P(P_1 = x - i, P_2 = y - i) \times P(P_{12} = i)$$
$$= \sum_{i=0}^{\min(x,y)} P(P_1 = x - i) \times P(P_2 = y - i) \times P(P_{12} = i)$$
$$= \sum_{i=0}^{\min(x,y)} \frac{e^{-\lambda_1} {\lambda_1}^{x-i}}{(x-i)!} \times \frac{e^{-\lambda_2} {\lambda_2}^{y-i}}{(y-i)!} \times \frac{e^{-\lambda_{12}} {\lambda_{12}}^{i}}{i!} \Rightarrow$$
$$p(x,y) = e^{-(\lambda_1+\lambda_2+\lambda_{12})} \sum_{i=0}^{\min(x,y)} \frac{{\lambda_1}^{x-i} {\lambda_2}^{y-i} {\lambda_{12}}^{i}}{(x-i)!\,(y-i)!\,i!}$$

E seu valor esperado e variância dado por:

$$E[X] = Var[X] = \lambda_1 + \lambda_{12}$$
$$E[Y] = Var[Y] = \lambda_2 + \lambda_{12}$$

E covariância:

$$Cov[X,Y] = \lambda_{12}$$

Quanto a estimação dos parâmetros λ_1, λ_2 e λ_{12}, de acordo com ARRUDA (2000), a estimação desses parâmetros não se dá de forma direta, como por Mínimos Quadrados Ordinários (MQO) ou máxima verossimilhança, por exemplo. Essa estimação ocorre de forma indireta, em outras palavras, a estimação se dá por meio de funções paramétricas como $E[X+Y], E[(X+Y)^2]$ ou $P(X=0, Y=0)$.

Essa forma de estimação ocorre porque os métodos de estimação usuais dependem de amostras de variáveis aleatórias independentes e identicamente distribuídas (i.i.d). No caso de jogos de futebol, isso não é viável, pois mesmo em jogos distintos entre as mesmas equipes existem diferenças de comparabilidade (como o clima, árbitros, jogadores, etc.).

2.2 Métodos da Soma e Diferença

Os métodos da Soma e Diferença (SD0 e SD1) foram desenvolvidos por ARRUDA (2000) e adaptados por SUZUKI (2007) para fazer uso das propriedades da classe de Poisson Bivariada "de Holgate" e obter a probabilidade dos possíveis desfechos de uma partida de futebol.

A partir das propriedades do modelo de Poisson bivariado de Holgate, é fácil ver que o valor esperado da diferença de gols e da soma de gols, respectivamente, são dados por:

$$E[X-Y] = E[X] - E[Y] = (\lambda_1 + \lambda_{12}) - (\lambda_2 + \lambda_{12}) = \lambda_1 - \lambda_2$$
$$E[X+Y] = E[X] + E[Y] = (\lambda_1 + \lambda_{12}) - (\lambda_2 + \lambda_{12}) = \lambda_1 + \lambda_2 + 2\lambda_{12}.$$

Para o método SD0, admite-se a covariância nula e portanto, $\lambda_{12} = 0$. Em outras palavras, no contexto do futebol, há a admissão de independência entre as quantidades de gols marcados pelas equipes Mandante e Visitante, e respectivamente, entre as variáveis X e Y.

Com isso, pode-se obter a expressão dos valores estimados dos parâmetros de interesse para o método SD0 como:

$$\begin{cases} E[X-Y] = \lambda_1 - \lambda_2 \\ E[X+Y] = \lambda_1 + \lambda_2 \end{cases}$$

Que por sua vez, pode ser estimado por meio dos modelos lineares dados por:

$$(X+Y)_i = \boldsymbol{S}_i\alpha + \varepsilon_{ai} \qquad (1)$$
$$\text{e } (X-Y)_i = \boldsymbol{T}_i\beta + \varepsilon_{bi}, i = 1,2,3,\dots,n \qquad (2)$$

Em que, n é o número de jogos no banco de dados; ε_{ai} e ε_{bi} são erros independentes com média 0.

No modelo (1), $(X+Y)_i$ refere-se a soma do número de gols marcados por ambas as equipes no i-ésimo jogo; o vetor α é composto por $N+1$ parâmetros,

sendo um parâmetro para cada uma das N equipes na base de dados, mais um parâmetro associado ao local que será realizada a partida; a matriz $\boldsymbol{S}_i$ possui $N+1$ elementos, sendo N elementos associados ao *status* de cada equipe no jogo em questão, ou seja, 1 caso a equipe participe do jogo em questão e 0 caso contrário, mais um elemento referente ao local da partida, assumindo valores de 1 caso o jogo tenha ocorrido no campo do time Mandante e 0 caso a partida for em campo neutro ou estranho a ambas as equipes.

