

# **Métrica de Pressão Intrínseca e Extrínseca no Momento de Execução**

**Master em Big Data Aplicado ao Futebol**

**PROJETO FIM DE MASTER**



**Bernardo Nogueira Schlemper**

**Gustavo Corbellini Berardi**

**Martin Paré**

**Victor Augusto Savani Vergara**

## RESUMO:

O presente trabalho, apresentado junto ao Sports Data Campus em colaboração com o departamento de dados do Sporting Clube de Portugal, tem como orientação de desenvolvimento explorar a análise de dados no futebol, com foco na influência da pressão psicológica no desempenho dos jogadores.

Na concepção do problema tratado, com base em uma análise de dados que combina informações de *open data* e dados internos anonimizados fornecidos pelo Sporting CP, propõe-se um ponderador de contexto/pressão que considera tanto a pressão intrínseca quanto a extrínseca vivenciada pelos jogadores. Esse ponderador é desenvolvido a partir de diversos fatores que podem afetar o desempenho dos atletas, detalhadamente descritos e justificados. Em seguida, é realizada uma aplicação do ponderador para avaliar seu comportamento e impacto na atribuição de pontuações às ações dos jogadores.

A contribuição científica desta pesquisa é proporcionar uma abordagem inovadora para a compreensão da influência da pressão psicológica no desempenho dos jogadores de futebol, colaborando para o avanço da análise de dados e o desenvolvimento de estratégias mais eficazes para as equipes.

Palavras-chave: Análise de dados, Futebol, Pressão psicológica, Métrica.

## ABSTRACT:

This present work, presented in collaboration with the Sports Data Campus and the data department of Sporting Clube de Portugal, aims to explore data analysis in football, with a focus on the influence of psychological pressure on players' performance.

In addressing the problem, based on a data analysis combining open data and anonymized internal data provided by Sporting CP, a context/pressure weighting system is proposed, considering both intrinsic and extrinsic pressure experienced by the players. This weighting system is developed using various factors that can affect athletes' performance, thoroughly described and justified. Subsequently, the weighting system is applied to evaluate its behavior and impact on the assignment of scores to players' actions.

The scientific contribution of this research is to provide an innovative approach to understanding the influence of psychological pressure on football players' performance, contributing to the advancement of data analysis and the development of more effective strategies for teams.

Keywords: Data analysis, Football, Psychological Pressure, Metric.

## SUMÁRIO:

1. Introdução
  - 1.1. Contextualização
  - 1.2. Estado da arte
  - 1.3. Métricas Similares
2. Objetivos
3. Arquitetura conceitual e tecnológica do projeto
  - 3.1. Arquitetura conceitual
4. Metodologias e técnicas aplicadas
  - 4.1. Pressão Pré-Jogo
  - 4.2. Pressão Durante o Jogo
  - 4.3. Avaliação do Desempenho Psicológico
5. Desenvolvimento do trabalho
  - 5.1. Avaliação da Pressão
    - 5.1.1. Pressão Pré-Jogo
    - 5.1.2. Pressão Durante o Jogo
    - 5.1.3. Pressão Psicológica Final
  - 5.2. Go-live: fase de implementação
  - 5.3. Discussão dos resultados
6. Conclusões e trabalho futuro
  - 7.1. Conclusões
  - 7.2. Trabalhos Futuros
7. Bibliografia
8. Anexos

## LISTA DE FIGURAS:

Figura 1: Diagrama de Fluxo de Trabalho

## LISTA DE QUADROS:

Quadro 1: Número de finalizações de cada jogador em diferentes níveis de pressão

Quadro 2: Número de finalizações de cada jogador (com mais de 20 finalizações) em diferentes níveis de pressão

## LISTA DE GRÁFICOS:

Gráfico 1: Número de finalizações em diferentes níveis de pressão

Gráfico 2: Média de PSxG das finalizações (mais que 20) dos jogadores sob diferentes níveis de pressão

Gráfico 3: Média de xG das finalizações (mais que 20) dos jogadores sob diferentes níveis de pressão

Gráfico 4: Variação da média de xG e PSxG do jogador 431540 sob diferentes níveis de pressão

Gráfico 5: Variação da média de xG e PSxG do jogador 521814 sob diferentes níveis de pressão

Gráfico 6: Variação da média de xG e PSxG do jogador 559904 sob diferentes níveis de pressão

Gráfico 7: Variação da média de xG e PSxG do jogador 72331 sob diferentes níveis de pressão

Gráfico 8: Variação da média de xG e PSxG do jogador 551442 sob diferentes níveis de pressão

Gráfico 9: Mapa de finalizações dos jogadores 551442 e 72331

Gráfico 10: Variação da Pressão Média ao longo do campeonato

# **1. Introdução**

## **1.1. Contextualização**

A análise de desempenho no futebol é um processo multifatorial que envolve um estudo psicológico, técnico e tático, individual e coletivo. Devido ao avanço tecnológico é possível ter acesso a bases de dados e informações de jogadores, times e competições em grande escala com maior facilidade. O volume dos dados aumenta na mesma proporção da complexidade dos problemas. No entanto, no momento, a grande quantidade de informações disponíveis ainda é explorada de forma limitada (GARCÍA-ALIAGA, et al., 2020), em particular os fatores externos que afetam o desempenho dos jogadores durante uma partida.

Tendo em vista que a relevância do futebol pode ser medida pelo fato de ser o esporte mais popular em todo planeta (FIFA, 2006), os jogadores precisam lidar com pressões internas e externas ao longo do jogo. A pressão psicológica pode ser causada por diversos fatores, como a competição, o resultado do jogo, o adversário, o tempo restante, dentre outros. (LAGO; GOMÉZ, POLLARD, 2021).

Nesse contexto, este trabalho tem como objetivo explorar a análise de dados no futebol, discutindo os métodos, técnicas e ferramentas utilizados, bem como sua importância e aplicações práticas. Mais especificamente, será proposto o desenvolvimento de um ponderador de contexto (pressão, tanto intrínseca quanto extrínseca) para influir na pontuação atribuída a cada ação de jogadores de futebol. Será feita uma revisão da literatura sobre o tema, buscando entender como a pressão psicológica pode afetar o desempenho do jogador em campo e quais as possíveis consequências para o resultado da partida.

A compreensão dos efeitos da pressão na performance dos jogadores de futebol pode fornecer informações importantes para atletas, treinadores e outros setores de um clube. Essas informações podem ser utilizadas para ajudar a melhorar o desempenho dos jogadores, treinamento, decisões táticas, captação de atletas e maximizar as chances de sucesso no campo (BRANSEN, et al., 2019).

## **1.2. Estado da Arte**

Num primeiro momento, procura-se determinar os fatores que impactam o jogo de futebol em diferentes escalas, de modo que, em seguida, seja possível distinguir as variáveis

psicológicas que afetam o jogador de futebol numa partida. O desempenho dos atletas e a prática do futebol dependem de uma diversidade de fatores. A abordagem atual pode, então, ser relacionada à teoria da ecologia do desenvolvimento humano, que explica todos os principais fatores que afetam o desempenho, descritos pelo estudo de Bronfenbrenner (1979), adaptado por Gómez (2021), visando o desempenho de um jogador de futebol. Ao se considerar um cronossistema (*chronosystem*), que examina as influências do tempo no desenvolvimento do indivíduo e as mudanças que ocorrem no seu decorrer, nota-se que a performance é influenciada em diversas escalas. Em nível individual, restringe-se à tomada de decisão, à técnica, à tática e ao físico. Em uma escala micro, consideram-se os adversários, as condições coletivas, tais como estratégia e tática. Já em uma escala meso, as influências são provenientes das variáveis das condições da partida, ou seja, o ambiente, o local, o *status* do jogo, o estágio da competição, entre outros. Levando em consideração escalas exo e macro, os aspectos cambiantes que influenciam a performance de um jogador serão o sistema de competição e a ideologia/cultura do seu clube, respectivamente.

Em suma, como será exposto na Metodologia, seguindo essa ótica de *chronosystem*, foram designados os fatores que ponderam a pressão psicológica exercida sobre um atleta de futebol, levando em consideração as escalas individuais, meso, exo e macro.

### **1.3. Métricas Similares**

Seguindo o objetivo de avaliar a pressão intrínseca e extrínseca em uma partida de futebol, buscaram-se métricas que quantificassem circunstâncias similares. Num primeiro momento, o estudo voltou-se para métricas popularizadas no basquetebol, como o *Clutch Player*. Com isso, ao compreender, a partir do artigo “Measuring Clutch Play in the NBA” (INPREDICTABLE, 2014), a abordagem da importância do bom desempenho em momentos cruciais do jogo, conhecidos como “*clutch moments*”, foi proposta uma métrica para avaliar o desempenho dos jogadores nessas situações. O artigo em questão visa reelaborar o conceito da NBA, segundo o qual o “*clutch moment*” é definido como os últimos cinco minutos de um jogo em que a diferença de pontos é de 5 ou menos. Embora esta seja uma definição direta e razoável de situações *clutch*, fica claro que ela pode ser aprimorada. Utilizando dados estatísticos da NBA, os pesquisadores desenvolveram o “índice de *clutch*”, que representa a eficiência dos jogadores sob pressão. Os resultados revelaram diferenças significativas entre os jogadores nesses momentos, destacando a importância de habilidades como precisão nos arremessos e capacidade de criar jogadas.

Nesse contexto, foi feita, aqui, uma investigação em busca de métricas análogas empregadas no contexto do futebol, visando compreender a viabilidade da aplicação desse conceito. A partir de então, torna-se válido introduzir os artigos da plataforma “Analytics FC”: “Developing new metrics: “*Who are the most “clutch” goalscorers in football?*” (Analytics FC, 2019) e “Who are the most “clutch” creators in football?” (Analytics FC, 2021). Esses artigos foram fontes de motivação deste projeto, na medida em que uma parte das métricas desenvolvidas foram inspiradas na noção de “*Goal Importance*” do TransferLab. Essa métrica implica no fato de que, quando um gol é marcado, a probabilidade de vitória a partir desse gol pode ser calculada como a diferença entre a curva da pontuação anterior e a curva da nova pontuação, para o minuto em que o gol foi marcado.

Ademais, o trabalho “*Choke or Shine? Quantifying Soccer Players' Abilities to Perform Under Mental Pressure*” teve também uma importância fundamental no desenvolvimento conceitual deste projeto. Tem por objetivo oferecer uma compreensão clara de como os jogadores de futebol se comportam e se desempenham em situações de alta pressão mental. Para isso, é adotada uma abordagem que combina modelos de *Machine Learning* com características do contexto da partida, como rivalidade entre as equipes e a posição na tabela, juntamente com o estado atual do jogo, como por exemplo o placar e o tempo restante. Além disso, são desenvolvidos modelos para avaliar a escolha e a execução das ações dos jogadores, bem como a contribuição esperada dessas ações para o placar do jogo. No artigo, os *Event Data* de milhares de partidas em diferentes ligas foram considerados, e as métricas resultantes podem ser utilizadas pelas equipes para uma melhor avaliação do desempenho de seus jogadores, por exemplo, para fins de recrutamento e comparação com jogadores de outras equipes e ligas.



## **2. Objetivos**

Conforme descrito anteriormente, ao longo de uma temporada, campeonato ou partida, os times e os jogadores passam por diferentes fases e situações com as quais têm que lidar. E em decorrência disso, cada partida ou ação realizada em campo acontece em um determinado contexto. Uma chance de gol no começo do primeiro jogo da temporada não tem o mesmo impacto psicológico no jogador que uma chance semelhante durante um jogo no final do campeonato, valendo a liderança.

Posto isso, observou-se a necessidade de criar uma medida, um indicador que levasse em consideração esse fator da pressão sofrida pelos times e jogadores ao executarem suas ações ao longo de um campeonato ou de uma partida. Para a criação desse indicador, foram utilizados fatores que afetam diretamente na performance e que podem, portanto, influenciar na tomada de decisão.

O intuito deste trabalho é criar um ponderador de pressão, intrínseca e extrínseca, utilizando esses fatores, atribuindo um peso a cada um deles, a ser utilizado para ajustar a importância dos eventos, a depender do momento em que ocorrem.

### 3. Arquitetura conceitual e tecnológica do projeto

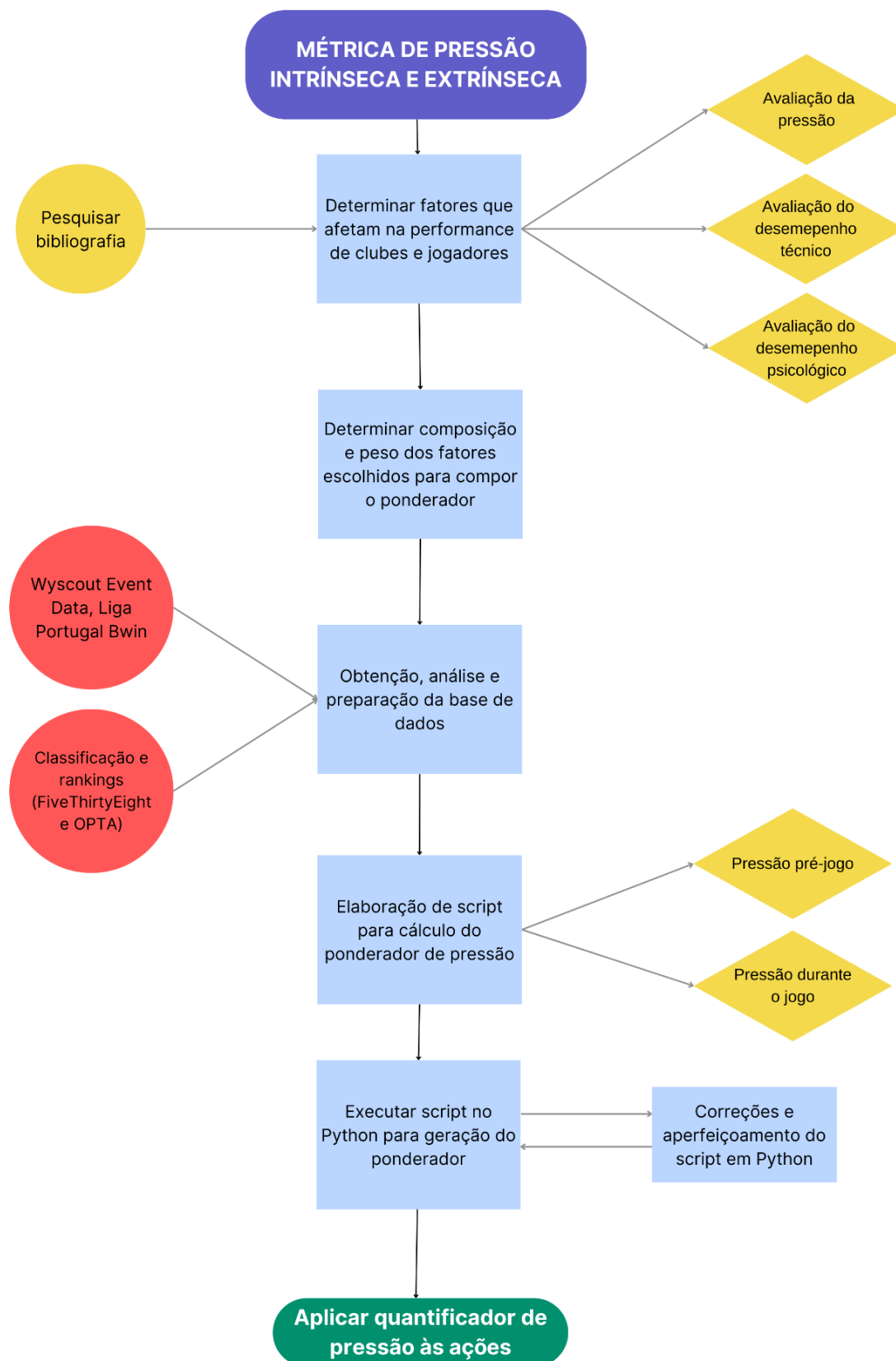
Para desenvolver um ponderador de pressão que influencia na pontuação de cada ação em uma partida, é necessário que se tenha uma base de dados com essas ações, assim como um ferramental que permita a utilização desses dados para se chegar ao objetivo final. A ferramenta utilizada neste trabalho foi o software Python.

