Soccer Analytics - Ciência de dados aplicada ao futebol

Gabriel Bortoli[1], **Lucas Maretti**[2], **and Roseli A.P. Romero**[3]

[1]**Universidade de São Paulo - Instituto de Ciencias Matemáticas e Computação**

ABSTRACT

The accurate prediction of expected goals (xG) in soccer matches has gained significant attention in recent years due to its potential applications in various aspects of the game. In this study, we investigate the performance of different classification models, namely Logistic Regression, Random Forest, XGBoost, LightGBM, and LSTM; for developing an xG model using the open data provided by the renowned company Wyscout.

The dataset used in this research contains comprehensive information about various match events, player actions, and contextual features. We preprocess the data and engineer relevant features to enhance the model's predictive capabilities. We then compare the performance of the aforementioned models by evaluating their log-loss scores.

Our experiments reveal that the XGBoost model outperforms other classification models, achieving a log-loss of 0.2793. The XGBoost model demonstrates superior predictive accuracy in estimating the likelihood of a goal-scoring opportunity based on the available match data. The robustness and flexibility of the XGBoost algorithm make it an excellent choice for modeling expected goals in soccer matches.

These findings highlight the potential of machine learning techniques in soccer analytics and provide valuable insights for both researchers and practitioners in the field. The developed xG model can be utilized in various applications, such as player evaluation, team performance analysis, and strategic decision-making during matches.

Keywords: football analytics, probabilistic classification, soccer match data

1 INTRODUÇÃO

De acordo com a Nielsen, líder em análises de audiência, o futebol é o esporte com maior número de fãs no mundo. Em seu *Annual World Report* de 2022 (Nielsen (2022)), que conta com análises de dados sobre a audiência do esporte, estima-se que 40% da população mundial tenha o futebol como esporte predileto, o que o torna um dos mais relevantes em termos econômicos. Em matéria publicada também em 2022, a revista Forbes (Forbes (2022)) listou o clube espanhol Real Madrid como o mais valioso do mundo, com um valor de mercado de 5.1 bilhões de euros, seguido de perto por clubes como Barcelona, com 5.1 bi e Liverpool, com 4.45 bi. No cenário brasileiro, a situação é análoga: Flamengo e Palmeiras lideram a avaliação de mercado de acordo com estudo feito pela consultoria Sport Value (Exame (2022)) com 3.4 e 3.1 bilhões de reais de valor, respectivamente. Além disso, no Brasil o futebol trascende apenas ao entretenimento, tendo também um aspecto socio-econômico relevante: muitos garotos e garotas de baixa renda encontram no esporte uma oportunidade de ascender socialmente. Além disso, o sucesso da realização da Copa do Mundo de 2014 e das Olimpíadas de 2016 mostra que a união entre turismo e esporte é um vetor importante de impulsionamento econômico do país.

O *boom* da ciência de dados nos últimos 10 anos, que revolucionou setores da economia como *e-commerce*, varejo e manufatura, impulsionada pelo aumento do poder computacional disponível e da amplitude da coleta de dados através de dispositivos IoT chegou também ao esporte. Atualmente, dispositivos de sensoriamento vestidos por jogadores coletam dados da movimentação destes ao longo de partidas (também conhecidos como dados de *tracking*), sendo possível mapear a uma granularidade de segundo quais são as coordenadas do jogador com referência ao campo de futebol. Além disso, empresas especializadas como Wyscout, Statsbomb e Opta realizam o chamado *tagging* de eventos durante partidas, registrando detalhadamente as ações que acontecem no campo como passes, chutes, divididas, roubadas de bola, entre outros, gerando os chamados dados de *eventos*.

A união de dados de tracking e de eventos permitiu uma verdadeira revolução na maneira de se enxergar o futebol nos últimos cinco anos pelas lentes da ciência de dados. São poucos os clubes europeus que hoje não contam com um departamento de Analytics em suas estruturas organizacionais. No Brasil, este é um cenário ainda incipiente. Dos grandes clubes da Série A do campeonato brasileiro, apenas Atlético-MG e Red Bull Bragantino contam como uma estrutura dedicada a análise de dados, sendo que a do time mineiro foi a pioneira, sendo iniciada apenas em 2021, de acordo com o jornal O Tempo (Tempo (2021)). Sendo assim, este artigo tem como objetivo adicional contribuir para a literatura sobre o tema nesta área em âmbito nacional.

Desde que a ciência de dados passou a ser utilizada para analisar partidas de futebol, novas métricas e conceitos surgiram para tentar explicar a natureza do jogo, que em sua essência é aleatória e sujeita a incertezas. A principal dessas métricas é o chamado *expected goals* (esperança ou expectativa de gols, em português), conhecido também pela sigla xG. A expectativa de gols atribui uma probabilidade entre 0 e 1 a cada finalização feita por um jogador em um jogo (0 indicando nenhuma possibilidade de a finalização ser um gol e 1 indicando certeza de gol). Esta é uma maneira melhor de lidar com a aleatoriedade no futebol do que, por exemplo, uma métrica tradicional baseada em número de gols, já que um chute é um evento muito mais comum do que um gol. Nas palavras de David Sumpter, famoso matemático e autor do livro *Soccermatics*: "xG é a probabilidade de que em um dia típico de futebol um chute particular de uma determinada localização resultaria em um gol. Costuma ser baseado em medidas tomadas de muitos chutes dentro de uma mesma liga e temporada, ou agregando-se dados de diferentes ligas." (Sumpter (2017))

No livro *Football Hackers*, o autor Christoph Biermann conta a história de Ted Knutson, um dos pioneiros da utilização do expected goals, inicialmente como analista de dados para um empresa especializada em apostas esportivas e depois como fundador da empresa Statsbomb, hoje uma das líderes de mercado no fornecimento de soluções baseadas em coleta e análise de dados para clubes de futebol. Knutson atuou também como consultor esportivo para a equipe Brentford da Inglaterra.