Já no modelo (2), $(X-Y)_i$ trata-se da diferença de gols marcados entre as equipes a favor do mandante no i-ésimo jogo em questão; o vetor β tem sua composição dada por $N+1$ parâmetros, sendo um parâmetro associado a cada uma das N equipes presentes na base de dados, mais um adicional referente ao tipo de local onde a partida acontece; Semelhante a matriz $\boldsymbol{S}_i$, a matriz $\boldsymbol{T}_i$ também possui $N+1$ componentes, N referentes ao *status* de cada equipe em relação a partida, assumindo valores de 1 caso a equipe seja o Mandante no jogo em questão, -1 se é visitante ou 0 se a dada equipe não participa do jogo, além de uma adicional referente ao local do jogo.

Além disso, a estimação dos parâmetros λ_1 e λ_2 são dados por:

$$\begin{cases} \hat{\lambda}_1 = \dfrac{\hat{E}[X-Y] + \hat{E}[X+Y]}{2} \\ \hat{\lambda}_2 = \dfrac{\hat{E}[X+Y] - \hat{E}[X-Y]}{2} \end{cases}$$

2.2.1 Método SD1

O método SD1 possui a mesma estruturação do método SD0, diferenciando-se apenas na estimação de covariância entre X e Y (λ_{12}), ao qual **não** é considerada nula no método SD1.

A partir das propriedades das variâncias e covariâncias dispostas no modelo de Holgate e dos resultados discutidos anteriormente, tem-se que:

$$E[(X+Y)^2] - E^2[X+Y] = Var[X+Y]$$
$$= Var[X] + 2Cov[X,Y] + Var[Y]$$

$$= \lambda_1 + \lambda_{12} + 2\lambda_{12} + \lambda_2 + \lambda_{12}$$
$$= \lambda_1 + 4\lambda_{12} + \lambda_2$$

E assim, temos o seguinte sistema de equações:

$$\begin{cases} E[X-Y] = \lambda_1 - \lambda_2 \\ E[X+Y] = \lambda_1 + \lambda_2 + 2\lambda_{12} \\ E[(X+Y)^2] - E^2[X+Y] = \lambda_1 + 4\lambda_{12} + \lambda_2. \end{cases} \quad \text{(a)}$$

Por sua vez, $E[X-Y], E[X+Y]$ tem estimação dada pelos modelos (1) e (2) (discutidos na sessão 2.2) e $E[(X+Y)^2]$ dado pelo modelo que segue:

$$\boldsymbol{S}_i\gamma + \varepsilon_{ci} \text{(3), onde } i = 1,2,3,\dots,n,$$

onde n é o número de jogos no banco de dados; ε_{ci} são os erros aleatórios e independentes com média 0.

Semelhante aos modelos apresentados na sessão anterior, no modelo (3), $[(X+Y)^2]_i$ é o quadrado da soma de gols marcados por ambas as equipes na i-ésima partida da base de dados; o vetor γ se compõe por $N+1$ parâmetros, sendo N parâmetros associados as N equipes na base de dados e um adicional referente ao local onde o jogo se realiza; A matriz $\boldsymbol{S}_i$ é a mesma definida no modelo (1);

Com base no sistema de equações (a), podemos obter as estimativas para os parâmetros λ_1, λ_2 e λ_{12} no método SD1 por:

$$\begin{cases} \hat{\lambda}_1 = \dfrac{\hat{E}[X-Y] + 2\hat{E}[X+Y] - \{\hat{E}[(X+Y)^2] - \hat{E}^2[X+Y]\}}{2} \\ \hat{\lambda}_2 = \dfrac{2\hat{E}[X+Y] - \hat{E}[X-Y] - \{\hat{E}[(X+Y)^2] - \hat{E}^2[X+Y]\}}{2} \\ \hat{\lambda}_{12} = \dfrac{\{\hat{E}[(X+Y)^2] - \hat{E}^2[X+Y]\} - \hat{E}[X+Y]}{2} \end{cases}$$

A partir do modelo discutido durante o trabalho e dos valores estimados dos parâmetros acima, torna-se possível estimar as probabilidades dos desfechos de futebol.