A base de dados contendo os eventos das partidas foi cedida pelo Sporting Clube de Portugal, tratada e utilizada como objeto de aplicação principal do *script* elaborado no Python, visto que é nesta base em que estão os eventos a serem ajustados pelo ponderador a ser proposto.

Foram utilizadas também bases auxiliares, contendo informações fundamentais para determinar os fatores que compõem o ponderador. Essas bases foram as seguintes: uma tabela contendo todos os resultados da liga portuguesa na atual temporada; uma tabela com as pontuações das equipes portuguesas da primeira divisão no Opta Power Rankings (ranking desenvolvido pela OPTA que designa uma nota entre 0 e 100 para cerca de 13.500 equipes, sendo zero a nota da pior equipe ranqueada e 100 a nota da melhor equipe ranqueada); uma tabela com informações de público das equipes (capacidade do estádio, a atendêcia média e a taxa de ocupação); e uma tabela com o SPI (Soccer Power Index) e outras estatísticas calculadas pelo FiveThirtyEight para os clubes da primeira divisão portuguesa.

O fluxo de trabalho utilizado foi semelhante ao existente na metodologia CRISP-DM (Cross Industry Standard Process for Data Mining). Essa metodologia foi criada em 1996 e consiste em um conjunto de boas práticas a serem realizadas ao se executar um projeto em ciência de dados. De maneira resumida, esse método define seis passos para a análise de dados: entendimento do negócio; entendimento, preparação e modelagem dos dados; avaliação do modelo e execução.

O diagrama a seguir apresenta os passos seguidos neste trabalho para a obtenção do ponderador de pressão.



**Figura 1: Diagrama de Fluxo de Trabalho**

**Fonte: Autoria Própria**

### **3.1. Arquitetura Conceitual**

A arquitetura conceitual proposta para a métrica de desempenho de jogadores de futebol sob diferentes níveis de pressão psicológica consiste em três componentes inter-relacionados: avaliação da pressão, avaliação do desempenho técnico e avaliação do desempenho psicológico.

#### **Avaliação da Pressão:**

Nesta etapa, o objetivo é identificar e quantificar os diferentes níveis de pressão aos quais um jogador de futebol está exposto durante uma partida. Para isso, esta avaliação será dividida em dois períodos: a pressão pré-partida e a pressão durante a partida. Assim, serão quantificados os indicadores intrínsecos e extrínsecos, a fim de determinar os níveis de pressão para cada momento em que o atleta desempenha.

#### **Avaliação do Desempenho Técnico:**

Após estabelecer os níveis de pressão, é crucial entender o método de medição do desempenho do jogador em cada contexto. Isso pode ser feito utilizando métricas tradicionais, como passes completos, gols marcados, dribles bem-sucedidos, além de outras medidas específicas de desempenho para diferentes posições em campo. Entretanto, por tratar-se de um projeto em conjunto com o clube Sporting CP, será proposta uma adaptação para posterior integração a seus métodos de avaliação próprios. Para que seja possível entender a performance de cada atleta ou equipe em diferentes circunstâncias, é necessário compreender a sua variação ao decorrer de uma partida de futebol. É fundamental considerar, nesse sentido, a consistência do desempenho técnico em diferentes níveis de pressão para observar o impacto da pressão psicológica no jogador ou na equipe.

#### **Avaliação do Desempenho Psicológico:**

O terceiro componente da arquitetura é a avaliação do desempenho psicológico do jogador sob diferentes níveis de pressão. Isso envolve a análise e comparação do desempenho dos jogadores sob os diversos contextos evidenciados anteriormente. Este momento é fundamental para a transformação da informação em uma aplicação, de fato, prática. Desse modo, será possível empregar esta métrica em momentos de treinos, análises, scouting, estratégias, táticas, entre outras circunstâncias.

A integração dos três componentes permite uma compreensão abrangente do desempenho do jogador de futebol em diferentes contextos de pressão psicológica. Essa arquitetura conceitual oferece uma estrutura sólida para o desenvolvimento de uma métrica objetiva que considera tanto os aspectos técnicos quanto os psicológicos, proporcionando uma avaliação mais completa e precisa do desempenho de um jogador sob pressão intrínseca ou extrínseca.

## 4. Metodologias e técnicas aplicadas

### 4.1. Pressão Pré Jogo

No desenvolvimento da Análise de Pressão, considera-se a avaliação desta em um momento que precede a partida, a ser chamado, aqui, Pressão Pré-Jogo. Na elaboração dessa métrica, diversos fatores que impactam psicologicamente a performance de um jogador são ponderados, no intuito de quantificar seu desempenho em diferentes circunstâncias. Assim, constatarem-se os seguintes indicadores:

1. **Importância do jogo:** A relevância do jogo. Caracterizaram-se os momentos de riscos de eliminação, rebaixamento, chances de atingir competições continentais, como descrito no artigo “Measuring the importance of football match” (SCARF, SHI, 2008) como ponderadores do nível de pressão mental sobre os jogadores.

Atribuem-se valores de 1 a 5, sendo 5 a maior importância, decididos em consenso com profissionais do Sporting CP.

CrITÉRIOS Utilizados: Partida da fase de grupos (1), Partida decisiva de fase de grupos (2), Oitavas de final (2), Quartas de final (3), Semifinal (4), Final (5), Risco de Zona de Rebaixamento  $((5/n)*r)$ , Chance de classificação para competições continentais  $((5/n)*r)$ . Onde  $n$  e  $r$  são o número de rodadas da competição e a rodada em que está sendo jogada a partida, respectivamente. Além disso, propõe-se somar +1 aos confrontos dependendo do nível das rivalidades.

2. **Histórico do Confronto:** Também é necessário considerar o histórico recente do confronto, os últimos resultados entre as equipes.

Atribuem-se valores com base nos resultados recentes nos confrontos diretos.

Exemplo: Vitórias recentes contra o oponente (1), Resultados mistos recentes (mesmo número de pontos) (2), Derrotas recentes contra o oponente (4).

3. **Placar Parcial:** Esse fator só é relevante em partidas eliminatórias de ida e volta. Portanto, o placar parcial da primeira partida é de se levar em consideração na análise de pressão pré-jogo da partida de volta. Por exemplo, uma vitória ou uma derrota por

pouca diferença no placar pode ter um valor mais alto em comparação com uma vitória ou uma derrota significativa.

Atribuem-se valores com base nos diferentes placares, indicando a proximidade do resultado.

Exemplo: Vitória por pequena margem (1 gol de diferença) (2), Derrota por pequena margem (1 gol de diferença) (3), Vitória por margem significativa (2+ gols de diferença) (2), Derrota por margem significativa (2+ gols de diferença) (1).

4. **Qualidade do oponente:** O nível de qualidade da equipe adversária também pode afetar a pressão mental vivenciada pelos jogadores. Eles podem sentir mais pressão se enfrentarem uma equipe altamente qualificada. As equipes obtiveram notas de 1 a 5 (*elos*) como será descrito posteriormente.

Atribuem-se valores de 1 a 5, sendo 5 a maior qualidade.

Exemplo: Oponente bem mais fraco (2 pontos de *elo* a menos) (1), Oponente mais fraco (1 ponto de *elo* a menos) (2), Confronto equilibrado (*elos* iguais) (3), Oponente superior (1 ponto de *elo* a mais) (4), Oponente bem superior (2 pontos de *elo* a mais) (5).

5. **Apoio da torcida:** O nível de apoio da torcida também pode afetar a pressão mental vivenciada pelos jogadores. Os atletas podem sentir mais pressão se estiverem jogando diante de uma torcida hostil ou crítica, por mais difícil que seja quantificar esse fator. No intuito de quantificação, a capacidade do estádio, a taxa de ocupação e a assistência média foram tomados em consideração para o cálculo da média ponderada que será descrito na parte de desenvolvimento. Desse modo, foram criadas 3 categorias para classificar o nível de apoio da torcida, de acordo com as variáveis citadas.

Atribuem-se valores de 1 a 5, sendo 5 o maior nível de apoio.

Exemplo: Torcida neutra (1), Torcida apoiadora e entusiasmada (3), Torcida hostil e crítica (5).

6. **Jogar fora de casa:** Jogar fora de casa pode ser uma fonte de pressão mental adicional.

Atribuem-se o valor 0 para jogos em casa e 1 para jogos fora.

7. **Fase das Equipes:** O desempenho anterior também pode afetar o nível de pressão mental. Por exemplo, se uma equipe perder várias partidas seguidas, os jogadores podem sentir pressão adicional para terem um bom desempenho no jogo atual.

Atribuem-se valores com base em indicadores de desempenho recentes, analisando os pontos obtidos nos últimos 5 jogos realizados.

Exemplo: Excelente forma recente (Pontos > 10) e excelente forma recente do adversário (Pontos > 10) (3), Excelente forma recente (Pontos > 10) e forma recente aceitável do adversário (Pontos > 5) (2), Excelente forma recente (Pontos > 10) e forma recente fraca do adversário (Pontos <= 5) (1), Forma recente aceitável (Pontos > 5) e excelente forma recente do adversário (Pontos > 10) (4), Forma recente aceitável (Pontos > 5) e forma recente aceitável do adversário (Pontos > 5) (2), Forma recente aceitável (Pontos > 5) e forma recente fraca do adversário (Pontos <= 5) (1), Forma recente fraca (Pontos <= 5) e excelente forma recente do adversário (Pontos > 10) (5), Forma recente fraca (Pontos <= 5) e forma recente aceitável do adversário (Pontos > 5) (4), Forma recente fraca (Pontos <= 5) e forma recente fraca do adversário (Pontos <= 5) (3).

Assim, como dito anteriormente, será proposta uma média ponderada para a métrica tal como foi decidido em consenso com profissionais do Sporting CP.

Importância do jogo: 35%

Histórico do Confronto: 10%

Placar Parcial: 10% ou 0% (caso não houver)

Qualidade do Oponente: 15%

Apoio da torcida: 10%

Jogar fora de casa: 5%

Fase das equipes: 15%



Em casos de cálculos para jogos de pontos corridos os valores de 10% serão distribuídos de maneira equânime para os demais fatores.

Em seguida, os valores obtidos serão normalizados para que seja possível realizar uma comparação mais justa, eliminando vieses de escala e para uma interpretação simplificada.

Portanto, após o cálculo da média ponderada (MP), evidenciado anteriormente, o valor final normalizado de cada jogo (VF) será calculado a partir da fórmula:

$$VF = (MP - MIN) / (MAX - MIN)$$

```
def normalize_metric(metric):
    min_metric = 0.61
    max_metric = 4.85
    normalized_metric = (metric - min_metric) / (max_metric -
min_metric)
    return normalized_metric

normalized_metric = normalize_metric(metric)
print("The pre-game mental pressure normalized metric is:
{:.3f}".format(normalized_metric)) # Final
```

## 4.2. Pressão Durante o Jogo

Após ser quantificada a Pressão Pré-Jogo, propõe-se ponderar a quantidade de pressão psicológica sofrida pelos atletas durante a partida. Para isso, foi implementado um algoritmo que calcula a pressão em um determinado momento do jogo com base nas probabilidades de vitória e empate para diferentes diferenças de gols. Ou seja, definiram-se os cenários mais críticos da partida, os momentos em que um gol marcado teria um maior impacto no resultado final. Logo, nas circunstâncias em que um jogo esteja empatado ou com 1 gol de diferença de placar, conjunturas psicologicamente mais críticas se estabelecem. Além disso, considera-se a variação das probabilidades ao longo do tempo, permitindo uma análise mais dinâmica da partida.

Para realizar este estudo, foram necessários dados contendo informações sobre as equipes, seus ratings e suas habilidades ofensivas e defensivas, para que as probabilidades de resultado variassem conforme o confronto. Os dados foram obtidos a partir do *dataframe* da FiveThirtyEight, que contém informações sobre as equipes e suas respectivas estatísticas. Esses dados foram utilizados como entrada para o modelo de previsão utilizado aqui.

Assim, com base nas habilidades ofensivas e defensivas das equipes são calculadas as médias de gols esperados para cada uma delas, na partida em questão. Em seguida, aplica-se a distribuição de Poisson para gerar as probabilidades de cada equipe marcar um determinado número de gols durante a partida.

A distribuição de Poisson é uma distribuição de probabilidade discreta, amplamente utilizada para modelar eventos aleatórios que ocorrem em uma taxa média conhecida. Isso porque essa distribuição expressa a probabilidade de um ou mais eventos discretos acontecerem dentro de um determinado período contínuo. Neste trabalho, utilizou-se a distribuição de Poisson para modelar o processo de geração de gols (eventos discretos) em uma partida (período de tempo contínuo).

Portanto, para calcular a probabilidade de vitória, empate e derrota, as etapas anteriores permitem determinar probabilidades, considerando todas as combinações possíveis de gols para cada equipe ao comparar os resultados para determinar os diferentes cenários.