O Brentford é frenquentemente descrito como um clube *Moneyball*, referência ao filme de 2011 estrelado por Brad Pitt e Jonah Hill que descreve o time de beisebol Oakland Athletics, que tornou-se pioneiro ao utilizar análise de dados para montagem de seu elenco, uma estratégia inovadora à época para superar as limitações de receita do clube (Biermann (2019)).

O uso de analytics na tomada de decisão marcou uma nova era para o pequeno time de Londres. Na temporada 2013-14, o clube disputava a terceira divisão da liga inglesa (League One) e desde 1947 não jogavam a primeira divisão, bem como nunca ganhara nenhuma taça de Copa da Inglaterra ou de liga. Atualmente o clube está em sua segunda temporada seguida na Premier League, a primeira divisão do futebol inglês, na nona posição do campeonato tendo conseguido resultados expressivos na temporada atual como vitórias sobre Liverpool, Chelsea e Manchester United na atual temporada 2023-2024.

Os dados necessários para construir qualquer modelo de xG no futebol (eventos e/ou tracking com informações posicionais) são difíceis de obter, pois as empresas que coletam os dados costumam usá-los para construir seus próprios modelos. Felizmente, como parte da iniciativa Soccer Data Challenge (um evento de análise de dados aplicada ao futebol realizado na Itália em 2019), os organizadores forneceram o que acreditam ser a maior coleção de dados de eventos já divulgada ao público (Pappalardo et al. (2019)). Os dados, que foram coletados pela empresa Wyscout, compreendem todos os eventos das 5 principais ligas europeias (Premier League Inglesa, La Liga Espanhola, Bundesliga Alemã, Serie A Italiana e Ligue 1 Francesa) para a temporada 2017–18.

De posse desses dados da Wyscout o objetivo deste estudo foi treinar diversos classificadores, a saber, Regressão Logística, Random Forest, LightGBM, XGBoost e LSTM nos dados das 5 ligas para todas as partidas disponíveis para a métrica de expected goals (xG). Para avaliar os modelos utilizou-se duas métricas principais: a perda logarítmica (log loss) e AUC (area under the curve).

Nas próximas seções serão apresentados em maiores detalhes a organização dos dados fornecidos pela Wyscout, bem quais modelos e métricas foram escolhidos para a modelagem como os dados foram tratados. Em seguida, tem-se uma descrição da análise exploratória dos dados e levantamento de hipóteses. Por fim o resultado final da modelagem após treinamento e comparação dos resultados entre os modelos e conclusão.

2 TRABALHOS RELACIONADOS

O artigo de Pappalardo et al. (2019) é fundamental de ser mencionado pois foi o artigo que introduziu a base de dados que hoje é a mais utilizada para pesquisas em *football analytics*. Neste artigo a base de dados é descrita em detalhes, dando exemplos também de como uma análise exploratória bem feita já pode trazer *insights* relevantes para os clubes. Já Mead et al. (2023) e Umami et al. (2021) trabalharam especificamente na modelagem de *expected goals*. Os primeiros testaram modelos como Regressão Logística, Random Forest, Multilayer Perceptron, Adaboost e XGBoost utilizando como métricas principalmente a perda logarítmica e AUC. A tabela a seguir mostra um compilado dos resultados obtidos pelos autores:

Figure 1. Resultados de log-loss para os modelos testados

League	Before Tuning					After Tuning				
	LR	MLP	RF	AB	XGB	LR	MLP	RF	AB	XGB
Premier Leaguex	0.28554	**0.28315**	0.36957	0.66474	0.38324	0.28364	0.28337	0.30365	0.31471	0.29268
La Liga	**0.30629**	0.31796	0.34123	0.66277	0.41489	0.32109	0.31975	0.31128	0.32538	0.31397
Bundesliga	0.29629	0.28814	0.33268	0.66909	0.34123	0.28685	0.2883	0.29733	0.31481	**0.28425**
Serie A	0.28907	0.2841	0.29934	0.67201	0.32746	0.28945	0.28922	0.29233	0.30801	**0.28295**
Ligue 1	**0.29118**	0.29171	0.34387	0.66371	0.36942	0.29366	0.29873	0.30114	0.32408	0.29752
All Leagues	0.28614	0.285	0.30698	0.671	0.30594	0.28563	0.28286	0.2897	0.31368	**0.28184**

Fonte: Mead et al. (2023)

Neste mesmo artigo, os autores trazem um comparativo com resultados obtidos de outras publicações sobre o mesmo tema. Os resultados estão resumidos e são apresentados através da tabela 2 a seguir.

Figure 2. Resultados de log-loss e AUC comparando a outras publicações sobre o mesmo tema

Model	Brier score	AUC ROC	Log-loss value
This model	0.0799	0.8	0.28184
Noordman [5]	0.0799		0.2787
Eggels [13]		0.823	
Anzer and Bauer [2]		0.814	

Fonte: Mead et al. (2023)

Já Umami et al. (2021) focaram na utilização de Regressão Logística com a métrica AUC, obtendo resultados similares aos citados anteriormente.

3 MATERIAIS E MÉTODOS

Conforme descrito na seção introdutória os dados a serem usados foram fornecidos pela empresa Wyscout como parte da iniciativa Soccer Data Challenge em 2019, que contempla todos os eventos das 5 principais ligas europeias (Premier League Inglesa, La Liga Espanhola, Bundesliga Alemã, Serie A Italiana e Ligue 1 Francesa) para a temporada 2017–18. Esse é um *dataset* interessante de se utilizar pois se tornou referência em estudos acadêmicos sobre o tema.