2.3 Ferramentas computacionais

O *software* R é uma das mais importantes e mais utilizadas ferramentas para análise de dados, além de uma linguagem de programação, se trata de um ambiente computacional que permite ao usuário realizar cálculos, simulações, construção de gráficos e desenvolvimento de modelos estatísticos. O R foi desenvolvido a partir de Ross Ihaka e Robert Gentleman na universidade de Auckland na Nova Zelândia, no entanto, ele é resultado dos esforços de toda a comunidade colaborativa ao redor do mundo. O software se encontra disponível de forma gratuita para os sistemas operacionais mais populares, como Windows, Linux e MacOs.

Além disso, o R possui um ambiente de desenvolvimento integrado chamado *RStudio*. O *RStudio* pode ser encontrado em duas versões: *RStudio Desktop*, para aqueles usuários que desejam executar os códigos localmente, e *RStudio Server*, que permite o acesso remoto de vários colaboradores dedicados a um mesmo projeto. O ambiente está disponível para os sistemas Windows, Linux e MacOS (MACHADO e BECHER, 2016).

O Python é outra linguagem de programação bastante utilizada por analistas em todo o mundo. Contando com uma comunidade bastante ativa e possuindo uma sintaxe muito acessível, o Python foi criado por um programador Holandês chamado Guido Van Rossum com objetivo de construir uma linguagem simples e de alto nível[1].

Na prática, as duas linguagens são muito boas para análise de dados, possuindo diversas funções prontas para facilitar o dia a dia de um analista. A escolha de qual linguagem de programação utilizar depende muito da preferência do usuário, em algumas situações é mais conveniente utilizar a linguagem Python devido a suas diversas aplicabilidades além do escopo das funções de análises, por exemplo.

O GitHub é uma plataforma de hospedagem de projetos muito utilizada para versionamento de código e colaboração entre os usuários. Além disso, programadores de todo o mundo usam o website do GitHub como portfólio dos seus projetos pessoais, podendo também ser uma opção de repositórios de ferramentas “independentes” nos

[1] As linguagens de programação podem ser classificadas de acordo com o grau de abstração que possuem. Assim sendo, existem três categorias de classificação: alto nível, médio nível e baixo nível. Quanto mais baixo o nível da linguagem, mais próximo das características da máquina (como o binário ou Assembly, por exemplo), ou seja, quanto menor o nível da linguagem mais difícil a sua compreensão para os humanos.

softwares R, Python, entre outros. A plataforma também possui sua própria linguagem de versionamento de código, semelhante a linguagem de programação, chamada Git.

O Git, de acordo com a documentação disponível, se trata de um sistema de controle de versão de código aberto que permite ao usuário gerenciar projetos de *software*. Por meio do Git, é possível monitorar o histórico de alterações à medida que as pessoas e equipes colaboram em projetos em conjunto.

2.3.1 Web Scraping

No âmbito do futebol, muitas vezes enfrenta-se o problema da falta de dados públicos disponíveis, sobretudo em ligas com menor visibilidade ou popularidade. Isso pode dificultar a obtenção das informações necessárias para realizar adequadamente o processo de modelagem, podendo também comprometer a qualidade e precisão dos resultados. A escassez de dados abertos e estruturados pode ser um obstáculo significativo para qualquer processo de análise estatística e criação de modelos preditivos que forneça resultados confiáveis, principalmente no contexto do esporte.

Uma técnica bastante útil para situações as quais não se possui uma base estruturada de dados, como no contexto citado, chama-se *Web Scraping*. *Web Scraping*, ou raspagem de dados, trata-se de uma técnica para extração de dados de forma automática usando como fonte as páginas web. Essa técnica possibilita extrair grandes volumes de dados sobre clientes, produtos, ou quaisquer outras informações a partir de um site ou canais de mídias sociais (BANDI, 2018).

Por ser um esporte bastante popular, existem diversos sites e mídias sociais que fornecem diversas atualizações sobre o futebol, o Globo Esporte por exemplo, é um dos principais veículos de informação sobre esportes no Brasil, disponibilizando diversas estatísticas, análises, entrevistas e reportagens sobre as equipes em seu site. Nesse contexto, o *Web Scraping* torna-se uma alternativa viável para a resolução do problema.

Existem algumas ferramentas que são utilizadas para esse propósito, cada uma contendo suas respectivas limitações. Uma das premissas que devem ser consideradas antes da escolha da ferramenta se dá ao formato da página, são elas:

- Site estático,
- Site dinâmico

Em poucas palavras, o site estático é aquele ao qual já se disponibiliza todas as informações presentes na página para o usuário, já o site dinâmico, torna-se necessário que o usuário interaja com a página para disponibilidade das informações (SOUL DIGITAL, 2018). Com base nas premissas citadas e com o propósito do experimento detalhado na sessão 3, a ferramenta escolhida para uso da técnica de raspagem de dados chama-se *Selenium*.