Finalmente, a fim de quantificar a pressão psicológica efetiva, indica-se um minuto e uma diferença de gols, e calcula-se a pressão nesse momento específico do jogo. A pressão é determinada pela diferença das probabilidades de vitória e empate. Ela é calculada através da subtração da probabilidade de vitória pela probabilidade de vitória caso um gol seja marcado, e também da subtração da probabilidade de empate pela probabilidade de empate caso um gol seja marcado. Em seguida, a pressão é multiplicada pelo valor do minuto indicado, dividido pelo tempo total da partida, no intuito de representar a relação Tempo Restante x Pressão de maneira linear.

O código será dividido em várias funções para melhor organização e reutilização. Ao aplicar as técnicas estatísticas e implementar as funções correspondentes, poderá ser criado um modelo robusto e preciso para prever uma quantificação da pressão psicológica.

### **4.3 Avaliação do Desempenho Psicológico**

Finalmente, após a unificação desses dois momentos de pressão intrínseca e extrínseca, propõe-se o cálculo da pressão psicológica para todas ações e eventos do Sporting CP na Liga Portugal Bwin 2022/2023. Para isso, foram necessárias implementações no código, de maneira a automatizar processos que antes eram feitos pontualmente. Assim, foi necessário obter os gols e os minutos em que foram marcados, em cada partida, e outras alterações que serão descritas no item a seguir.

## 5. Desenvolvimento do trabalho

Neste capítulo, as funções, instruções e códigos implementados serão detalhados com base nas ideias elaboradas. O objetivo é fornecer uma visão geral e completa das etapas realizadas durante o desenvolvimento do projeto.

### 5.1. Avaliação da Pressão

#### 5.1.1. Pressão Pré-Jogo

A estrutura e o processo de desenvolvimento para a elaboração de um sistema de análise e determinação da pressão psicológica que precede, mais especificamente, as partidas de futebol da Liga Portuguesa são apresentadas aqui. É necessário notar que este modelo também é aplicável a qualquer outro contexto de partidas de futebol. As etapas de coleta de dados, pré-processamento, análise exploratória, cálculos estatísticos e modelagem preditiva utilizadas são descritas em detalhe.

A coleta de dados foi realizada a partir de diversas fontes, incluindo bancos de dados públicos, APIs e arquivos CSV. O principal conjunto de dados utilizado, nesse momento, foi o da Liga Portuguesa, que contém informações abrangentes sobre jogos, equipes e resultados. Além disso, foram utilizados dados adicionais, como classificação das equipes e rankings de desempenho providos pela FiveThirtyEight e Opta.

Antes de iniciar a análise, os dados coletados foram pré-processados e limpos para garantir qualidade e integridade. Foram realizadas etapas de tratamento de valores ausentes, padronização de formatos de data e nome das equipes, correção de inconsistências e normalização de dados. Abaixo, ilustra-se o momento de inicialização do código, a importação e adaptações nas bases de dados já tratadas.

```
import pandas as pd
import sys
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from datetime import datetime, timedelta

#connect code to google drive
```

```

from google.colab import drive
drive.mount('/content/drive')

#import crowd data
crowd_support_df = pd.read_excel(r'/content/Atendência Liga
Portuguesa.xlsx')
#import match data
LigaPTResults = pd.read_excel(r'/content/LigaPTResults.xlsx')
#import team quality 538
spi_portugal = pd.read_excel(r'/content/spi_portugal.xlsx') # pra
#import team quality Opta
Opta_Power_Ranking = pd.read_excel(r'/content/Opta_Power_Ranking.xlsx')
#put the 2 of them together
spi_portugal['rating'] =
spi_portugal['name'].map(Opta_Power_Ranking.set_index('name')['rating']
)
spi_portugal['elo'] = round((spi_portugal['spi'] * 2 +
spi_portugal['rating']) / 3, 2)
spi_portugal['elo_normalized'] = round((spi_portugal['elo'] -
spi_portugal['elo'].min()) / (spi_portugal['elo'].max() -
spi_portugal['elo'].min()), 3)
spi_portugal['elo_category'] = pd.cut(spi_portugal['elo_normalized'],
bins=[-0.1, 0.2, 0.4, 0.6, 0.8, 1], labels=[1, 2, 3, 4, 5])
spi_portugal_ranked = spi_portugal.sort_values(by='elo',
ascending=False)
spi_portugal

```

A partir desse momento, propõe-se ao beneficiário da análise fornecer um time e a data do jogo em questão a fim de descobrir o nível de pressão pré-jogo, na partida em foco. Ademais, definiu-se o dia anterior para o cálculo da tabela de resultados elaborado na sequência.

```

# Example
team = "Sporting"
date = "26/05/2023"
date = pd.to_datetime(date, format='%d/%m/%Y')

```

```
# Calculate the day before the given date
previous_day = date - timedelta(days=1)
```

### Função para revelar o adversário

A primeira função desenvolvida tem o intuito de revelar o adversário da partida designada. A função “find\_opposition” recebe o nome de uma equipe (team), uma data e o dataframe “LigaPTResults” como entrada. A função localiza a linha correspondente à equipe e à data fornecidas no dataframe “LigaPTResults”. Em seguida, verifica se a equipe fornecida corresponde à equipe da coluna “home.team” ou “away.team”. Caso ela corresponda à coluna “home.team”, retorna o valor da coluna “away.team” como “opposition”. Caso contrário, retorna o valor da coluna “home.team” como “opposition”.

```
def find_opposition(team_name, date, LigaPTResults):
    date = pd.to_datetime(date, format='%d/%m/%Y')
    # Find the row that matches the date and team name
    row = LigaPTResults[(LigaPTResults['date'] == date) &
((LigaPTResults['home.team'] == team_name) |
(LigaPTResults['away.team'] == team_name))]

    # Check if the team is in the home team column
    if row['home.team'].item() == team_name:
        opposition = row['away.team'].item()
    else:
        opposition = row['home.team'].item()

    return opposition

opposition = find_opposition(team, date, LigaPTResults)
```

### Cálculo da Métrica de Pressão Pré-Jogo

Neste momento, é desenvolvida a função determinante efetiva para o cálculo final da pressão pré-jogo. A função “calculate\_mental\_pressure\_metric” recebe o nome de uma equipe (team), o dataframe “LigaPTResults” e a data (date) como entrada. A função busca os jogos anteriores da equipe na coluna “home.team” e “away.team” do dataframe “LigaPTResults” e calcula a métrica de pressão mental. A métrica é determinada com base

em todos os critérios que serão descritos posteriormente. A função retorna o valor da métrica de pressão mental como resultado de uma média ponderada, como exposto na metodologia.

```
def calculate_mental_pressure_metric(importance, scoreline,
opposition_quality, crowd_support, away_game, team_form,
recent_encounters):
    # Pesos
    weights = {
        'importance': 0.35,
        'scoreline': 0.1,
        'opposition_quality': 0.15,
        'crowd_support': 0.1,
        'away_game': 0.05,
        'team_form': 0.15,
        'recent_encounters': 0.1
    }

    # Divisão equânime dos pesos caso não seja partida eliminatória ou
    análise individual de jogador: scoreline or recent_performance is 0
    if scoreline == 0:
        non_zero_weights = [w for w in weights if w not in
['scoreline']]
        adjusted_weight_sum = sum(weights[w] for w in non_zero_weights)
        adjusted_weight = adjusted_weight_sum / len(non_zero_weights)

        weights['scoreline'] = adjusted_weight
    # Média Ponderada
    weighted_sum = (importance * weights['importance']) + (scoreline *
weights['scoreline']) + \
        (opposition_quality * weights['opposition_quality'])
+ \
        (crowd_support * weights['crowd_support']) + \
        (away_game * weights['away_game']) + \
        (team_form * weights['team_form']) + \
        (recent_encounters * weights['recent_encounters'])

    return weighted_sum
```

## Cálculo da Tabela da Liga

Foi desenvolvida uma função para calcular a tabela da Liga Portuguesa com base nos resultados dos jogos. Essa função recebeu como entrada o conjunto de dados da Liga, juntamente com a data específica até a qual a tabela deveria ser calculada. A partir dos resultados dos jogos, a função realizou cálculos estatísticos das equipes, tais como o número de jogos disputados, vitórias, empates, derrotas, gols marcados, gols sofridos, saldo de gols e pontos acumulados.

A função avalia o resultado de cada partida com base na coluna “winner”, da mesma forma que na função “team\_form”, que será vista posteriormente. Os pontos são atribuídos às equipes de acordo com o resultado de cada partida.

A tabela foi então ordenada com base em critérios de classificação, como pontos, vitórias, diferença de gols e gols marcados, usando a função “sort\_values”. Esse processo forneceu uma visão atualizada da classificação das equipes na Liga Portuguesa.

Então, foram calculadas duas tabelas: uma para a data do jogo em questão, outra para o dia anterior do jogo. Este passo foi fundamental para a quantificação da importância do jogo, uma vez que a função exige uma diferenciação desses dois momentos.

```
def calculate_league_table(date, LigaPTResults):  
    # Filter the results dataframe based on the specified date  
    date = pd.to_datetime(date, format='%d/%m/%Y')  
    filtered_results = LigaPTResults[LigaPTResults['date'] <= date]  
  
    # Create an empty table with columns for team, games, wins, draws,  
    losses, points, goals scored, goals conceded, and goal difference  
    table = pd.DataFrame(columns=['Team', 'Games', 'Win', 'Draw',  
    'Loss', 'Points', 'Goals Scored', 'Goals Conceded', 'Goal Difference'])  
  
    # Iterate over each row in the filtered results dataframe  
    for index, row in filtered_results.iterrows():  
        home_team = row['home.team']  
        away_team = row['away.team']  
        home_goals = row['home.goals']  
        away_goals = row['away.goals']
```



```

# Update the table with the match results for the home team
if home_team not in table['Team'].values:
    table = pd.concat([table, pd.DataFrame([[home_team, 0, 0,
0, 0, 0, 0, 0]], columns=table.columns)])
if away_team not in table['Team'].values:
    table = pd.concat([table, pd.DataFrame([[away_team, 0, 0,
0, 0, 0, 0, 0]], columns=table.columns)])

    table.loc[table['Team'] == home_team, 'Goals Scored'] +=
home_goals
    table.loc[table['Team'] == home_team, 'Goals Conceded'] +=
away_goals
    table.loc[table['Team'] == home_team, 'Goal Difference'] +=
home_goals - away_goals

    table.loc[table['Team'] == away_team, 'Goals Scored'] +=
away_goals
    table.loc[table['Team'] == away_team, 'Goals Conceded'] +=
home_goals
    table.loc[table['Team'] == away_team, 'Goal Difference'] +=
away_goals - home_goals

# Update the Games, Wins, Draws, Losses, and Points columns for
both teams
table.loc[table['Team'] == home_team, 'Games'] += 1
table.loc[table['Team'] == away_team, 'Games'] += 1

if home_goals > away_goals:
    table.loc[table['Team'] == home_team, 'Win'] += 1
    table.loc[table['Team'] == home_team, 'Points'] += 3
    table.loc[table['Team'] == away_team, 'Loss'] += 1
elif home_goals < away_goals:
    table.loc[table['Team'] == away_team, 'Win'] += 1
    table.loc[table['Team'] == away_team, 'Points'] += 3
    table.loc[table['Team'] == home_team, 'Loss'] += 1
else:
    table.loc[table['Team'] == home_team, 'Draw'] += 1
    table.loc[table['Team'] == home_team, 'Points'] += 1

```

```

        table.loc[table['Team'] == away_team, 'Draw'] += 1
        table.loc[table['Team'] == away_team, 'Points'] += 1

    # Calculate the goal difference column
    table['Goal Difference'] = table['Goals Scored'] - table['Goals
Conceded']

    # Sort the table in descending order of points
    table = table.sort_values(by=['Points'],
ascending=False).reset_index(drop=True)

    return table

league_tablebefore = calculate_league_table(previous_day,
LigaPTResults)
league_tableafter = calculate_league_table(date, LigaPTResults)

# Sort the dataframe by the specified columns in descending order
league_tablebefore = league_tablebefore.sort_values(by=["Points",
"Win", "Goal Difference", "Goals Scored"], ascending=False)
league_tableafter = league_tableafter.sort_values(by=["Points", "Win",
"Goal Difference", "Goals Scored"], ascending=False)

# Reset the index of the dataframe
league_tablebefore = league_tablebefore.reset_index(drop=True)
league_tableafter = league_tableafter.reset_index(drop=True)

# Add the position column
league_tablebefore["Position"] = league_tablebefore.index + 1
league_tableafter["Position"] = league_tableafter.index + 1

if len(league_tablebefore) < 18:
    league_tablebefore = league_tableafter.copy()

```

## Cálculo do Valor de Importância

Foi implementada uma função para calcular o valor da importância de uma partida para uma designada equipe. Essa função recebeu como entrada o nome da equipe, a data do jogo, as tabelas da Liga Portuguesa calculadas e os resultados dos jogos. Com base na posição atual da equipe na tabela, a função verificou se a equipe tinha a possibilidade de subir ou descer na classificação, em cenários mais críticos.