Apesar do fato de não haver dados posicionais (portanto, algumas *features* influentes examinados em outros trabalhos não podem ser incluídos nos modelos), esse conjunto de dados foi a fonte mais completa e disponível ao público que foi encontrada e contém as informações necessárias para cumprir os objetivos deste estudo.

3.1 Sobre o dataset

A Wyscout disponibilizou os dados em formato *.json* separados em 3 categorias: *matches*, com metadados das partidas, como localização, a qual liga se refere, nome do estádio, resultado da partida, entre outros; *events*, com os eventos (ações que ocorrem na bola, conforme descrito anteriormente) de cada uma das partidas (arquivo mais pesado de todos) e *players*, com metadados sobre os jogadores, como nome completo, idade, pé favorito para chutar, entre outros.

A seguir descreveremos as features presentes no arquivo de eventos, que é o principal para a modelagem (os demais podem ser encontrados no Anexo):

Dados de eventos (events.json):

- eventId: o identificador do tipo de evento. Cada eventId está associado a um nome de evento (consulte o próximo ponto);
- eventName: nome do tipo de evento. São sete tipos de eventos: passe, falta, chute, duelo, falta, impedimento e toque;
- subEventId: o identificador do tipo do subevento. Cada subEventId está associado a um nome de subevento (consulte o próximo ponto);
- subEventName: o nome do tipo do subevento. Cada tipo de evento está associado a um conjunto diferente de tipos de subeventos;
- tags: uma lista de tags de evento, cada uma descreve informações adicionais sobre o evento (por exemplo, um passe como evento, se esse passe foi preciso). Cada tipo de evento está associado a um conjunto diferente de tags;
- eventSec: o tempo em que o evento ocorre (em segundos desde o início do atual tempo da partida);
- id: um identificador único do evento;
- matchId: o identificador do match ao qual o evento se refere. O identificador refere-se ao campo "wyId" no conjunto de dados de partidas;
- matchPeriod: o período da partida. Pode ser "1H" (primeiro tempo da partida), "2H" (segundo tempo da partida), "E1" (primeiro tempo extra), "E2" (segundo tempo extra) ou "P" (pênaltis) ;
- playerId: o identificador do jogador que gerou o evento. O identificador refere-se ao campo "wyId" em um conjunto de dados do jogador;
- positions: as posições de origem e destino associadas ao evento. Cada posição é um par de coordenadas (x, y). As coordenadas x e y estão sempre no intervalo [0, 100] e indicam a porcentagem do campo na perspectiva do time atacante. Em particular, o valor da coordenada x indica a proximidade do evento (em porcentagem) ao gol adversário, enquanto o valor da coordenada y indica a proximidade do evento (em porcentagem) ao lado direito do campo;
- teamId: o identificador do time do jogador. O identificador refere-se ao campo "wyId" no conjunto de dados da equipe.

3.2 Modelos testados

A obtenção de um modelo de xG é um problema de machine learning do tipo classificação, dado que queremos obter a probabilidade de um chute resultar em um gol. Com isso em mente, cinco algoritmos foram testados neste trabalho, a saber: Regressão Logística, Random Forest Classifier, LightGBM, XGBoost e LSTM. A implementação destes modelos se deu através da biblioteca scikit-learn.

Para a Regressão Logística, os principais hiperparâmetros avaliados foram:

- C: Inverso da força de regularização; deve ser um número positivo. Valores menores especificam uma regularização mais forte. Valores testados: [0.3,0.5,0.8,1.0]
- Tipo de solver: algoritmo a utilizar no problema de otimização. Testaram-se os solvers 'lbfgs', que usa regularização L2 e 'newton-cholesky', recomendado quando se tem várias features com *one hot encoding*, como é o caso deste projeto.

Para a Random Forest, os hiperparâmetros estudados foram:

- Número de estimadores: Equivalente ao número de árvores a serem treinadas. Valores testados foram [30,40,60], dado que valores maiores tendiam a gerar overfit.
- Max depth: A profundidade máxima da árvore. Se None, os nós serão expandidos até que todas as folhas sejam puras. Valores testados: [4,5,6,7,8,9,10]
- Min sample split: O número mínimo de amostras necessárias para dividir um nó interno. Valores testados: [2,4,6,8]

Para o modelo LightGBM, os hiperparâmetros estudados foram:

- Número de estimadores: Equivalente ao número de árvores a serem treinadas. Valores testados foram [20, 50, 100, 200, 300]
- Max depth: A profundidade máxima da árvore. Se None, os nós serão expandidos até que todas as folhas sejam puras. Valores testados: [4,5,6,7,8,9,10]
- Número de folhas: representa o número máximo de folhas (leaf nodes) em uma árvore de decisão. Em outras palavras, determina a complexidade ou o tamanho máximo da árvore. O valor desse hiperparâmetro afeta diretamente a capacidade do modelo de se ajustar aos dados de treinamento. Ao aumentar o valor de "num leaves", a árvore se torna mais profunda e complexa, permitindo que o modelo se ajuste melhor aos dados de treinamento. Valores testados: [14,31,63,127]

Para o modelo XGBoost, os hiperparâmetros estudados foram:

- Número de estimadores: Equivalente ao número de árvores a serem treinadas. Valores testados foram [20, 50, 100, 200, 300].
- Max depth: A profundidade máxima da árvore. Se None, os nós serão expandidos até que todas as folhas sejam puras. Valores testados: [4,5,6,7,8,9,10]
- col sample by tree: controla a fração de features (variáveis) a serem consideradas em cada árvore durante o treinamento do XGBoost. Ele define a proporção de colunas (features) que são amostradas aleatoriamente em cada árvore. Valores testados: [2,4,6,8]
- learning rate: controla a taxa de aprendizado do modelo. É um fator que multiplica as atualizações dos gradientes durante o processo de treinamento. Um valor maior de "learning rate" permite que o modelo se ajuste mais rapidamente aos dados, mas também pode levar a uma maior probabilidade de overfitting. Valores testados: [0.05, 0.1, 0.3]

Para a LSTM, as distintas arquiteturas de rede estudadas foram:

- Rede 1:
- model = Sequential()
- model.add(LSTM(units=2))
- model.add(Dense(units=1))
- model.add(Dense(units=1, activation='sigmoid'))
- model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['binary_crossentropy'])
- epochs = 10 ; batch_size = 32

- Rede 2:
- model = Sequential()
- model.add(LSTM(units=4))
- model.add(Dense(units=1))
- model.add(Dense(units=1, activation='sigmoid'))
- model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['binary_crossentropy'])
- epochs = 10 ; batch_size = 32

- Rede 3:
- model = Sequential()
- model.add(LSTM(units=8))
- model.add(Dense(units=2))
- model.add(Dense(units=1, activation='sigmoid'))
- model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['binary_crossentropy'])
- epochs = 10 ; batch_size = 32

- Rede 4:
- model = Sequential()
- model.add(LSTM(units=12))
- model.add(Dropout(0.2))
- model.add(Dense(units=4))
- model.add(Dense(units=1, activation='sigmoid'))
- model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['binary_crossentropy'])
- epochs = 10 ; batch_size = 32

Além disso, foi escolhida uma divisão dos dados de treinamento e teste em uma proporção de 75% e 25%, respectivamente. Como os gols são relativamente raros (em uma proporção de 1 gol para cada 10 chutes, aproximadamente, considerando-se todas as ligas) a separação foi feita de modo que as proporções de gols fossem equivalentes em ambos os conjuntos de dados. Avaliou-se também a realização de scaling nos atributos do modelo para evitar que pesos maiores fossem atribuídos a variáveis com valores maiores e vice-versa de acordo com o mínimo e máximo de cada feature no dataset de treino, técnica conhecida como *min-max scaling*, dada pela fórmula:

$$x_{\text{scaled}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

em que x é o valor original da feature, x_{min} é o mínimo valor da feature no dataset, x_{max} é o máximo valor da feature no dataset, e x_{scaled} é o valor parametrizado da feature em escala que vai de 0 a 1.

Utilizou-se também validação cruzada no treinamento dos modelos usando-se 5 *folds* em todos os treinamentos realizados, incluindo-se o processo de otimização de hiperparâmetros, em que se utilizou a técnica de *grid search*.

3.3 Métricas

Para algoritmos de classificação, a função de custo padrão usada é a perda logarítmica. Para modelos com *output* binário, como é o caso do modelo de expected goals, a função log loss é dada por:

$$\mathscr{L}(\hat{y}, y) = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

em que $\hat{y}$ é probabilidade prevista da classe positiva, , y é o valor real (0 ou 1), e n é o número de amostras no dataset.

Embora a maioria das métricas de avaliação para problemas de classificação envolva a análise da capacidade preditiva do modelo em várias situações usando métricas tradicionais como precisão e recall, essa abordagem não é ideal ao problema de modelagem de xG. Isso ocorre porque o resultado desejado do modelo é a probabilidade de que um chute específico seja gol, e não uma previsão de se um chute é um

gol (ou seja, uma saída binária). Por este motivo, este trabalho usa como principal métrica comparativa de modelos a própria perda logarítmica, em que quanto menor essa pontuação, melhor o algoritmo de classificação estima com precisão a probabilidade de gol.

Devido à menção em alguns trabalhados pesquisados na revisão bibliográfica, avaliou-se também como métrica secundária a AUC (area under curve), que é a área da curva ROC.

4 RESULTADOS E DISCUSSÃO

4.1 Análise Exploratória

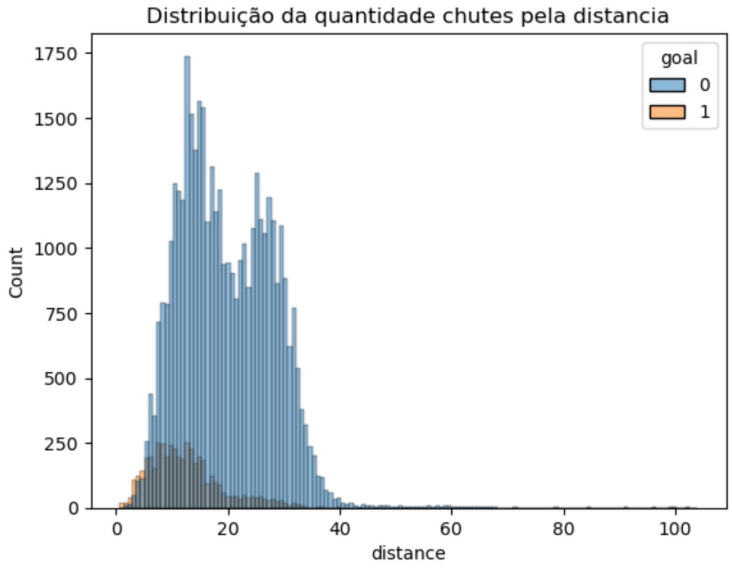
A partir dos dados fornecidos o primeiro passo foi fazer o pré-processamento dos dados, que consistiu em ler os 3 arquivos .json e uni-los em formato tabular (pandas Dataframe).