O *Selenium* é uma ferramenta proposta inicialmente para automação de testes para aplicações *web* que permite aos desenvolvedores simular o comportamento de um usuário na página, ou seja, a ferramenta permite simular a interação humana com navegadores web. Ele geralmente é utilizado por meio de linguagens de programação como Python, Java e C#. Ao simular a interação humana com a página, é possível coletar informações em sites que possuem elementos dinâmicos, como JavaScript, que é responsável pela dinamicidade da página, como botões, formulários e coleta de informações (MUTHUKADAN, 2018).

2.3.2 Pacote para coleta de dados: soccerdatas

Nessa primeira versão, o pacote implementado no *software R* foi desenvolvido com base na técnica de *Web Scraping* usando como base a ferramenta *Selenium* na linguagem Python, possibilitando a coleta das seguintes informações por meio das páginas do GE:

- Sigla da equipe mandante (**Home**);
- Número de gols da equipe mandante (**X**);
- Número de gols da equipe visitante (**Y**);
- Sigla da equipe visitante (**Visitor**);
- **Local**, sendo uma variável binária onde 1, caso o time mandante esteja jogando em casa e 0 caso contrário.

Atualmente disponível para os campeonatos nacionais série A e Paraibano.

A primeira etapa do processo utilizado no desenvolvimento da ferramenta se inicia no Python, usando os artifícios disponibilizados pelo *Selenium,* fazendo a coleta das informações diretamente pela página do GE referente ao respectivo campeonato,

realizando o batimento dos mandos de campo e distinção dos jogos passados e jogos futuros com base no seu placar (semelhante a divisão de bases de treino e teste, para modelos de *machine learning*) e devolve ao usuário uma planilha no formato csv com as informações expostas anteriormente.

Por meio de um pacote proposto por Allaire, Ushey e Tang (RStudio, 2021), chamado *reticulate*, é possível ter acesso a um conjunto de ferramentas para interoperabilidade entre o Python e o R, ou seja, a ferramenta permite ao programador a utilização do Python e do R ao mesmo tempo. Com base nisso, torna-se possível fazer a “tradução” de objetos Python para R, incorporando uma sessão Python em outra sessão R, incluindo ambientes virtuais e ambientes Conda.

A partir disso, a segunda etapa do processo se dá pela tradução das funções no ambiente R. Os códigos feitos em Python são executados em um ambiente virtual do R criado pelo *reticulate.* Nesse ambiente virtual são instaladas as bibliotecas necessárias para execução dos códigos Python, além de fornecer todos insumos necessários para execução da raspagem de dados, como o *driver* do navegador utilizado pelo *Selenium*, por exemplo.

O *driver* citado atua como uma interface entre o código do *Selenium* e o navegador, possibilitando o envio e a leitura dos resultados das ações programadas pelos desenvolvedores. Cada navegador possui um driver específico, nesse trabalho, o navegador utilizado será o ***FireFox,*** além disso, o *driver* do *Selenium* é responsável por iniciar e encerrar o navegador, bem como interagir com o site.

Portanto, deve-se destacar a necessidade do usuário possuir o navegador FireFox instalado em sua máquina local para que as funções disponíveis no pacote funcionem. Dito isso, abaixo detalho como utilizar as funções disponibilizadas pelo pacote:

- **set_env():** A função tem como objetivo preparar um ambiente virtual na linguagem R, ajustando a versão do Python, bem como as bibliotecas utilizadas para execução dos códigos para o *Web Scraping*. Portanto, é essencial executar essa função antes de realizar a coleta de dados das competições disponíveis. Além disso, a função não possui parâmetros para execução.
- **serie_a():** A função executa um código Python ao qual faz a coleta de dados da página https://globoesporte.globo.com/futebol/brasileirao-

serie-a/, devolvendo duas tabelas chamadas *'database'* e *'future_matches'*, referentes a objetos *dataframe* dos jogos que já ocorreram e dos jogos futuros, respectivamente.

- **PB():** A função executa um código Python ao qual faz a coleta de dados da página https://globoesporte.globo.com/pb/futebol/campeonato-paraibano/, devolvendo duas tabelas chamadas *'database'* e *'future_matches'*, referentes a objetos *dataframe* dos jogos que já ocorreram e dos jogos futuros, respectivamente.