A função calcula o valor de importância com base nas instruções fornecidas. Primeiro, verifica se a equipe pode subir para a 6<sup>a</sup>, 5<sup>a</sup>, 4<sup>a</sup>, 3<sup>a</sup>, 2<sup>a</sup> ou 1<sup>a</sup> posição se vencer ou empatar. Em seguida, verifica se a equipe pode descer para a 16<sup>a</sup>, 17<sup>a</sup> ou 18<sup>a</sup> posição, caso não vença. A função considera também as circunstâncias em que a equipe tem chance de perder as primeiras colocações ou de sair da zona de rebaixamento. Se as condições forem atendidas, a função retorna o valor calculado. O valor de importância do jogo foi calculado com base no número de jogos disputados até a data fornecida. Portanto, quanto mais próximo do final do campeonato, maior será a pressão existente para o jogo em questão. Além disso, foram adicionadas condições especiais para rivalidades entre equipes específicas, resultando em um ajuste adicional no valor de importância.

```
def calculate_importance_value(team, date, league_tablebefore,
league_tableafter, LigaPTResults, opposition):
    date = pd.to_datetime(date, format='%d/%m/%Y')

    team_positionbefore =
league_tablebefore.loc[league_tablebefore['Team'] == team,
'Position'].values[0]

    team_pointsbefore =
league_tablebefore.loc[league_tablebefore['Team'] == team,
'Points'].values[0]

    team_gamebefore = league_tablebefore.loc[league_tablebefore['Team']
== team, 'Games'].values[0]

    team_positionafter =
league_tableafter.loc[league_tableafter['Team'] == team,
'Position'].values[0]

    team_pointsafter = league_tableafter.loc[league_tableafter['Team']
== team, 'Points'].values[0]

    team_gamesafter = league_tableafter.loc[league_tableafter['Team']
== team, 'Games'].values[0]
```

```

derbi = 0

# Check if the opposition matches any of the specified rivalries
rivalries = [
    ("Porto", "Benfica"),
    ("Porto", "Sporting"),
    ("Benfica", "Sporting"),
    ("Braga", "Vitória"),
    ("Porto", "Boavista")
]

for rivalry in rivalries:
    if (team == rivalry[0] and opposition == rivalry[1]) or (team
== rivalry[1] and opposition == rivalry[0]):
        derbi += 1

position2 = list(range(team_positionbefore + 1, 12))
position3 = list(range(10, team_positionbefore - 1))

# Check if the team can go up in position
for position in [6, 5, 4, 3, 2, 1]:
    if team_positionbefore > position:
        competitors_points =
league_tablebefore.loc[league_tablebefore['Position'] == position,
'Points'].values
        if team_pointsbefore + 1 >= competitors_points.min() or
team_pointsbefore + 3 >= competitors_points.min():
            return ((5 / 34) * team_gamesafter) + derbi

# Check if the team can go down in position
else:
        competitors_points =
league_tablebefore.loc[league_tablebefore['Position'].isin(position2),
'Points'].values
        if competitors_points.max() + 1 >= team_pointsbefore or
competitors_points.max() + 3 >= team_pointsbefore:
            return ((5 / 34) * team_gamesafter) + derbi

# Check if the team can go down in position

```

```

for position in [16, 17, 18]:
    if team_positionbefore < position:
        competitors_points =
league_tablebefore.loc[league_tablebefore['Position'] == position,
'Points'].values
        if team_pointsbefore < competitors_points.max() + 1 or
team_pointsbefore < competitors_points.max() + 3:
            return ((5 / 34) * team_gamesafter) + derbi
            # Check if the team can go up in position
        else:
            competitors_points =
league_tablebefore.loc[league_tablebefore['Position'].isin(position3),
'Points'].values
            if competitors_points.min() < team_pointsbefore + 1 or
competitors_points.min() < team_pointsbefore + 3:
                return ((5 / 34) * team_gamesafter) + derbi

        return ((1 / 34) * team_gamesafter) + derbi # If the team doesn't
meet the criteria for going up or down

```

## Cálculo do Histórico do Confronto

A função “calculate\_recent\_encounters” recebe o nome de duas equipes (*team* e *opposition*) e o dataframe “LigaPTRResults” como entrada. A função busca no dataframe os jogos mais recentes entre as duas equipes e realiza uma análise estatística desses encontros. A análise envolve a contagem de vitórias, empates e derrotas de cada equipe nos jogos recentes, fornecendo uma visão do desempenho histórico e da dinâmica dos confrontos entre as equipes. Essas estatísticas podem ser utilizadas para avaliar a rivalidade, o equilíbrio ou a superioridade de uma equipe sobre a outra. A função retorna as estatísticas dos encontros recentes, permitindo uma avaliação mais precisa do histórico e do relacionamento competitivo entre as equipes.

```

Confronto_Direto = pd.DataFrame({
    'Team': [team, opposition],
    'Games': [0, 0],
    'Win': [0, 0],
    'Draw': [0, 0],

```

```

        'Lost': [0, 0],
        'Points': [0, 0]
    })

def calculate_recent_encounters(team, opposition, LigaPTResults,
                                Confronto_Direto, date):
    # Create a copy of the Confronto_Direto dataframe
    confronto_direto_copy = Confronto_Direto.copy()

    # Filter the dataframe based on team and opposition
    team_matches = LigaPTResults[
        (((LigaPTResults['home.team'] == team) &
        (LigaPTResults['away.team'] == opposition)) |
        ((LigaPTResults['home.team'] == opposition) &
        (LigaPTResults['away.team'] == team))) &
        (pd.to_datetime(LigaPTResults['date'], format='%d/%m/%Y') <
        pd.to_datetime(date, format='%d/%m/%Y'))
    ]

    if len(team_matches) == 0:
        return 1

    # Count the number of matches and update the Confronto_Direto
    dataframe copy
    matches_count = len(team_matches)
    confronto_direto_copy.loc[confronto_direto_copy['Team'] == team,
    'Games'] += matches_count
    confronto_direto_copy.loc[confronto_direto_copy['Team'] ==
    opposition, 'Games'] += matches_count

    # Update the points based on the match results
    for index, match in team_matches.iterrows():
        if match['winner'] == 0:
            confronto_direto_copy.loc[confronto_direto_copy['Team'] ==
            match['away.team'], 'Draw'] += 1
            confronto_direto_copy.loc[confronto_direto_copy['Team'] ==
            match['home.team'], 'Draw'] += 1

```

```

        confronto_direto_copy.loc[confronto_direto_copy['Team'] ==
match['away.team'], 'Points'] += 1
        confronto_direto_copy.loc[confronto_direto_copy['Team'] ==
match['home.team'], 'Points'] += 1
        elif match['winner'] == match['teamId.home']:
            confronto_direto_copy.loc[confronto_direto_copy['Team'] ==
match['home.team'], 'Win'] += 1
            confronto_direto_copy.loc[confronto_direto_copy['Team'] ==
match['home.team'], 'Points'] += 3
            confronto_direto_copy.loc[confronto_direto_copy['Team'] ==
match['away.team'], 'Lost'] += 1
        elif match['winner'] == match['teamId.away']:
            confronto_direto_copy.loc[confronto_direto_copy['Team'] ==
match['away.team'], 'Win'] += 1
            confronto_direto_copy.loc[confronto_direto_copy['Team'] ==
match['away.team'], 'Points'] += 3
            confronto_direto_copy.loc[confronto_direto_copy['Team'] ==
match['home.team'], 'Lost'] += 1

    # Verify if there are matching rows before accessing the points
values
    if team in confronto_direto_copy['Team'].values and opposition in
confronto_direto_copy['Team'].values:
        team_points =
confronto_direto_copy.loc[confronto_direto_copy['Team'] == team,
'Points'].values[0]
        opposition_points =
confronto_direto_copy.loc[confronto_direto_copy['Team'] == opposition,
'Points'].values[0]
        if team_points == opposition_points:
            return 2 # Draw
        elif team_points > opposition_points:
            return 1 # Team 1 has more points
        else:
            return 4 # Team 2 has more points
    else:
        return 1 # Default value if there are no matching rows

```

## Análise de Fase das Equipes

Foi desenvolvida uma função para avaliar a fase das equipes com base em seus desempenhos recentes. A função recebeu como entrada os dados da equipe definida e de seu oponente. Utilizou-se os pontos acumulados pelas equipes nas últimas partidas como métrica de desempenho. Classificamos as equipes em categorias, com base na combinação dos pontos das duas equipes, como exposto na metodologia deste trabalho, permitindo uma análise comparativa dos desempenhos.

Portanto, a função “team\_form” recebe o nome de uma equipe e uma data como entrada e retorna um *dataframe* chamado “team\_form” com as colunas “Team”, “Games”, “Win”, “Draw”, “Lost” e “Points”. Esta função busca os últimos 5 jogos da equipe, cujo nome é igual a “home.team” ou “away.team” na coluna “LigaPTResults” e cuja data é inferior à data fornecida na entrada.

A função avalia o resultado de cada partida com base na coluna “winner”. Se “winner” for igual a “0”, é considerado um empate e adiciona +1 aos pontos na coluna “Points” do *dataframe* “team\_form”. Se “winner” for igual ao valor da coluna “teamId.home” e a equipe do *dataframe* “team\_form” na coluna “Team” for igual à coluna “home.team” do *dataframe* “LigaPTResults”, adiciona +3 pontos ao *dataframe* “team\_form” e conta como uma vitória. Se “winner” for igual ao valor da coluna “teamId.away” e a equipe do *dataframe* “team\_form” na coluna “Team” for igual à coluna “away.team” do *dataframe* “LigaPTResults”, adiciona +3 pontos ao *dataframe* “team\_form” e conta como uma vitória. Caso contrário, considera como uma derrota e não adiciona pontos na coluna “Points”.

```
def get_team_form(team, date, LigaPTResults):
    date = pd.to_datetime(date, format='%d/%m/%Y')
    # Filter the dataframe based on team and date
    team_matches = LigaPTResults[(LigaPTResults['home.team'] == team) |
    (LigaPTResults['away.team'] == team)]
    team_matches = team_matches[team_matches['date'] < date]

    # Sort the matches by date in descending order and select the last
    5 matches
    team_matches = team_matches.sort_values('date',
    ascending=False).head(5)
```



```

        # Create the team_form dataframe with columns: "Team", "Games",
"Win", "Draw", "Lost", "Points"
        team_form = pd.DataFrame({'Team': [team],
                                'Games': [0],
                                'Win': [0],
                                'Draw': [0],
                                'Lost': [0],
                                'Points': [0]})

        # Update the team_form dataframe based on the match results
        for index, match in team_matches.iterrows():
            if match['winner'] == 0:
                team_form['Draw'] += 1
                team_form['Points'] += 1
                elif match['winner'] == match['teamId.home'] and
team_form['Team'].iloc[0] == match['home.team']:
                team_form['Win'] += 1
                team_form['Points'] += 3
                elif match['winner'] == match['teamId.away'] and
team_form['Team'].iloc[0] == match['away.team']:
                team_form['Win'] += 1
                team_form['Points'] += 3
            else:
                team_form['Lost'] += 1

        # Update the Games column
        team_form['Games'] = team_form['Win'] + team_form['Draw'] +
team_form['Lost']

        return team_form

team_form = get_team_form(team, date, LigaPTResults)
opposition_form = get_team_form(opposition, date, LigaPTResults)

def calculate_form_value(team_form, opposition_form):
    team_points = team_form['Points'].values[0]
    opposition_points = opposition_form['Points'].values[0]

```

```

if team_points >= 10:
    if opposition_points > 10:
        return 3
    elif opposition_points > 5:
        return 2
    else:
        return 1
elif team_points > 5:
    if opposition_points > 10:
        return 4
    elif opposition_points > 5:
        return 2
    else:
        return 1
else:
    if opposition_points > 10:
        return 5
    elif opposition_points > 5:
        return 4
    else:
        return 3

```

## Análise da Qualidade do Oponente

Ao receber o *dataframe* “spi\_portugal” e o *dataframe* “Opta\_Power\_Ranking” como entrada, adicionou-se a coluna “rating” do *dataframe* “Opta\_Power\_Ranking” ao *dataframe* “spi\_portugal” quando os valores das colunas “name” são iguais.

Em seguida, calculou-se a coluna “elo”. Os valores são calculados usando a fórmula  $(\text{“spi”} * 2 + \text{“rating”}) / 3$ . A coluna “elo” é adicionada ao *dataframe* “spi\_portugal”. Os valores são normalizados usando a fórmula  $(x - \min) / (\max - \min)$ , em que x é o valor original, min é o valor mínimo da coluna “elo” e max é o valor máximo da coluna “elo”.

Criou-se a coluna “elo\_category” que atribui valores de 1 a 5 às equipes com base nos valores da coluna “elo\_normalized”. As categorias são divididas em intervalos de 0.2, começando em 0.2. Por exemplo, se “elo\_normalized” for menor ou igual a 0.2, “elo\_category” será igual a 1. Se “elo\_normalized” for menor ou igual a 0.4, “elo\_category” será igual a 2, e assim por diante.

A função “opposition\_quality\_value” recebe o dataframe “spi\_portugal” e os nomes de duas equipes (team e opposition) como entrada. Ela compara os valores da coluna “elo\_category” no dataframe “spi\_portugal” para as equipes team e opposition. Com base nas comparações, a função retorna valores de 1 a 5. Se “elo\_category” de team for maior ou igual a “elo\_category” de opposition + 2, retorna 1. Se “elo\_category” de team for maior ou igual a “elo\_category” de opposition + 1, retorna 2. Se “elo\_category” de team for igual a “elo\_category” de opposition, retorna 3. Se “elo\_category” de team for menor ou igual a “elo\_category” de opposition - 1, retorna 4. Se “elo\_category” de team for menor ou igual a “elo\_category” de opposition - 2, retorna 5.

```
def opposition_quality(spi_portugal, team, opposition):
    team_elo_category = spi_portugal.loc[spi_portugal['name'] == team,
    'elo_category'].values[0]
    opposition_elo_category = spi_portugal.loc[spi_portugal['name'] ==
    opposition, 'elo_category'].values[0]

    if team_elo_category >= opposition_elo_category + 2:
        return 1
    elif team_elo_category >= opposition_elo_category + 1:
        return 2
    elif team_elo_category == opposition_elo_category:
        return 3
    elif team_elo_category <= opposition_elo_category - 2:
        return 5
    elif team_elo_category <= opposition_elo_category - 1:
        return 4
    else:
        return None
```

### Cálculo do Nível de Apoio da Torcida

A função “calculate\_crowd\_support” recebe o nome da equipe adversária (opposition), o dataframe “LigaPTResults”, a data do jogo (date) e o dataframe “crowd\_support\_df” como entrada. A função calcula o nível de apoio da torcida para a equipe em questão. Ela usa dados do dataframe “crowd\_support\_df” para obter informações sobre a capacidade do estádio, a média de atendimento e a taxa de ocupação. Com base

nesses dados, a função calcula uma média ponderada para determinar o nível de apoio da torcida. O nível de apoio é categorizado em valores de 1 a 5, dependendo do resultado do cálculo ponderado. A função retorna o valor do nível de apoio da torcida.

```
def calculate_crowd_support(opposition, LigaPTResults, date,
crowd_support_df):
    # Filter the dataframe based on the home_team_name
    team_row = crowd_support_df[crowd_support_df['TIME'] == opposition]
    date = pd.to_datetime(date, format='%d/%m/%Y')

    if len(team_row) == 0:
        # If the team is not found in the dataframe, return a default
value (e.g., neutral crowd)
        print("Time não encontrado.")
        return None

    # Calculate the weighted average
    capacidade_weight = 0.1
    atendimento_weight = 0.4

    weighted_avg = (
        (team_row['CAPACIDADE'] * capacidade_weight +
team_row['ATENDÊNCIA MÉDIA'] * atendimento_weight) *
        team_row['TAXA DE OCUPAÇÃO']
    )

    # Determine the crowd support value based on the weighted average,
since the Portuguese league has 18 teams
    # we divided it into 3 categories of 6 stadiums. These results are
exclusive for the Portuguese league.
    crowd_support_value = None

    if len(LigaPTResults.loc[(LigaPTResults['date'] == date) &
(LigaPTResults['away.team'] == opposition)]) > 0:
        crowd_support_value = 1 # Home crowd
    else:
        if weighted_avg.apply(lambda x: x >= 1100).any():
            crowd_support_value = 5 # Hostile crowd
```

```

elif weighted_avg.apply(lambda x: x >= 370).any():
    crowd_support_value = 3 # Enthusiastic crowd
else:
    crowd_support_value = 1 # Neutral crowd

return crowd_support_value