De posse de um dataset unificado e com base na revisão bibliográfica foram identificadas algumas features que poderiam ser relevantes para a criação de um modelo de expected goals (xG) como ângulo do chute e distância do chute até o gol, calculadas através do atributo *positions*, que traz as coordenadas x e y de cada chute. Além disso, criaram-se variáveis binárias referentes a eventos pré chute, como tipo de passe, se foi lançamento, dividida ou no momento do chute, como por exemplo se o chute era originado de uma falta ou escanteio, entre outros. Neste passo já foi possível excluir diversas variáveis que sabidamente não teriam impacto no modelo, como alguns metadados das partidas (nome do estádio, país) e dos jogadores (data de nascimento, sobrenome). Além disso, nesta etapa inicial verificou-se que não haviam dados faltantes.

Ao final, a base de dados que seguiu para ser utilizada na fase de análise exploratória e modelagem continha 45284 entradas (linhas) e 34 features (colunas).

A etapa inicial da análise exploratória consistiu em avaliar o comportamento das features com relação ao target, que é a coluna ”goal”, que possui valores 0 (não gol) e 1 (gol). Um exemplo de análise, mostrada a seguir, é a distribuição da quantidade de chutes com relação à distância. É possível notar que chutes que resultam em gol tendem a serem feitos mais próximo do gol adversário.

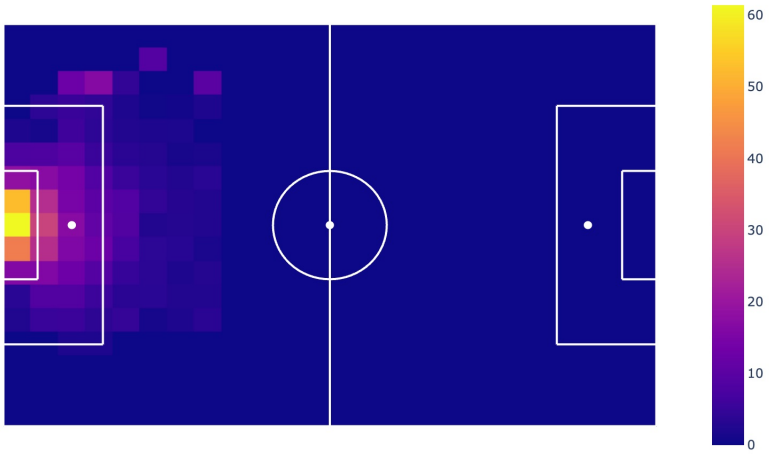
Figure 3. Distribuição do número de gols em relação à distância do chute



Fonte: Do Autor.

Além disso, é possível também calcular a probabilidade de marcar um gol com base na distância em um gráfico estilo *heatmap*. Para isso dividimos o campo em subseções e para cada uma destas subseções dividimos o número de gols marcado naquela região pelo número de chutes para obtermos a probabilidade. O gráfico resultante pode ser observado na Figura 4, a seguir.

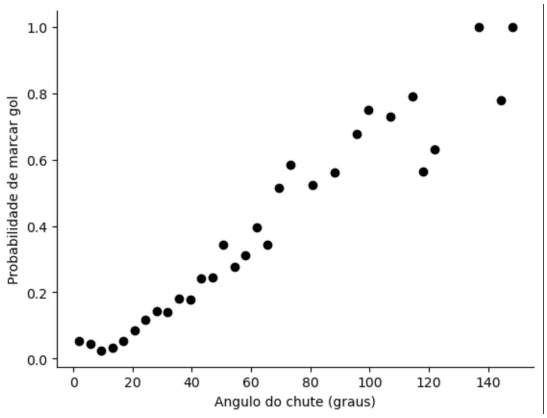
Figure 4. Probabilidade de marcar com relação à posição no campo



Fonte: Do Autor.

A partir das análises gráficas iniciais, duas hipóteses principais foram levantadas: qual era a relação entre distância e ângulo do chute e a probabilidade de se marcar um gol? Para responder a estas questões os gráficos das Figuras 5 e 6 foram construídos.

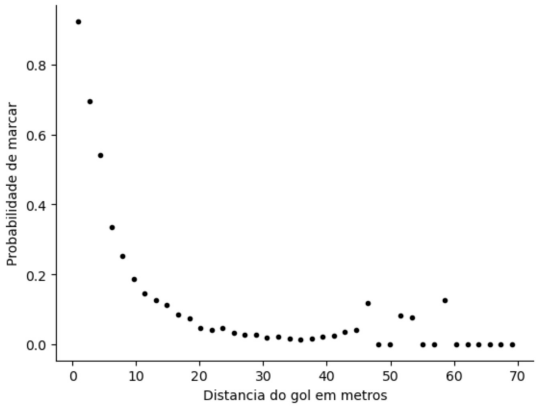
Figure 5. Probabilidade de se marcar um gol com relação ao ângulo do chute



Fonte: Do Autor.