2.3.3 Pacote para cálculo de probabilidades: SoccerProbs

Com base nesses conceitos, a versão atual do pacote tem sua composição dada por funções responsáveis por realizar os procedimentos essenciais para o cálculo das probabilidades. Abaixo descrevo as funções desenvolvidas:

- **crMatrixSD(data):** A partir de uma planilha pré-definida, a formata adequadamente para os cálculos a serem realizados pelas demais funções;
- **calcCoefSD1(data):** A partir da matriz de saída da função **crMatrixSD**, calcula os coeficientes utilizados no método SD1, obtendo como retorno uma matriz contendo em suas colunas os parâmetros estimados para α, β, γ;
- **calcLambdas(coef, home, visitor, local = 1):** É a função responsável por calcular os parâmetros $\lambda_1, \lambda_2, \lambda_{12}$ da distribuição de Poisson bivariada "de Holgate". Possui 4 parâmetros, 'coef' se refere a uma matriz de três colunas provindas da função **calcCoefSD1**; 'home' se refere a sigla da equipe mandante, 'visitor' se refere a sigla da equipe visitante e 'local' define o fator local para a partida, assume os valores de 1 e 0 (por padrão, 1). Seu retorno é um vetor de tamanho três com os parâmetros estimados da distribuição de Holgate.
- **calcMatProb(lambdas, dim.matrix=11):** Calcula as probabilidades para uma determinada partida de futebol. Possui 2 parâmetros, são eles: 'lambda', referente a um vetor de tamanho três contendo os parâmetros

da distribuição de Poisson bivariada de Holgate para o jogo (retorno da função **calcLambdas**); ‘dim.matrix’ se refere a dimensão da matriz (deve ser quadrada) a ser usada para calcular todas as probabilidades.

3 METODOLOGIA

Nessa sessão abordaremos a aplicação do pacote *SoccerProbs* para calcular as probabilidades dos desfechos de uma partida nos jogos das rodadas 16ª, 17ª e 18ª da série C do Campeonato Brasileiro de 2020, bem como as probabilidades de classificação e rebaixamento dos times.

3.1 Aplicação dos métodos

O pacote *SoccerProbs* teve uma importante aplicação na elaboração de um artigo para o Globo Esporte da Paraíba, cujo objetivo foi calcular as probabilidades de vitória do mandante, empate e vitória do visitante para as rodadas 16ª, 17ª e 18ª da série C do Campeonato Brasileiro de 2020. Os resultados obtidos foram satisfatórios e as probabilidades foram disponibilizadas ao público. Além disso, o pacote também permitiu a simulação das probabilidades de classificação para a próxima fase do campeonato e de rebaixamento para a série C de 2021, o que foi possível graças à técnica de *Web Scraping* discutida anteriormente para obtenção dos dados referentes aos jogos já realizados.

Os cálculos realizados pelo *SoccerProbs* foram apresentados nas Tabelas 1 e 2, que mostram as classificações anteriores à realização da rodada 16 para os clubes dos grupos A e B, respectivamente. Nessas tabelas, as probabilidades de classificação para a próxima fase e de rebaixamento para a série C de 2021 também foram apresentadas em forma de porcentagem. Adicionalmente, a Tabela 3 apresenta as probabilidades de desfecho de todos os jogos da rodada 16, também convertidas em porcentagem.

Em resumo, a utilização do pacote *SoccerProbs* e suas funções permitiram a obtenção de resultados consistentes, que foram disponibilizados para o público e contribuíram para uma cobertura mais detalhada e completa do Campeonato Brasileiro Série C na temporada de 2020. Além disso, o experimento gerou insumos para o desenvolvimento da técnica de *Web Scraping* no pacote *soccerdatas*.

A matéria realizada pelo Globo Esporte da paraíba pode ser vista no link disponível nas referências do trabalho.

4 RESULTADOS E DISCUSSÕES

Os cálculos realizados pelo *SoccerProbs* são ilustrados por meio dos resultados obtidos antes da 16ª rodada do Campeonato Brasileiro de 2020. As classificações anteriores à realização da rodada 16 para os clubes dos grupos A e B serão apresentadas nas tabelas abaixo:

Tabela 1 - Tabela prévia à rodada 16 para o grupo A da Série C.