```

## Jogo Dentro ou Fora de Casa

A função “check\_team\_home\_or\_away” recebe o nome de uma equipe (*team*), a data (*date*) e o dataframe “LigaPTResults” como entrada. A função realiza uma verificação no *dataframe* para determinar se a equipe fornecida joga em casa ou como visitante na data especificada. Para isso, a função compara o nome da equipe com os valores das colunas “home.team” e “away.team” do dataframe “LigaPTResults”. Caso a equipe seja identificada como mandante, a função retorna o valor 1, indicando que a equipe joga em casa. Caso contrário, a função retorna o valor zero, indicando que a equipe joga como visitante.

```

def check_team_home_or_away(team, date, LigaPTResults):
    date = pd.to_datetime(date, format='%d/%m/%Y')
    # Filter the dataframe based on the specified team name and date
    home_match = LigaPTResults[(LigaPTResults['home.team'] == team) &
(LigaPTResults['date'] == date)]
    away_match = LigaPTResults[(LigaPTResults['away.team'] == team) &
(LigaPTResults['date'] == date)]

    if not home_match.empty:
        return 0 # Team played as the home team on the given date
    elif not away_match.empty:
        return 1 # Team played as the away team on the given date
    else:
        return -1 # Team did not play on the given date

```

## Aplicação Definitiva da Pressão Pré-Jogo

```

# Exemplo de Valores
importance_value = calculate_importance_value(team, date,
league_tablebefore, league_tableafter, LigaPTResults,opposition)

```

```

scoreline_value = 0 # From 1 to 3 (0 if it is not a knockout 2nd leg
match)
opposition_quality_value = opposition_quality(spi_portugal, team,
opposition)
crowd_support_value = calculate_crowd_support(opposition,
LigaPTRResults, date, crowd_support_df)
away_game_value = check_team_home_or_away(team, date, LigaPTRResults)
team_form_value = calculate_form_value(team_form, opposition_form)
recent_encounters_value = calculate_recent_encounters(team, opposition,
LigaPTRResults, Confronto_Direto, date)

# Cálculo da métrica

metric = calculate_mental_pressure_metric(importance_value,
scoreline_value, opposition_quality_value,
crowd_support_value,
away_game_value,
team_form_value,
recent_encounters_value)

print(f"The importance value for {team} on {date} is
{importance_value}") # tópico 1
print("The recent encounters value for the teams is:",
recent_encounters_value) # tópico 2
print("Opposition Quality Value:", opposition_quality_value) # tópico 4
print("The crowd support value for the game is:", crowd_support_value)
# tópico 5
print("The away game value for the game is:", away_game_value) # tópico
7
print("The teams form for the game team is:", team_form_value) # tópico
8
print("The pre-game mental pressure metric is: {:.3f}".format(metric))
# Final

```

### 5.1.2. Pressão Durante o Jogo

Dando prosseguimento à elaboração do código, descreve-se a metodologia adotada e as técnicas aplicadas para desenvolver o modelo de previsão de resultados de partidas de futebol. O objetivo deste estudo é fornecer uma abordagem baseada no *rating* “elo” das equipes e suas habilidades ofensivas e defensivas, usando uma distribuição de Poisson para calcular as probabilidades de vitória, empate e derrota.

O primeiro passo na geração das probabilidades de vitória, empate e derrota é o cálculo do número de gols esperados para cada equipe. Para isso, utilizamos as habilidades ofensivas e defensivas de cada equipe, representadas pelos valores “off” e “def” do *dataframe*.

```
def extract_team_data(team_name):

    team_data = spi_portugal[spi_portugal['name'] == team_name]
    elo_rating = team_data['spi'].values[0]
    offensive_ability = team_data['off'].values[0]
    defensive_ability = team_data['def'].values[0]
    return elo_rating, offensive_ability, defensive_ability

opposition = find_opposition(team, date, LigaPTResults)

elo1, off1, def1 = extract_team_data(team)
elo2, off2, def2 = extract_team_data(opposition)
```

Na função “calculate\_expected\_goals”, através do cálculo das médias entre as habilidades ofensivas de uma equipe e as habilidades defensivas da equipe adversária, os parâmetros “lambda1” e “lambda2” foram determinados, e representam o número médio de gols esperados para cada equipe. Esses parâmetros são utilizados na distribuição de Poisson para gerar as probabilidades de cada equipe marcar um determinado número de gols.

```
def calculate_expected_goals(elo1, elo2, off1, off2, def1, def2):
    # Calculate the expected number of goals for each team
    lambda1 = (off1 + def2) / 2
    lambda2 = (off2 + def1) / 2
```

```
# Return the expected number of goals
return lambda1, lambda2
```

Com os gols esperados para cada equipe calculados, foram percorridas todas as possibilidades de gols para cada equipe, considerando valores inteiros, e calculou-se a probabilidade de ocorrência de cada resultado possível. Portanto, a função “calculate\_poisson\_probability” calcula a probabilidade de ocorrência de um determinado número de gols para uma distribuição de Poisson, dada uma taxa média (lambda) e um número específico de gols (k).

```
def calculate_poisson_probability(lambda_val, k):
    # Calculate the Poisson probability for a given lambda value and k
    (number of goals)
    probability = (math.exp(-lambda_val) * (lambda_val ** k)) /
    math.factorial(k)
    return probability
```

As probabilidades de vitória, empate e derrota são calculadas comparando o número de gols marcados por cada equipe. Se a equipe 1 marcar mais gols que a equipe 2, a probabilidade de vitória é incrementada com base nas probabilidades calculadas pela distribuição de Poisson para os gols esperados de cada equipe. Da mesma forma, se a equipe 2 marcar mais gols, a probabilidade de derrota é incrementada. Se as equipes marcarem a mesma quantidade de gols, a probabilidade de empate é incrementada.

```
def calculate_win_draw_loss_probabilities(elo1, elo2, off1, off2, def1,
def2, max_goals1, max_goals2, goal_diff):
    lambda1, lambda2 = calculate_expected_goals(elo1, elo2, off1, off2,
def1, def2)

    # Calculate win/draw/loss probabilities
    win_prob = 0.0
    draw_prob = 0.0
    loss_prob = 0.0

    for goals1 in range(max_goals1 + 1):
```



```

        for goals2 in range(max_goals2 + 1):
            if goals1 + goal_diff > goals2:
                win_prob += calculate_poisson_probability(lambdal,
goals1) * calculate_poisson_probability(lambda2, goals2)
            elif goals1 + goal_diff < goals2:
                loss_prob += calculate_poisson_probability(lambdal,
goals1) * calculate_poisson_probability(lambda2, goals2)
            else:
                draw_prob += calculate_poisson_probability(lambdal,
goals1) * calculate_poisson_probability(lambda2, goals2)

        return win_prob, draw_prob, loss_prob

```

Com as probabilidades de vitória, empate e derrota calculadas, tornou-se possível calcular a pressão em um determinado minuto de jogo. A pressão é uma medida da importância do resultado em um momento específico do jogo.

Para calcular a pressão, considerou-se um valor X que varia dependendo da diferença de gols. Se a diferença de gols for menor que zero, X é calculado somando o módulo das diferenças das probabilidades de vitória e empate, caso um gol seja sofrido ou marcado naquele instante da partida. Se a diferença de gols for maior ou igual a zero, X é calculado somando o módulo da diferença das probabilidades de vitória, caso um gol seja sofrido ou marcado naquele instante da partida. A pressão é então calculada multiplicando X pelo valor do minuto do jogo dividido por 98,66. Desse modo, é possível traduzir um aumento linear da pressão ao decorrer do tempo de jogo. O valor 98,66 representa a duração total de um jogo de futebol genérico em minutos. Este valor, em particular, foi obtido em uma palestra no Sports Data Forum 2023, regida por Luis Dias, analista do SC Braga, em que ele compartilha que o tempo útil de jogo médio na Liga de Portugal (98,66 minutos) é o maior em comparação com as demais ligas europeias.

```

def calculate_goal_difference(goals1, minutes1, goals2, minutes2,
specific_minute):

    # Find the number of goals scored by each team before the specific
minute

```

```

    goals1_before = sum([1 for minute in minutes1 if minute <=
specific_minute])
    goals2_before = sum([1 for minute in minutes2 if minute <=
specific_minute])

    # Calculate the goal difference
    goal_diff = goals1_before - goals2_before

    return goal_diff

goal_diff = calculate_goal_difference(team1_goals, team1_minutes,
team2_goals, team2_minutes, specific_minute)

def pressure_minute(minute, goal_diff):

    # Calculate the win, draw, loss probabilities for the given goal
difference

    win_prob, draw_prob, loss_prob =
calculate_win_draw_loss_probabilities(elo1, elo2, off1, off2, def1,
def2, max_goals1, max_goals2, goal_diff)

    if goal_diff < 0:
        # Calculate X for goal difference less than 0
        X = abs(win_prob - calculate_win_draw_loss_probabilities(elo1,
elo2, off1, off2, def1, def2, max_goals1, max_goals2, goal_diff -
1)[0]) + \
            abs(win_prob - calculate_win_draw_loss_probabilities(elo1,
elo2, off1, off2, def1, def2, max_goals1, max_goals2, goal_diff +
1)[0]) + \
            abs(draw_prob - calculate_win_draw_loss_probabilities(elo1,
elo2, off1, off2, def1, def2, max_goals1, max_goals2, goal_diff -
1)[1]) + \
            abs(draw_prob - calculate_win_draw_loss_probabilities(elo1,
elo2, off1, off2, def1, def2, max_goals1, max_goals2, goal_diff +
1)[1])
    else:
        # Calculate X for goal difference greater than or equal to 0

```

```

        X = abs(win_prob - calculate_win_draw_loss_probabilities(elo1,
elo2, off1, off2, def1, def2, max_goals1, max_goals2, goal_diff -
1)[0]) + \
            abs(win_prob - calculate_win_draw_loss_probabilities(elo1,
elo2, off1, off2, def1, def2, max_goals1, max_goals2, goal_diff +
1)[0])
    # Calculate the pressure
    pressure = X * minute / 98.66
    return pressure

```

Em suma, foi descrita a metodologia adotada e as técnicas aplicadas para desenvolver o modelo de previsão de resultados de partidas de futebol. Utilizou-se os *ratings* “elo” das equipes, suas habilidades ofensivas e defensivas, e a distribuição de Poisson para calcular as probabilidades de vitória, empate e derrota.

Através do cálculo dos gols esperados e da análise das probabilidades, foi introduzido o conceito de pressão, que permite avaliar a importância de um momento específico durante o jogo.

Antes de proceder com a implementação do modelo, foram realizados uma série de testes para verificar a validade das funções e a precisão das previsões geradas. Os testes foram executados utilizando diferentes conjuntos de dados e cenários, permitindo assim validar o funcionamento adequado do modelo.

Verificou-se se as probabilidades geradas estavam dentro do esperado, considerando o contexto das partidas e as características das equipes envolvidas. Os resultados obtidos foram comparados com os resultados reais das partidas para avaliar a precisão das previsões, de modo que fosse possível prosseguir com o cálculo da métrica de pressão durante o jogo.

### 5.1.3. Pressão Psicológica Final

Descreve-se aqui o desenvolvimento do código para calcular a coluna de “pressão” no *dataframe* “eventsSCP”. O objetivo é calcular a pressão final para cada linha do *dataframe*, levando em consideração as etapas e funções desenvolvidas anteriormente. Inicialmente, definiu-se a equipe alvo como “Sporting” e prosseguiu-se para iterar sobre cada linha do *dataframe* “eventsSCP”.

Para cada linha, primeiro extraiu-se a data da partida e a equipe adversária utilizando a função “get\_match\_date(match\_id, LigaPTResults)” e “find\_opposition(team, date, LigaPTResults)”, respectivamente. Em seguida, é calculada a tabela de classificação da liga naquela data usando a função “calculate\_league\_table(date, LigaPTResults)”. A tabela de classificação é ordenada com base nos critérios de classificação e uma nova coluna “Position” é adicionada para indicar a posição atual da equipe.

A seguir, demonstra-se a adaptação necessária para a automatização deste cálculo. São extraídos os valores de gols e minutos em que foram marcados sob forma de listas, usando a função “extract\_goals\_minutes(match\_id, goalstime)”. Essas informações são necessárias para calcular a diferença de gols e as probabilidades de vitória, empate e derrota.

A diferença de gols é calculada usando a função “calculate\_goal\_difference(team1\_goals, team1\_minutes, team2\_goals, team2\_minutes, specific\_minute)”. Essa métrica considera os gols marcados pela equipe alvo e pelo adversário até o minuto específico de cada linha do *dataframe*.

Finalmente, a pressão final da ação foi calculada usando a função “live\_and\_pre\_game\_pressure(normalized\_metric, pressure)”. Essa função combina a métrica normalizada e a pressão específica da partida para obter a pressão final.

Após o cálculo da pressão final, a coluna “pressure” no *dataframe* “eventsSCP” foi atualizada com o valor calculado para cada linha. Ao finalizar o processo de iteração sobre todas as linhas, tem-se o *dataframe* “eventsSCP” com a nova coluna de “pressure”, fornecendo uma medida da pressão para cada evento de partida.