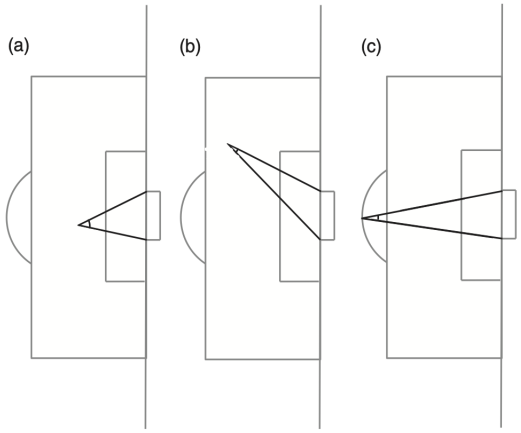
Estes resultados indicaram que o ângulo do chute e distância com relação ao gol adversário poderiam ser features com bom poder preditivo para os modelos almejados. Com relação ao ângulo, a relação é direta: maiores ângulos favorecem a probabilidade de se marcar um gol e com relação à distância a relação é inversa: quanto mais perto do gol, melhor. A Figura 7 a seguir, tirada de Sumpter (2017) ajuda a ilustrar o porquê: o ângulo é calculado a partir do triângulo com vértices na bola e nas duas traves, sendo assim, maiores ângulos significam que o jogador tem uma visão mais ampla do gol adversário. A proximidade ao gol adversário também auxilia neste sentido, como podemos ver ao comparar os diagramas a) e c) na Figura 7.

Figure 6. Probabilidade de se marcar um gol com relação à distância do chute



Fonte: Do Autor.

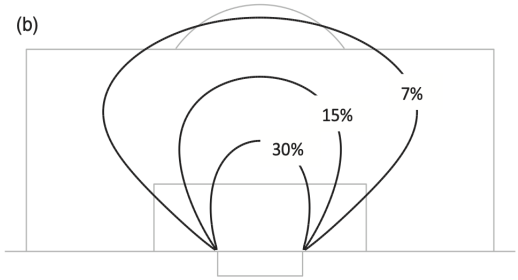
Figure 7. Relação entre ângulo e distância do chute



Fonte: Sumpter (2017)

Este tipo de análise ajuda a explicar uma outra vantagem da expectativa de gols: auxiliar os jogadores na tomada de decisão. A partir das análises anteriores e tendo-se um modelo de xG é possível traçar um diagrama como da Figura 8, em que se tem as probabilidades de marcar dentro da área adversária como em um gráfico estilo de contorno. Ao apresentar um gráfico deste tipo a um jogador é possível orientá-lo a de onde ele pode tentar chutar de dentro da área para maximizar suas chances de marcar um gol.

Figure 8. Probabilidades de marcar dentro da área adversária



Fonte: Sumpter (2017)

4.2 Modelagem

Conforme descrito na seção de material e métodos foi escolhida uma divisão dos dados de treinamento e teste em uma proporção de 75% e 25%, respectivamente. O *scaling* das *features* foi feito usando valores mínimos e máximos e a validação cruzada no treinamento dos modelos usando-se 5 *folds* em todos os treinamentos realizados, incluindo-se o processo de otimização de hiperparâmetros, em que se utilizou a técnica de *grid search*.

A título de referência como baseline para a métrica de perda logarítmica para comparar performance entre os modelos, um modelo usando apenas o valor médio esperado de gols (10.17%) foi usado e obteve um resultado de 0.333 de *log-loss* (perda logarítmica). Em seguida testou-se um modelo de Regressão Logística usando-se todas as features com otimização de hiperparâmetros, com resultado de *log-loss* de 0.281, seguido de um de Random Forest Classifier, cujo melhor resultado foi de 0.280, posteriormente usou-se o LGBM, com resultado equivalente ao Random Forest e, na sequência, obteve-se um resultado ligeiramente melhor com o XGBoost, com *log-loss* de 0.279 e também um resultado um pouco pior com o LSTM, chegando a 0.283. Um resumo dos resultados obtidos pode ser encontrado na tabela a seguir, em que se incluiu os valores de AUC juntamente à perda logarítmica:

Table 1. Resumo dos resultados da modelagem (RL = Regressão Logistica; LGBM = LightGBM, XGB = XGBoost)

Métrica/Modelo	Baseline	RL2	Random Forest	LGBM	XGB	LSTM
Log-loss	0.333	0.281	0.280	0.280	0.279	0.283
AUC	-	76.59%	78.21%	78.39%	78.32%	-

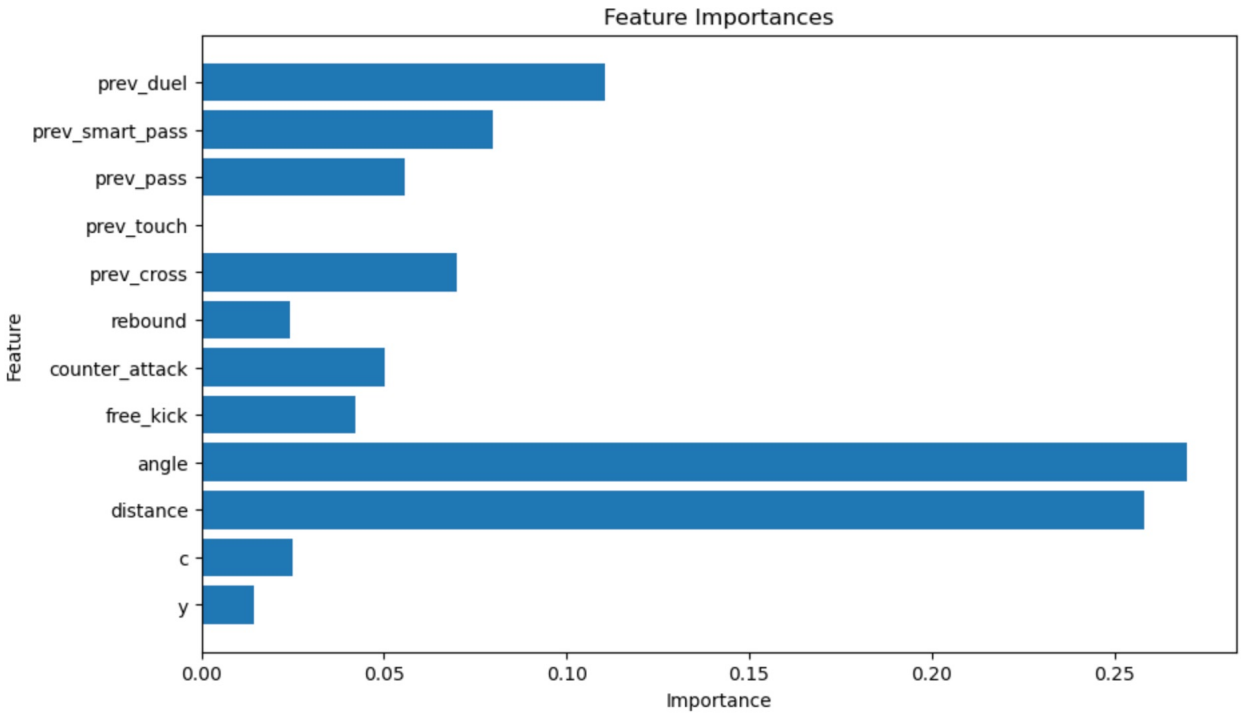
Sendo assim, tem-se que o algoritmo XGBoost forneceu o melhor desempenho, com *log-loss* de 0.279. Os resultados obtidos se aproximam muito dos da revisão bibliográfica, especialmente os relatados por Mead et al. (2023), cujo melhor modelo havia sido para um modelo de XGBoost com *log-loss* de 0.28184. Para Regressão Logistica o resultado foi praticamente idêntico, em que os autores reportaram um valor de *log-loss* de 0.285.