Times	Pontos	Vitórias	Saldo	Gols	Prob-Class (%)	Prob-Reb (%)
SCZ	36	11	18	28	100,00	0,00
REM	26	7	8	18	99,88	0,00
VIL	24	6	6	15	98,51	0,00
PAY	22	6	7	21	77,41	0,00
MAN	20	4	1	15	9,21	1,17
FER	19	5	1	17	12,65	1,81
BOT	18	4	1	15	9,21	1,17
JAC	18	4	-3	15	1,07	28,44
TRZ	17	4	-1	15	0,19	56,17
IMP	1	0	-40	8	0,00	100,00

Fonte: Elaborado pelo autor, 2023.

Tabela 2 - Tabela prévia à rodada 16 para o grupo B da Série C.

Times	Pontos	Vitórias	Saldo	Gols	Prob-Class (%)	Prob-Reb (%)
BRU	28	8	6	20	99,25	0,00
LON	24	7	3	16	74,11	0,00
YPI	24	7	2	22	80,31	0,00
ITU	23	6	5	20	81,19	0,00
TOM	23	6	2	17	59,61	0,00
VRE	18	4	1	20	4,98	0,87
CRI	17	4	-3	14	0,46	6,17
SJO	16	4	-8	12	0,09	18,53
BOA	14	2	-4	13	0,00	81,94
SBE	13	2	-4	11	0,00	92,49

Fonte: Elaborado pelo autor, 2023.

As probabilidades de classificação para a próxima fase e de rebaixamento para a série C de 2021 são apresentadas nas duas últimas colunas dessas tabelas, em

porcentagem. Os times são identificados de acordo com o que era apresentado na página do Globo Esporte para o campeonato de 2020.

Na tabela abaixo são apresentadas as probabilidades (também convertidas em porcentagens) relacionadas aos resultados de todos os jogos da rodada 16. Os resultados destacados (em negrito) na Tabela 3 representam a intersecção de resultados esperados pelo modelo e resultados obtidos nas partidas.

Tabela 3 - Tabela de probabilidades associadas aos jogos da rodada 16.

Jogo	Vit. Mand. (%)	Empate (%)	Vit. Vis. (%)
IMP vs VIL	4,52	19,42	**76,05**
SCZ vs MAN	78,96	20,25	0,80
JAC vs TRZ	**52,39**	22,97	24,64
PAY vs FER	**55,55**	20,08	24,36
BOT vs REM	18,91	**63,62**	17,47
CRI vs YPI	46,58	19,00	34,42
ITU vs BOA	**66,09**	19,73	14,19
TOM vs BRU	**46,76**	21,94	31,30
SJO vs LON	16,59	**65,89**	17,52
VRE vs SBE	68,16	23,76	8,08

Fonte: Elaborado pelo autor, 2023.

Diante da Tabela 3, é possível perceber que os desfechos com maiores probabilidades realmente aconteceram em sete dos dez jogos da 16ª rodada. Um outro resultado interessante se dá pelo jogo entre SCZ vs MAN, em que o time visitante (MAN) venceu com uma probabilidade de apenas 0,80%, mostrando que surpresas podem ocorrer no futebol.

Além disso, o segundo pacote (*soccerdatas*), também demonstrou resultados interessantes ao oferecer a capacidade de obter dados dos campeonatos Paraibano e da série A do Brasileiro de forma automática e em poucos minutos. Isso é possível através da coleta direta dos dados na página do Globo Esporte, tornando todo o processo extremamente eficiente quanto a confiabilidade, amplitude de competições e atualização das partidas. O pacote *soccerdatas* também oferece compatibilidade com os principais sistemas operacionais como Linux, Windows e MacOS, permitindo a utilização por usuários de diferentes plataformas.

Para exemplificar o uso do pacote secundário, o código a seguir mostra como coletar os dados das partidas de uma das competições implementadas:

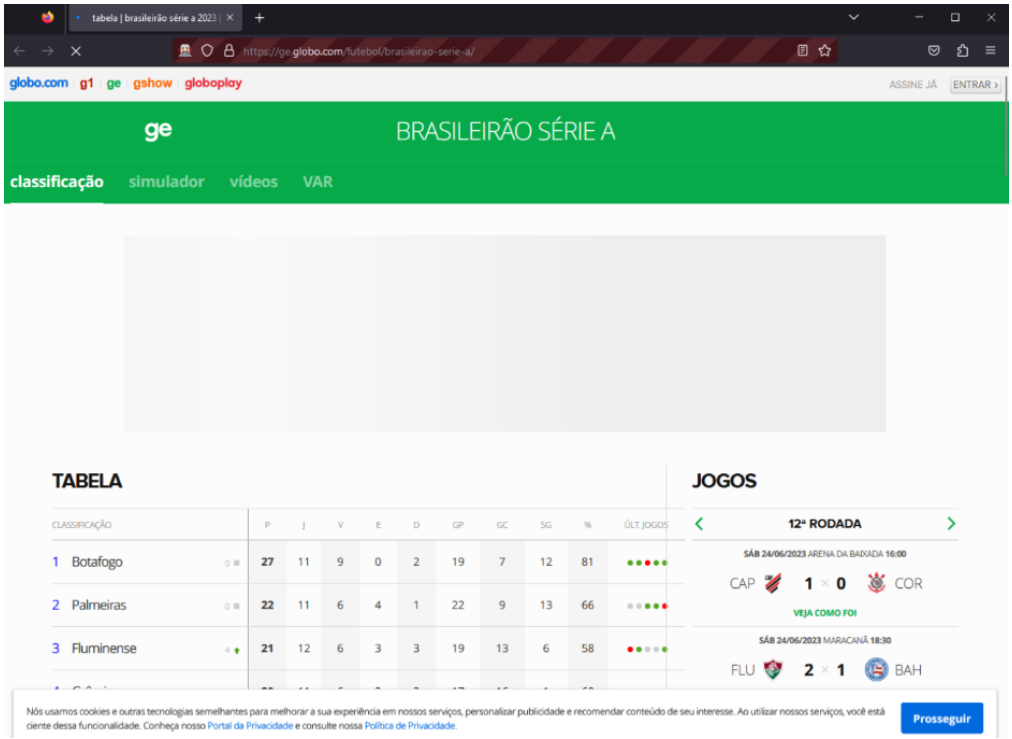
```
### Passo 1:
# Instalação do pacote ‘devtools’:
install.packages(‘devtools’)
library(devtools)

### Passo 2:
# Instalação do pacote ‘soccerdatas’:
install_github(‘samuel-souza/soccerdatas’)
library(soccerdatas)
```

Nos passos 1 e 2, são feitos os procedimentos necessários para instalação da ferramenta *soccerdatas*, sendo necessário também a instalação de suas devidas dependências[2]. No passo 3, a função *‘set_env’* cria um ambiente virtual, fazendo a configuração necessária para funcionamento dos códigos Python na interface R.

```
### Passo 4:
# Executar o comando para início da coleta dos dados:
serie_a()
```

Figura 1 - Exemplo de funcionamento



Fonte: Elaborado pelo autor, 2023.

[2] Dependências em pacotes no software R são outros pacotes ou bibliotecas que um pacote depende para seu funcionamento.

No passo 4, já é possível realizar a coleta de dados por meio das funções disponíveis no pacote, nesse caso, como exemplo, a competição escolhida será o Campeonato Brasileiro Série A (Brasileirão Série A). Após o comando ser executado, o navegador *FireFox* será aberto e direcionado a página da competição escolhida, realizando ações controladas pela ferramenta para coleta dos dados. Essa etapa deverá ocorrer como na Figura 1.

Nessa etapa, vale ressaltar que possíveis erros podem vir a ocorrer, os erros mais prováveis podem se dar por: Ter ocorrido alguma modificação na página utilizada para fazer a raspagem dos dados (incluindo cliques do usuário durante o uso da ferramenta), o usuário não ter o navegador *FireFox* instalado na máquina ou algum problema nos passos 1 e 2, geralmente envolvendo dependências.

```
### Passo 5:
# Visualizando os resultados:
head(database)
```

```
> ### Passo 5:
> # Visualizando os resultados:
> head(database)
  Home X Y Visitor Local
1  PAL 2 1     CUI     1
2  AME 0 3     FLU     0
3  BOT 2 1     SAO     1
4  RBB 2 1     BAH     0
5  CAP 2 0     GOI     1
6  FOR 1 1     INT     1
```

```
### Passo 6:
# Visualizando os resultados:
head(future_matches)
```

```
> ### Passo 6:
> # Visualizando os resultados:
> head(future_matches)
  Home  X  Y Visitor
1  PAL NA NA     BOT
2  GRE NA NA     CFC
3  SAN NA NA     FLA
4  RBB NA NA     GOI
5  AME NA NA     INT
6  VAS NA NA     CUI
```

Por fim, a função devolverá duas tabelas, sendo a tabela '*database*' referente aos jogos que já ocorreram e a tabela *'future_matches'* referente aos jogos que ainda não ocorreram.