Ademais, para que essa automatização fosse possível, foi necessário mecanizar a extração dos minutos e gols de cada partida, com o código observado a seguir:

```
def process_events(eventsSCP):  
  
    goalstime = pd.DataFrame(columns=['matchId', 'team1_goals',  
    'team1_minutes', 'team2_goals', 'team2_minutes'])  
  
    for matchId in eventsSCP['matchId'].unique():  
  
        match_events = eventsSCP[eventsSCP['matchId'] == matchId]
```

```

team1_goals = []

team1_minutes = []

team2_goals = []

team2_minutes = []

for index, row in match_events.iterrows():

    if row['shot.isGoal'] == 'True':

        team1_goals.append(1)

        team1_minutes.append(row['minute'])

        elif 'conceded_goal' in row['type.secondary'] or
'penalty_conceded_goal' in row['type.secondary']:

            team2_goals.append(1)

            team2_minutes.append(row['minute'])

goalstime = pd.concat([goalstime, pd.DataFrame({

    'matchId': [matchId],

    'team1_goals': [team1_goals],

    'team1_minutes': [team1_minutes],

    'team2_goals': [team2_goals],

    'team2_minutes': [team2_minutes]

})], ignore_index=True)

return goalstime

eventsSCP = pd.read_excel('/content/eventsSCP.xlsx')

goalstime = process_events(eventsSCP)

goalstime

```

Esse código fornece uma implementação eficiente para calcular a pressão ao vivo e pré-jogo com base em várias métricas e informações relevantes. Os resultados obtidos podem auxiliar na análise e compreensão do impacto psicológico e emocional nas partidas de futebol. Além disso, a metodologia utilizada pode ser adaptada e aplicada a outros conjuntos de dados e contextos relacionados ao esporte. A implementação pode ser observada a seguir:

```
import pandas as pd

from tqdm import tqdm

team = "Sporting" # Doesn't change

total_rows = len(filtered_events)

# Iterate over each row in the events dataframe

for index, row in tqdm(filtered_events.iterrows(), total=total_rows,
desc="Calculating Pressure"):

    Confronto_Direto = pd.DataFrame({

        'Team': [team, opposition],

        'Games': [0, 0],

        'Win': [0, 0],

        'Draw': [0, 0],

        'Lost': [0, 0],

        'Points': [0, 0]

    })

    # Get the match date and opposition

    date = get_match_date(row['matchId'], LigaPTResults)

    opposition = find_opposition(team, date, LigaPTResults)
```

```

date = pd.to_datetime(date, format='%d/%m/%Y')

# Calculate the day before the given date

previous_day = date - timedelta(days=1)

date = get_match_date(row['matchId'], LigaPTResults)

# Calculate the league table

league_tablebefore = calculate_league_table(previous_day,
LigaPTResults)

league_tableafter = calculate_league_table(date, LigaPTResults)

# Sort the dataframe by the specified columns in descending order

league_tablebefore = league_tablebefore.sort_values(by=["Points",
"Win", "Goal Difference", "Goals Scored"], ascending=False)

league_tableafter = league_tableafter.sort_values(by=["Points",
"Win", "Goal Difference", "Goals Scored"], ascending=False)

# Reset the index of the dataframe

league_tablebefore = league_tablebefore.reset_index(drop=True)

league_tableafter = league_tableafter.reset_index(drop=True)

# Add the position column

```

```

league_tablebefore["Position"] = league_tablebefore.index + 1

league_tableafter["Position"] = league_tableafter.index + 1


if len(league_tablebefore) < 18:

    league_tablebefore = league_tableafter.copy()


# Calculate the importance value

    importance_value = calculate_importance_value(team, date,
league_tablebefore, league_tableafter, LigaPTResults, opposition)


# Set the scoreline value (assuming it doesn't change)

scoreline_value = 0


# Calculate the opposition quality value

    opposition_quality_value = opposition_quality(spi_portugal, team,
opposition)


# Calculate the crowd support value

    crowd_support_value = calculate_crowd_support(opposition,
LigaPTResults, date, crowd_support_df)


# Calculate the away game value

    away_game_value = check_team_home_or_away(team, date,
LigaPTResults)

```



```

# Calculate the team form value

team_form = get_team_form(team, date, LigaPTRResults)

opposition_form = get_team_form(opposition, date, LigaPTRResults)

team_form_value = calculate_form_value(team_form, opposition_form)


# Calculate the recent encounters value if available

recent_encounters_value = calculate_recent_encounters(team,
opposition, LigaPTRResults, Confronto_Direto, date)


# Calculate the mental pressure metric

metric = calculate_mental_pressure_metric(importance_value,
scoreline_value, opposition_quality_value,

crowd_support_value,
away_game_value,

team_form_value,
recent_encounters_value)


# Normalize the metric

normalized_metric = normalize_metric(metric)


# Find the matchId for the specific row

match_id = row['matchId']


# Extract goals and minutes for the matchId

```

```

        team1_goals, team1_minutes, team2_goals, team2_minutes =
extract_goals_minutes(match_id, goalstime)

# Set the max goals for each team (assuming it doesn't change)

max_goals1 = 5

max_goals2 = 5

# Extract team data (elo, off, def)

elo1, off1, def1 = extract_team_data(team)

elo2, off2, def2 = extract_team_data(opposition)

# Calculate the goal difference using the specific minute from the
row

specific_minute = row['minute']

goal_diff = calculate_goal_difference(team1_goals, team1_minutes,
team2_goals, team2_minutes, specific_minute)

# Calculate the expected goals

expected_goals1, expected_goals2 = calculate_expected_goals(elo1,
elo2, off1, off2, def1, def2)

# Calculate the win/draw/loss probabilities

win_prob, draw_prob, loss_prob =
calculate_win_draw_loss_probabilities(elo1, elo2, off1, off2, def1,
def2,

```

```

max_goals1, max_goals2, goal_diff)

    # Calculate the pressure using the specific minute and goal
difference

    pressure = pressure_minute(specific_minute, goal_diff)

    # Calculate the final pressure

                                final_pressure      =
round(live_and_pre_game_pressure(normalized_metric, pressure), 4)

    # Update the 'pressure' column in the events dataframe with the
calculated value

    filtered_events.at[index, 'pressure'] = final_pressure

# Print the updated dataframe

filtered_events

```

## 5. 2. Go-live: fase de implementação

No intuito de ilustrar tudo que foi desenvolvido neste projeto, propõe-se uma aplicação do modelo, para que seja possível observá-la no ambiente do futebol. Para isso, foi descrito o processo de teste e implementação do estudo sobre o desempenho dos jogadores sob diferentes níveis de pressão, mais particularmente, como o desempenho dos jogadores variou em relação aos cinco níveis de pressão: “Very Low Pressure”, “Low Pressure”, “Normal Pressure”, “High Pressure” e “Very High Pressure”. Esses níveis de pressão foram designados de acordo com os valores dos quintis de pressão psicológica dos jogadores no momento de suas finalizações. Com isso, será possível identificar jogadores que se

destacaram positivamente e negativamente em índices elevados de pressão. Também serão apresentados gráficos que ilustram o desempenho dos jogadores com as métricas Expected Goals (xG) e Post Shot Expected Goals (PSxG), sob a ótica de pressão desenvolvida.

Os dados utilizados neste estudo foram obtidos a partir do *dataframe* “filtered\_events”, que contém informações detalhadas sobre as finalizações e eventos ocorridos durante as partidas. Esses eventos foram filtrados com base nos critérios definidos anteriormente, resultando em uma amostra representativa para análise. Ademais, foi adicionada uma coluna com os valores obtidos de pressão psicológica para cada evento, além de sua respectiva classificação de nível de pressão. Nota-se que por motivos de privacidade do Sporting CP, os nomes dos jogadores permanecem sigilosos, no desenvolvimento deste projeto.

A fim de analisar o desempenho dos jogadores sob diferentes níveis de pressão, foram calculadas as métricas de Expected Goals (xG) e Post Shot Expected Goals (PSxG) para cada jogador e para cada um dos cinco níveis de pressão. Essas métricas fornecem uma medida quantitativa do desempenho dos jogadores em termos de qualidade das chances criadas e efetividade após o chute.

A seguir, observa-se sob quais circunstâncias cada jogador realizou suas finalizações:

player.id	1. Very Low Pressure	2. Low Pressure	3. Normal Pressure	4. High Pressure	5. Very High Pressure
3296	0	0	0	0	1
7954	1	1	2	2	5
70960	4	3	2	2	3
72331	7	9	7	5	6
167145	0	1	4	0	0
279362	4	1	4	2	3
279365	13	8	11	11	8
297369	0	2	4	1	2
353237	11	12	10	12	7
359836	1	3	0	1	1
403453	6	0	3	4	3
431540	12	13	13	11	12
446705	3	3	4	2	6
496659	1	0	0	0	0
521814	16	28	16	23	11
551442	1	4	7	6	10
559904	4	7	4	6	6
589103	0	1	0	0	1
642792	0	0	2	0	0
645032	9	2	4	8	6
645035	0	0	0	1	1
645045	0	1	0	0	1
647122	5	1	3	2	5
658308	0	0	1	0	0
718833	2	0	0	0	0
729649	1	0	0	0	2
807210	2	2	1	3	3

**Quadro 1: Número de finalizações de cada jogador sob diferentes níveis de pressão**

**Fonte: Autoria Própria**

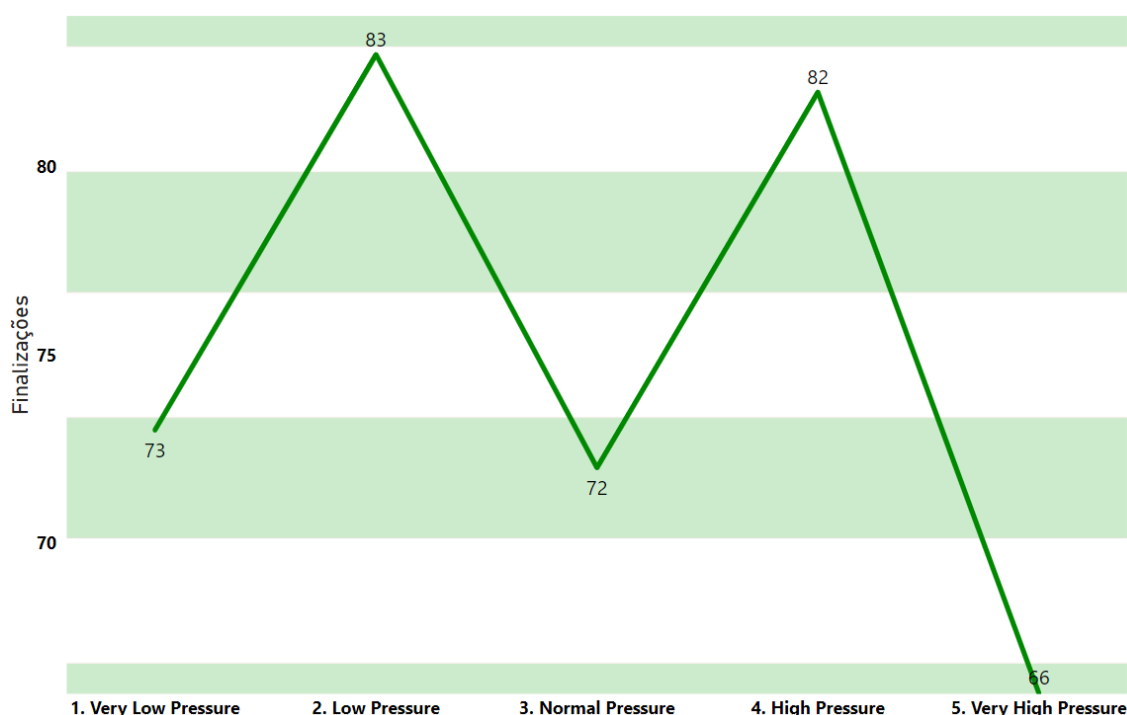
Para a elaboração da análise dessas finalizações, optou-se por selecionar apenas os jogadores com mais de 20 finalizações para que o tamanho amostral não deturpasse significativamente os resultados obtidos.

player.id	1. Very Low Pressure	2. Low Pressure	3. Normal Pressure	4. High Pressure	5. Very High Pressure
72331	7	9	7	5	6
279365	13	8	11	11	8
353237	11	12	10	12	7
431540	12	13	13	11	12
521814	16	28	16	23	11
551442	1	4	7	6	10
559904	4	7	4	6	6
645032	9	2	4	8	6

**Quadro 2: Número de finalizações (mais que 20) de cada jogador sob diferentes níveis de pressão**

**Fonte: Autoria Própria**

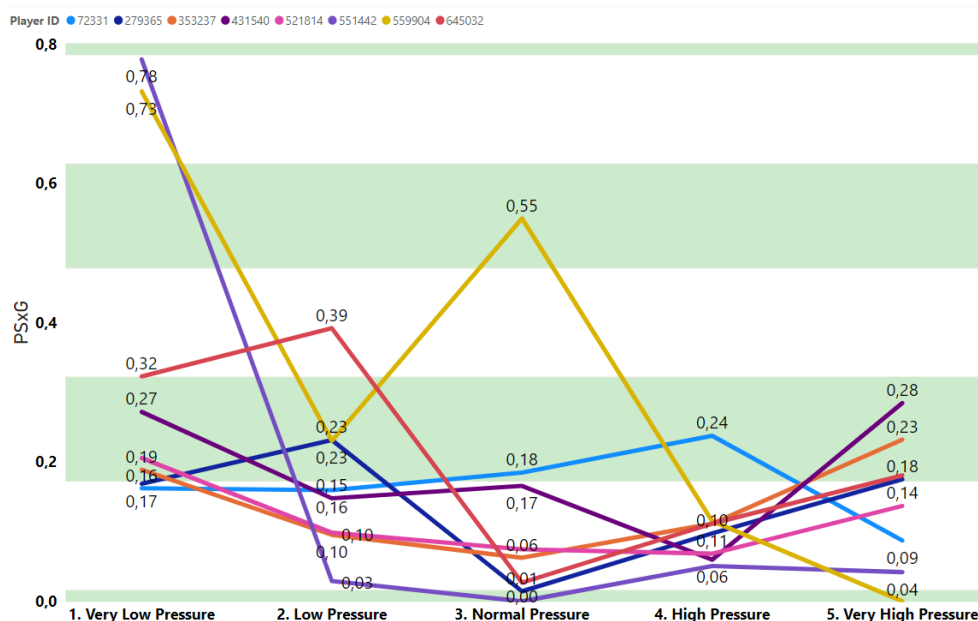
Num primeiro momento, ao analisar a quantidade de finalizações realizadas sob diferentes níveis de pressão, observa-se uma queda significativa na quantidade do último quintil. Portanto, nota-se que os jogadores têm tendência a diminuir o volume de finalizações em momentos críticos psicologicamente.