4.3 Importância das features

Para o melhor modelo (XGBoost) avaliou-se as features que mais impactavam o resultado do modelo e o resultado corroborou o que se visualizou na análise exploratória: ângulo e distância do gol são as features que mais impactam o resultado final do modelo. Para tanto, utilizou-se o atributo "feature_impotances_" do modelo treinado. Este método fornece uma pontuação para cada feature, indicando o quanto cada uma contribuiu para a melhoria do desempenho do modelo durante o processo de treinamento. Essa pontuação é calculada com base nas estatísticas dos ganhos de informação ou ganhos de precisão obtidos ao dividir os nós da árvore durante a construção do modelo. O cálculo da importância das *features* no XGBoost é baseado em árvores de decisão. Cada vez que uma feature é utilizada para dividir um nó em uma árvore, o modelo calcula o ganho de informação ou ganho de precisão resultante dessa divisão. O ganho de informação mede a redução na entropia após a divisão, enquanto o ganho de precisão representa a melhoria na precisão da classificação. O XGBoost calcula a importância das *features* somando os ganhos

de informação ou ganhos de precisão para cada *feature* ao longo de todas as árvores do modelo. Em seguida, normaliza essas pontuações em relação à soma total, para que a importância de todas as *features* some 1 ou 100%. Dessa forma, a pontuação resultante representa a proporção de contribuição de cada *feature* para o desempenho geral do modelo.

Figure 9. Features mais importantes para o modelo XGBoost



Fonte: Do Autor

5 CONCLUSÃO

Este projeto teve como principais objetivos contribuir para o conjunto limitado de pesquisas sobre *expected goals* em língua portuguesa e comparar o desempenho dos modelos em relação a estudos anteriores em outros países, utilizando os dados fornecidos pela Wyscout, que se tornou referência para esse tipo de análise. O modelo ótimo desenvolvido neste projeto (XGBoost) demonstrou competitividade em relação aos resultados de pesquisas existentes na literatura. As variáveis mais importantes foram identificadas como o ângulo e a distância do chute, revelando insights interessantes sobre como jogadas de bola parada e contra-ataques podem aumentar a probabilidade de um chute resultar em gol.

Apesar dos resultados interessantes obtidos neste estudo, é importante ressaltar suas limitações, que podem ser abordadas em pesquisas futuras. Uma das principais limitações está relacionada à engenharia de *features*. Existem diversas variáveis que podem ser criadas e calculadas para complementar o conjunto de dados, além das que já foram utilizadas. Por exemplo, o valor do jogador de acordo com o site TransferMarket poderia ser considerado.

Além disso, a modelagem não incluiu dados posicionais devido à indisponibilidade dessas informações no conjunto de dados utilizado. Caso contrário, seria possível criar *features* adicionais, como a posição do goleiro no momento do chute e avaliar a presença de defensores na trajetória do chute, enriquecendo ainda mais a análise. Apesar dessas limitações, os resultados deste estudo comprovam que a expectativa de gols pode trazer um valor significativo para analistas, comissões técnicas e torcedores, ao proporcionar uma nova perspectiva sobre o jogo, que é sujeito a incertezas e aleatoriedades. Os principais benefícios dessa abordagem são a capacidade preditiva do desempenho das equipes, que vai além do simples número de gols marcados, principalmente devido ao fato de o modelo de *expected goals* basear-se em chutes, que ocorrem em um número muito maior do que os gols durante as partidas. Além disso, um bom modelo de *expected goal*s pode auxiliar os jogadores em suas tomadas de decisão sobre quando arriscar um chute, informando as áreas do campo em que a probabilidade de marcar é maior, como ilustrado nas Figuras 7 e 8.

Esses resultados destacam a importância do modelo de *expected goals* como uma ferramenta valiosa para análises no futebol, oferecendo uma perspectiva mais abrangente e informativa sobre as probabilidades de marcar gols. Esperamos que este estudo inspire pesquisas futuras para aprimorar ainda mais os modelos de *expected goals*, explorando novas variáveis e técnicas de análise, e também encoraje a aplicação prática desses modelos em cenários reais, beneficiando equipes, jogadores e entusiastas do futebol.