5 CONCLUSÃO

A partir do experimento discutido na sessão anterior, pode-se concluir que as ferramentas desenvolvidas, e com o suporte teórico do material discutido no decorrer do trabalho, torna-se possível o cálculo automatizado das probabilidades no contexto do futebol de forma prática, podendo ser utilizada por qualquer torcedor curioso para verificar as chances do seu time em um campeonato.

Além disso, as ferramentas podem ser usadas como base para novas ideias e modelagens, e a atualização do código pode ser realizada conforme o avanço dos estudos sobre o assunto.

Os experimentos realizados com as ferramentas entregaram resultados satisfatórios, servindo como aprendizado tanto sobre as ferramentas de desenvolvimento computacional disponíveis quanto a respeito da modelagem estatística. No entanto, há necessidade de ampliação para outras competições, implementação de novas funções e uma medida para comparação de resultados entre diferentes modelos.

Por fim, o conhecimento adquirido e compartilhado durante o trabalho poderá ser utilizado por outros pesquisadores, colaboradores, curiosos e entusiastas no assunto, afim de desenvolver novas aplicações, abordagens teóricas e práticas no contexto do futebol.

REFERÊNCIAS

DYTE, D.; CLARKE, S. R. **A ratings based Poisson model for World Cup soccer simulation**. Journal of the Operational Research society, v. 51, n. 8, p. 993-998, 2000.

SUZUKI, A. K. **Modelagem estatística para a determinação de resultados de dados esportivos**. 2007. Citado nas páginas 15, 16 e 18.

ARRUDA, M. L. d. **Poisson, Bayes, Futebol e DeFinetti**. 2000. Tese de Doutorado. Universidade de São Paulo. Citado nas páginas 15, 16, 17 e 18.

DWASS, M. e TEICHER, H. (1957), **On Infinitely Divisible Random Vectors**, Annals of Mathematical Statististics, p.461-470.

CHIRE, V. A. Q. **Inferência em distribuições discretas bivariadas**. 2013.

MACHADO, L. B.; BECHER, E. L. **Aprendendo estatística com o software R**. Sociedade Brasileira de Educação Matemática. Encontro Nacional de Educação Matemática, 2016.

PRETATECH, **Linguagens de programação e suas classificações**. Medium, 2020. Disponível em: https://apretatech.medium.com/linguagens-de-programação-e-suas-classificações-708cfa69fa3a. Acesso em: 22 abr. 2023.

R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria, 2021. Disponível em: http://www.R-project.org/.

MCKINNEY, W. **Python for data analysis**. [S.l.]: O'Reilly, 2012. NULL. ISBN 9781449319793.

GITHUB. Sobre o Git, 2023. Disponível em: https://docs.github.com/pt/get-started/using-git/about-git. Acesso em: 22 abr. 2023

BANDI, A. **Web Scraping Using Selenium - Python**. Towards Data Science, 2018. Disponível em: https://towardsdatascience.com/web-scraping-using-selenium-python-8a60f4cf40ab. Acesso em: 22 abr. 2023.

DIGITAL, S. **Site estático e dinâmico, quais as diferenças?**. Medium, 2018. Disponível em: https://medium.com/@souldigitalbr/site-estático-e-dinâmico-quais-as-diferenças-2fb72c7bbbc2. Acesso em: 22 abr. 2023.

MUTHUKADAN, B. **Selenium with Python**, 2018. Disponível em: https://selenium-python.readthedocs.io. Acesso em: 22 abr. 2023.

USHEY, K; ALLAIRE, J; TANG, Y. ***reticulate: Interface to 'Python'***. 2023. Disponível em: https://rstudio.github.io/reticulate/, https://github.com/rstudio/reticulate. Acesso em: 22 abr. 2023.

GLOBOESPORTE. "**Brasileirão Série A**". Disponível em: https://globoesporte.globo.com/futebol/brasileirao-serie-a/. Acesso em: 29 abr. 2023.

GLOBOESPORTE. “**Campeonato Paraibano**”. Disponível em: https://globoesporte.globo.com/pb/futebol/campeonato-paraibano/. Acesso em: 29 abr. 2023.

Confira as chances de classificação e rebaixamento na Série C 2020, a quatro rodadas do fim da 1ª fase. **Globo Esporte**, Campina grande, 13, nov. 2020. Disponível em: https://ge.globo.com/pb/futebol/brasileirao-serie-c/noticia/confira-as-chances-de-classificacao-e-rebaixamento-na-serie-c-2020-a-quatro-rodadas-do-fim-da-1a-fase.ghtml. Acesso em: 29 abr. 2023.