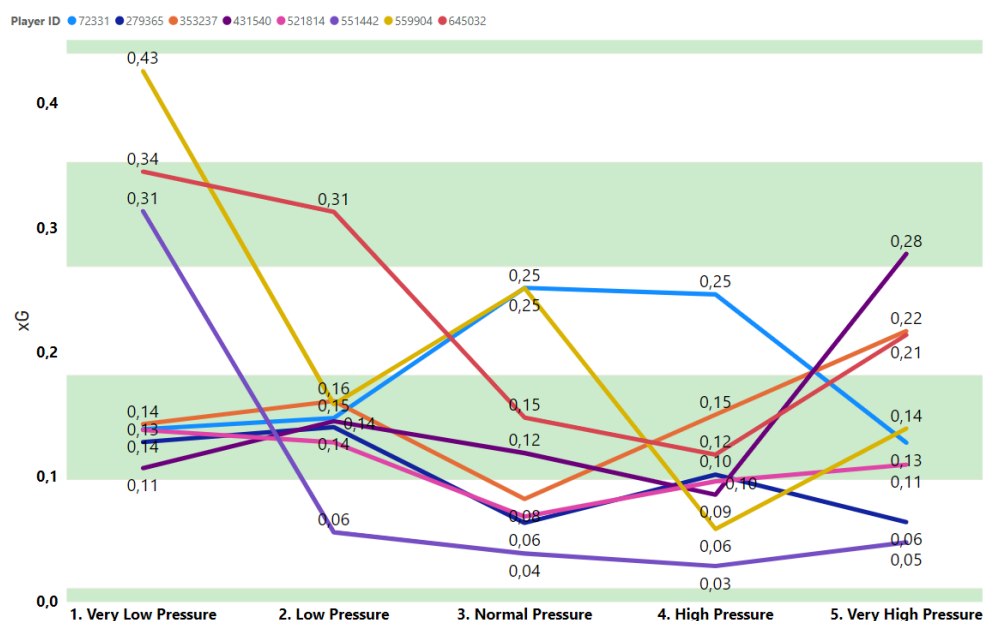
**Gráfico 1: Número de finalizações sob diferentes níveis de pressão**

**Fonte: Autoria Própria**

Além disso, observa-se variações significativas no desempenho dos jogadores em relação aos diferentes níveis de pressão. Alguns jogadores apresentaram um desempenho consistente em todos os níveis, enquanto outros tiveram um desempenho mais impactado pela pressão, tanto no xG, quanto no PSxG. Para facilitar a compreensão e a análise dos resultados, foram criados gráficos que ilustram o desempenho dos jogadores com as métricas de xG e PSxG em cada nível de pressão. Esses gráficos permitiram visualizar as diferenças no desempenho entre os jogadores em diferentes contextos de jogo.



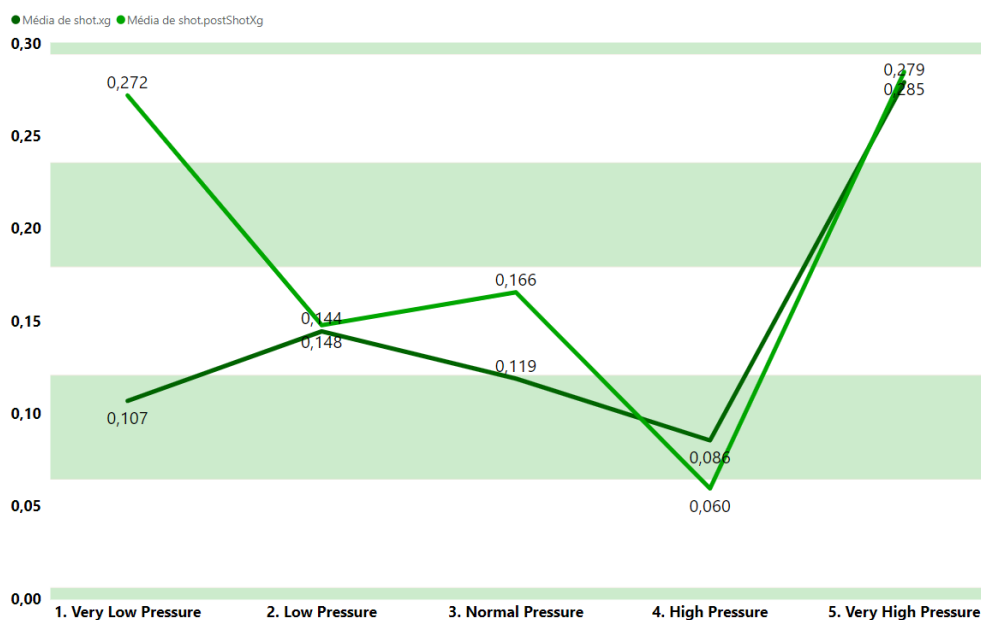
**Gráfico 2: Média de PSxG das finalizações (mais que 20) dos jogadores sob diferentes níveis de pressão**  
**Fonte: Autoria Própria**



**Gráfico 3: Média de xG das finalizações (mais que 20) dos jogadores sob diferentes níveis de pressão**  
**Fonte: Autoria Própria**

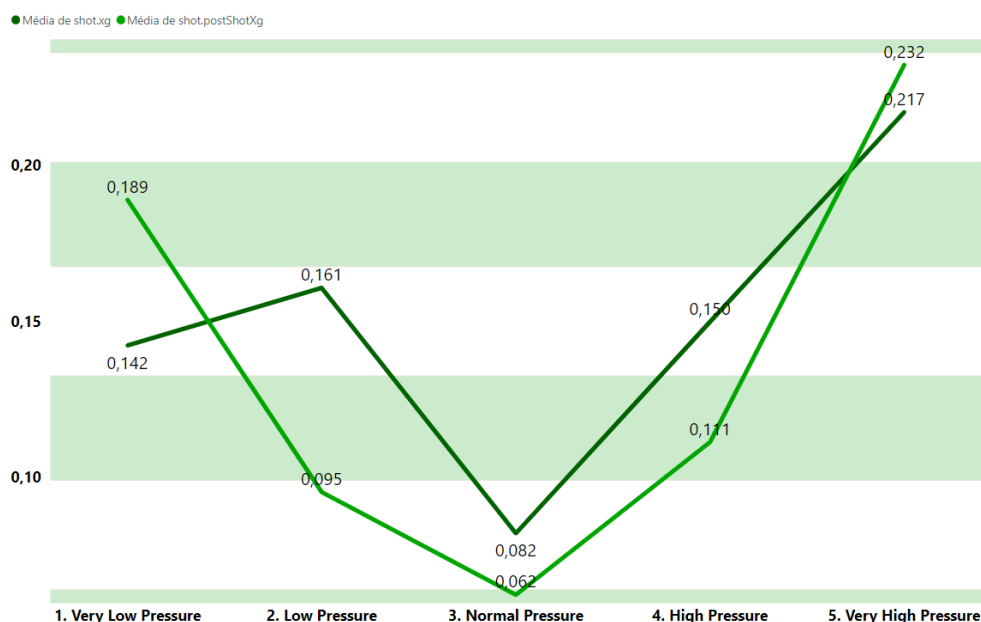
Em relação aos jogadores que se destacaram positivamente sob um índice alto de pressão, observamos que alguns atletas conseguiram manter um desempenho acima da média, mesmo sob condições mais desafiadoras. Esses jogadores demonstraram habilidades técnicas e capacidade de lidar com a pressão circunstancial. Suas métricas de xG e PSxG foram

consistentemente altas ou sensivelmente melhoradas, indicando um bom aproveitamento das oportunidades criadas e uma finalização eficiente. Mais especificamente, é possível citar os jogadores de índice 431540, 353237. Mesmo que esta não seja uma melhora significativa no seu desempenho em comparação aos demais atletas, eles se destacam.



**Gráfico 4: Variação da média de xG e PSxG do jogador 431540 sob diferentes níveis de pressão**

**Fonte: Autoria Própria**

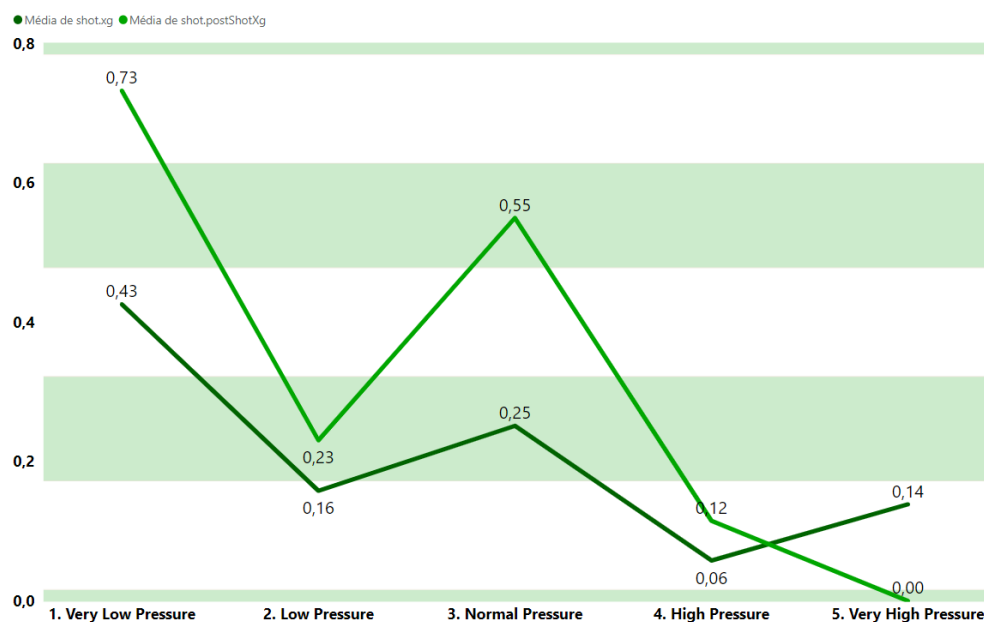


**Gráfico 5: Variação da média de xG e PSxG do jogador 353237 sob diferentes níveis de pressão**

**Fonte: Autoria Própria**

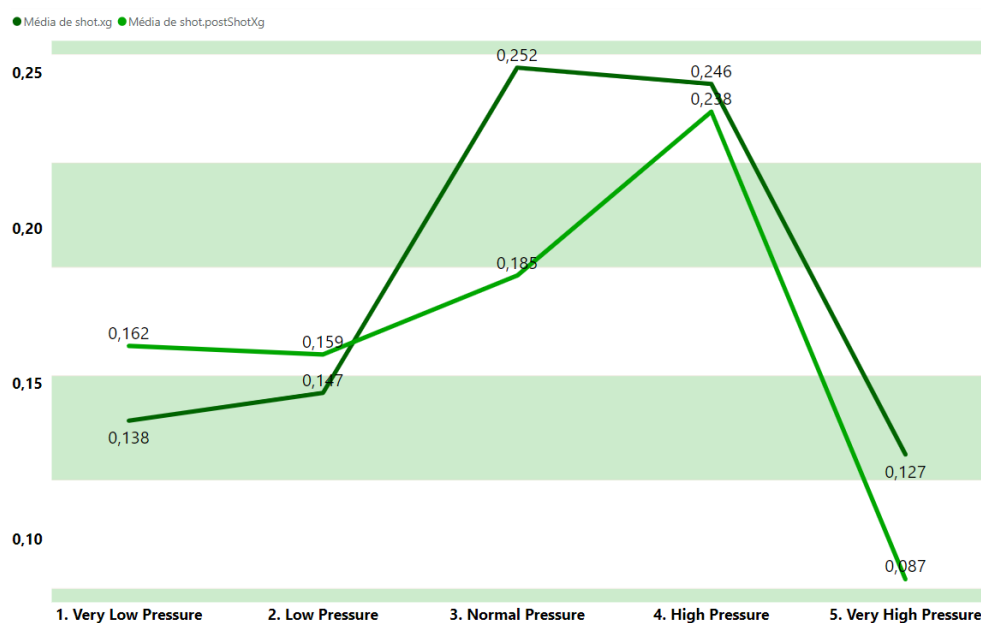


Por outro lado, foram identificados jogadores que apresentaram um desempenho negativamente afetado em índices altos de pressão. Esses jogadores mostraram dificuldades em lidar com a pressão adversária, resultando em um desempenho decrescente em termos de xG e PSxG, à medida que a pressão psicológica aumentava. Inclusive, observa-se essa queda de rendimento de maneira muito evidente dos jogadores de índice 559904, 72331 e 551442.



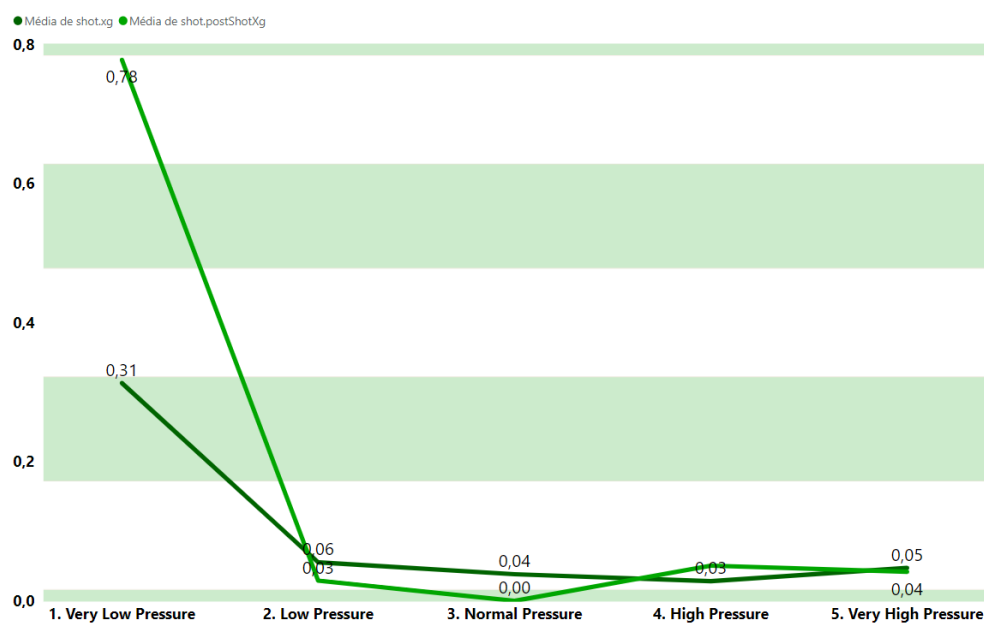
**Gráfico 6: Variação da média de xG e PSxG do jogador 559904 sob diferentes níveis de pressão**

**Fonte: Autoria Própria**



**Gráfico 7: Variação da média de xG e PSxG do jogador 72331 sob diferentes níveis de pressão**

**Fonte: Autoria Própria**

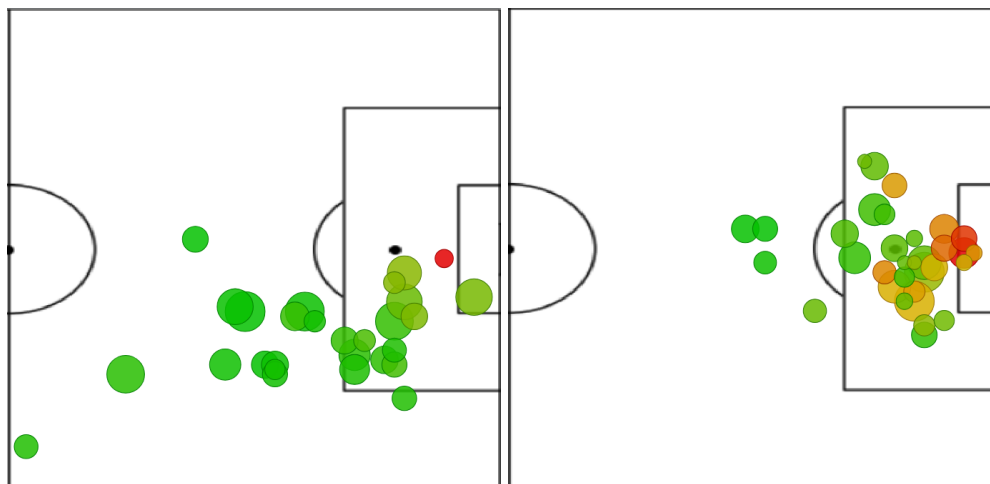


**Gráfico 8: Variação da média de xG e PSxG do jogador 551442 sob diferentes níveis de pressão**

**Fonte: Autoria Própria**

Uma dinâmica interessante é ilustrar esses cenários de finalizações, com o intuito de diagnosticar os pontos sensíveis do jogo do atleta em questão. Abaixo, a título de exemplo, encontra-se o mapa de finalizações dos jogadores 551442 e 72331. A partir disso, torna-se possível compreender se os seus baixos desempenhos foram resultado, por exemplo, de uma

tomada de decisão equivocada, posicionamento ou de um erro técnico. Nota-se que os tamanhos dos círculos aumentam de acordo com a pressão psicológica da ação. As cores representam o xG da finalização, no qual o verde denota um xG menor que o vermelho.