6 ANEXOS

6.1 Descrição dos dados de partidas e jogadores

Do original em inglês, conforme fornecido pela Wyscout.

Dados de partidas (matches.json)

- competitionId: the identifier of the competition to which the match belongs to. It is a integer and refers to the field ”wyId” of the competition document;
- date and dateutc: the former specifies date and time when the match starts in explicit format (e.g., May 20, 2018 at 8:45:00 PM GMT+2), the latter contains the same information but in the compact format YYYY-MM-DD hh:mm:ss;
- duration: the duration of the match. It can be ”Regular” (matches of regular duration of 90 minutes + stoppage time), ”ExtraTime” (matches with supplementary times, as it may happen for matches in continental or international competitions), or ”Penalities” (matches which end at penalty kicks, as it may happen for continental or international competitions);
- gameweek: the week of the league, starting from the beginning of the league;
- label: contains the name of the two clubs and the result of the match (e.g., ”Lazio - Internazionale, 2 - 3”);
- roundID: indicates the match-day of the competition to which the match belongs to. During a competition for soccer clubs, each of the participating clubs plays against each of the other clubs twice, once at home and once away. The matches are organized in match-days: all the matches in match-day i are played before the matches in match-day i + 1, even tough some matches can be anticipated or postponed to facilitate players and clubs participating in Continental or Intercontinental competitions. During a competition for national teams, the ”roundID” indicates the stage of the competition (eliminatory round, round of 16, quarter finals, semifinals, final);
- seasonId: indicates the season of the match;
- status: it can be ”Played” (the match has officially finished), ”Cancelled” (the match has been canceled for some reason), ”Postponed” (the match has been postponed and no new date and time is available yet) or ”Suspended” (the match has been suspended and no new date and time is available yet);
- venue: the stadium where the match was held (e.g., ”Stadio Olimpico”);
- winner: the identifier of the team which won the game, or 0 if the match ended with a draw;
- wyId: the identifier of the match, assigned by Wyscout;
- teamsData: it contains several subfields describing information about each team that is playing that match: such as lineup, bench composition, list of substitutions, coach and scores: - hasFormation: it has value 0 if no formation (lineups and benches) is present, and 1 otherwise;
- score: the number of
goals scored by the team during the match (not counting penalties); - scoreET: the number of goals scored by the team during the match, including the extra time (not counting penalties);
- scoreHT: the number of goals scored by the team during the first half of the match;
- scoreP: the total number of goals scored by the team after the penalties; - side: the team side in the match (it can be ”home” or ”away”);
- teamId: the identifier of the team;
- coachId: the identifier of the team’s coach;
- bench: the list of the team’s players that started the match in the bench and some basic statistics about their performance during the match (goals, own goals, cards);
- lineup: the list of the team’s players in the starting lineup and some basic statistics about their performance during the match (goals, own goals, cards);
- substitutions: the list of team’s substitutions during the match, describing the players involved and the minute of the substitution.

Dados de jogadores (players.json)

- birthArea: geographic information about the player’s birth area;
- birthDate: the birth date of the player, in the format ”YYYY-MM-DD”;
- currentNationalTeamId: the identifier of the national team where the players currently plays;
- currentTeamId: the identifier of the team where the player plays for. The identifier refers to the field ”wyId” in a team document;

- firstName: the first name of the player;
- lastName: the last name of the player;
- foot: the preferred foot of the player;
- height: the height of the player (in centimeters);
- middleName: the middle name (if any) of the player;
- passportArea: the geographic area associated with the player's current passport;
- role: the main role of the player. It is a subdocument containing the role's name and two abbreviations of it;
- shortName2: the short name of the player;
- weight: the weight of the player (in kilograms);
- wyId: the identifier of the player, assigned by Wyscout.

REFERENCES

Biermann, C. (2019). *Football Hackers: The Science and Art of a Data Revolution*. Blink Publishing.

Exame (2022). Estudo aponta os clubes mais valiosos do brasil. acessado em 06-05-2023 em https://exame.com/esporte/estudo-aponta-os-clubes-mais-valiosos-do-brasil-veja-o-ranking/.

Forbes (2022). The world's most valuable soccer teams 2022: Real madrid, worth 5.1 billion, is back on top. acessado em 05-05-2023 em https://www.forbes.com/sites/mikeozanian/2022/05/26/the-worlds-most-valuable-soccer-teams-2022-real-madrid-worth-51-billion-back-on-top/.

Mead, J., O'Hare, A., and McMenemy, P. (2023). Expected goals in football: Improving model performance and demonstrating value. *PLoS ONE*, 18(4).

Nielsen (2022). The 2022 world football report. acessado em 05-05-2023 em https://www.nielsen.com/wp-content/uploads/sites/2/2022/07/nielsen-world-football-report-2022.pdf.

Pappalardo, L., Cintia, P., and Rossi, A. (2019). A public data set of spatio-temporal match events in soccer competitions. *Scientific Data*, 236(6).

Sumpter, D. (2017). *Soccermatics: Mathematical Adventures in the Beautiful Game*. Bloomsbury Sigma.

Tempo, J. O. (2021). Galo é pioneiro e lança setor analytics, que 'transforma' dados em conhecimento, jornal o tempo. acessado em 06-05-2023 em https://www.otempo.com.br/sports/atletico/galo-e-pioneiro-e-lanca-setor-analytics-que-transforma-dados-em-conhecimento-1.2486872.

Umami, I., Gutama, D. H., and Hatta, H. R. (2021). Implementing the expected goal (xg) model to predict scores in soccer matches. *International Journal of Informatics and Information Systems*, 4(1).