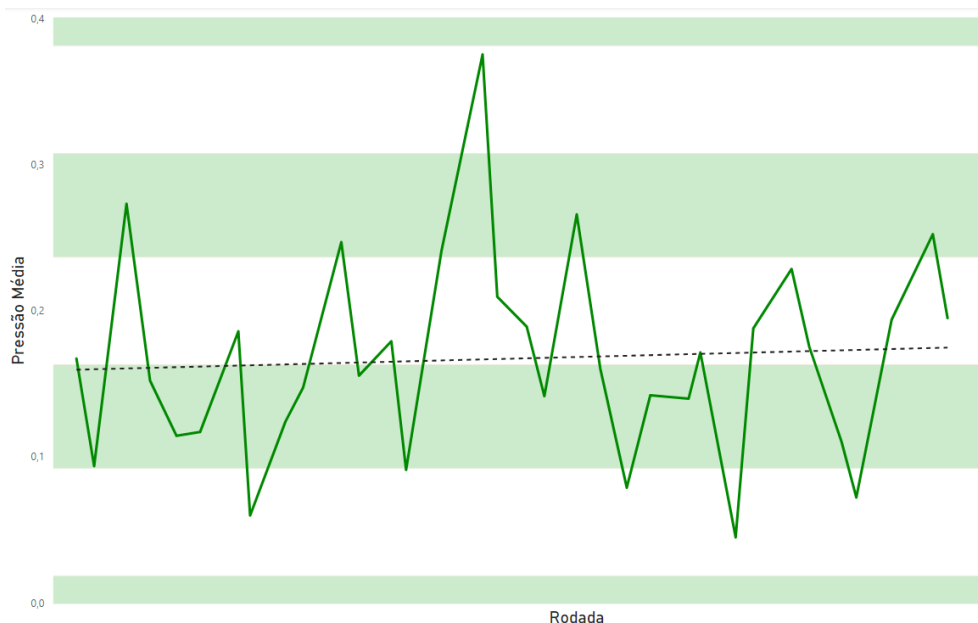


**Gráfico 9: Mapa de finalizações dos jogadores 551442 e 72331**

**Fonte: Autoria Própria**

### **5.3. Discussão dos resultados**

A coerência dos resultados de pressão psicológica pode ser avaliada pela análise da linearidade da variação da pressão média em cada partida ao longo do campeonato. Como os resultados apresentam uma tendência linear consistente, isso indica que a pressão psicológica é um fator determinante nas partidas, uma vez que a pressão média aumenta progressivamente ao longo do campeonato. Em suma, é possível inferir que os jogadores estão enfrentando uma maior pressão psicológica à medida que a competição se intensifica, como se observa a seguir.



**Gráfico 10: Variação da Pressão Média ao longo do campeonato**

**Fonte: Autoria Própria**

Os resultados obtidos fornecem esclarecimentos valiosos sobre a influência da pressão no desempenho dos jogadores. Essas informações podem ser utilizadas por treinadores, analistas e tomadores de decisão no futebol para entender melhor como os jogadores respondem em diferentes situações de pressão.

Esta análise, por exemplo, pode ser utilizada para orientar as estratégias da equipe. Com base nos dados obtidos, é possível identificar os jogadores que se destacam positivamente sob alta pressão, ou seja, aqueles que conseguem manter um bom desempenho, mesmo diante de adversidades. Esses jogadores podem ser designados como “clutch players”, como foi visto anteriormente, e em momentos decisivos, serem figuras centrais na elaboração de estratégias e táticas de jogo.

Por outro lado, a análise também pode revelar jogadores que apresentam dificuldades em situações de alta pressão, comprometendo o desempenho da equipe. Nesses casos, estratégias podem ser desenvolvidas para apoiar esses jogadores, como treinamentos específicos focados em reduzir a ansiedade e melhorar a confiança em momentos cruciais. Além disso, é possível criar jogadas táticas que facilitem a participação desses jogadores em momentos de menor pressão, proporcionando-lhes oportunidades para se destacarem e recuperarem a confiança.

Com base nos dados, a equipe técnica pode também identificar padrões de desempenho em diferentes momentos do jogo e ajustar as estratégias de acordo. Por exemplo, se a análise mostrar que a equipe tem um desempenho superior nas finalizações em momentos de pressão moderada, pode-se adotar uma abordagem tática que busque criar essas situações favoráveis nas circunstâncias apropriadas.

No geral, a análise do desempenho dos jogadores nas finalizações, em diferentes níveis de pressão, permite que a equipe técnica tome decisões embasadas para maximizar o desempenho da equipe como um todo. Essas estratégias podem envolver treinamentos específicos, apoio psicológico, ajustes táticos e a valorização dos jogadores que demonstram excelência sob pressão. Ao aplicar essas medidas, a equipe aumenta suas chances de sucesso nas finalizações e alcança um melhor desempenho global dentro de campo.

## **6. Conclusões e trabalhos futuros**

### **6.1. Conclusões**

Este trabalho teve o objetivo de explorar a análise de dados no futebol, com foco na avaliação do desempenho dos jogadores sob diferentes níveis de pressão psicológica. Ao longo do estudo, buscou-se compreender a influência da pressão nos resultados das partidas e nas performances individuais. Para isso, propôs-se um ponderador de contexto que considerasse tanto a pressão intrínseca quanto a extrínseca.

Ao revisar a literatura e examinar diferentes abordagens, foi possível estabelecer uma arquitetura conceitual para a métrica de desempenho dos jogadores de futebol sob pressão psicológica. Essa arquitetura consistiu em três componentes inter-relacionados: avaliação da pressão, avaliação do desempenho técnico e avaliação do desempenho psicológico.

A avaliação da pressão permitiu a identificação e a quantificação dos diferentes níveis de pressão enfrentados pelos jogadores durante uma partida. Dividindo a pressão em pré-partida e durante a partida, foi possível determinar os indicadores intrínsecos e extrínsecos que influenciam a pressão em cada momento.

A avaliação do desempenho psicológico comparou a performance dos jogadores nos diferentes contextos de pressão. A análise permitiu a transformação da informação em uma aplicação prática, fornecendo esclarecimentos valiosos para momentos de treino, análises, scouting, estratégias e táticas, fornecendo subsídios para orientar estratégias da equipe, identificando jogadores que se destacam positivamente sob alta pressão e aqueles que apresentam dificuldades nesses momentos.

Foi observado por meio da implementação do ponderador desenvolvido que as ações são efetivamente afetadas pela pressão psicológica enfrentada ao longo da partida e do campeonato. Essa constatação reforça a relevância de considerar esse fator ao avaliar o desempenho de atletas e equipes, em vez de analisar os eventos de maneira isolada, sem levar em conta o contexto em que ocorrem.

No entanto, é importante ressaltar que este estudo apresenta algumas limitações. A avaliação da pressão psicológica é um desafio complexo, e nem todos os fatores relevantes podem ser considerados neste modelo. Além disso, a generalização dos resultados para diferentes equipes, competições e contextos pode variar. Portanto, o aprofundamento na

compreensão dos aspectos psicológicos e o desenvolvimento de estratégias personalizadas para cada jogador podem contribuir para uma aplicação ainda mais eficaz do modelo proposto.

Em suma, este trabalho demonstrou a importância da análise de dados e da consideração da pressão psicológica no desempenho dos jogadores de futebol. Através da avaliação da pressão, do desempenho técnico e do desempenho psicológico foi possível avaliar o desempenho de atletas para apoiar a tomada de decisão e melhorar a performance individual e coletiva em momentos de pressão. Espera-se que essas descobertas contribuam para o avanço da análise de dados no futebol e auxiliem no desenvolvimento de estratégias mais eficazes para as equipes.

## **6.2. Trabalhos futuros**

Um aspecto importante a ser considerado é a inclusão dos cartões vermelhos na avaliação das probabilidades de resultado final de uma partida. Estudos anteriores, como o artigo “How Does a Red Card Affect the Outcome of a Football Match?” (IVANOV, 2018), demonstraram que a expulsão de um jogador pode ter um impacto significativo no desempenho da equipe e, conseqüentemente, nas probabilidades de resultado. Portanto, incorporar essa variável ao cálculo da pressão psicológica durante o jogo poderia fornecer uma visão mais completa e precisa das situações de alta pressão enfrentadas pelos jogadores.

No entanto, é importante ressaltar que os cartões vermelhos não foram considerados neste estudo devido à limitação dos dados disponíveis na base utilizada. As informações sobre cartões vermelhos não estavam disponíveis para análise, o que restringiu a inclusão dessa variável no modelo. Portanto, é recomendado que estudos futuros, fontes de dados mais abrangentes e detalhadas que forneçam informações completas sobre as ocorrências de cartões nas partidas sejam utilizadas.

Outro ponto a ser abordado é o cálculo da linearidade da pressão ao longo do jogo. Neste estudo, foi adotada uma minutagem padrão de tempo de partida de 96 minutos, uma vez que os dados disponíveis não forneciam os minutos totais de cada partida. Para trabalhos futuros, é sugerido que sejam utilizadas informações mais precisas sobre a duração real de cada partida, a fim de calcular a pressão em intervalos de tempo mais condizentes com a singularidade de cada partida.

Ademais, uma melhoria adicional para tornar o projeto mais completo seria a customização da variável "max\_goals" para cada equipe. No contexto da simulação de resultados com a distribuição de Poisson, o "max\_goals" foi definido como um valor genérico de 5 gols para cada equipe, limitando o número máximo de gols que cada equipe poderia marcar. Entretanto, para uma pesquisa mais aprofundada e precisa, recomenda-se adaptar esse número máximo de gols para cada equipe, levando em consideração fatores como o desempenho histórico, o estilo de jogo e a qualidade dos jogadores. Essa customização permitiria simulações mais realistas e coerentes, refletindo de forma mais precisa os resultados potenciais de cada equipe em um jogo.

Em suma, há várias oportunidades para pesquisas futuras no campo do desempenho dos jogadores sob diferentes níveis de pressão. A inclusão dos cartões vermelhos como uma variável significativa, a consideração da duração real das partidas e a customização da variável "max\_goals" são aspectos que podem aprimorar ainda mais a compreensão e a aplicação prática dos resultados obtidos.



## 7. Bibliografia

Analytics FC – HOBBS, Josh. Developing New Metrics: Who are the most “clutch” creators in football?. Analytics FC, 2021. Disponível em: <<https://analyticsfc.co.uk/blog/2021/10/14/developing-new-metrics-who-are-the-most-clutch-creators-in-football/>>.

Analytics FC – STEELE, Jeremy. Developing New Metrics: Who are the most “clutch” goalscorers in football?. Analytics FC, 2019. Disponível em: <<https://analyticsfc.co.uk/blog/2021/10/14/developing-new-metrics-who-are-the-most-clutch-creators-in-football/>>.

BRANSEN, Lotte et al. Choke or Shine? Quantifying Soccer Players' Abilities to Perform Under Mental Pressure. In: **Proceedings of the 13th MIT Sloan Sports Analytics Conference**. MIT SLOAN; <http://www.sloansportsconference.com/wp-content/uploads/2019/02/Choke-or-Shine-Quantifying-Soccer-Players-Abilities-to-Perform-Under-Mental-Pressure.pdf>, 2019. p. 1-25.

BRONFENBRENNER, Urie. Recent advances in research on the ecology of human development. **Development as action in context: Problem behavior and normal youth development**, p. 287-309, 1986.

DECROOS, Tom et al. Actions speak louder than goals: Valuing player actions in soccer. In: **Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining**. 2019. p. 1851-1861.

FIFA COMMUNICATIONS DIVISION. FIFA Big Count 2006: 270 million people active in football. FIFA Digital Hub, [s. l.], 31 maio 2007.

GARCÍA-ALIAGA, Abraham et al. Comparative analysis of soccer performance intensity of the pre–post-lockdown COVID-19 in LaLiga™. **International Journal of Environmental Research and Public Health**, v. 18, n. 7, p. 3685, 2021.

Inpredictable. Measuring Clutch Play in the NBA. inpredictable, 2014. Disponível em: <<https://www.inpredictable.com/2014/03/measuring-clutch-play-in-nba.html>>. Acesso em: 01 de junho de 2023.

LAGO-PEÑAS, Carlos; GOMEZ, Miguel-Ángel; POLLARD, Richard. Home advantage in elite soccer matches. A transient effect?. **International Journal of Performance Analysis in Sport**, v. 17, n. 1-2, p. 86-95, 2017.

SCARF, Philip A.; SHI, Xin. The importance of a match in a tournament. **Computers & Operations Research**, v. 35, n. 7, p. 2406-2418, 2008.

## 8. Anexos

Repositório Github: <https://github.com/martin-pare/PressureSportingCP>

