

# PREVISÃO DE RESULTADOS

MÉTODOS NUMÉRICOS E OTIMIZAÇÃO NÃO LINEAR



Realizado por:

Diogo Paiva	A100760
João Magalhães	A100740
Jorge Rodrigues	A101758
Rodrigo Gomes	A100555





## PERGUNTA AO CHATGPT

**"Há algum problema real que se possa resolver com os temas: Splines e Equações não lineares no ramo do futebol?"**

Sim, os conceitos de splines e equações não lineares também podem ser aplicados no ramo do futebol. Aqui estão alguns exemplos de problemas que podem ser resolvidos com esses temas:

1. Previsão de resultados de jogos: é possível utilizar splines para modelar o desempenho de uma equipe ao longo do tempo, levando em consideração fatores como a formação tática, a condição física dos jogadores e o histórico de resultados. As equações não lineares podem ser usadas para ajustar a curva de forma mais precisa e prever o resultado de um jogo futuro.
2. Análise do desempenho dos jogadores: a análise do desempenho dos jogadores é essencial para melhorar o desempenho da equipe. Através do uso de splines, é possível modelar o desempenho dos jogadores ao longo do tempo e identificar tendências. As equações não lineares podem ser usadas para ajustar esses modelos e identificar padrões mais precisos.



## FALHA DO CHATGPT

**"Que tipo de equação não linear devemos usar para o tema de previsão de jogos."**

A equação logística pode ser escrita como:

$$P(t) = \frac{K}{1 + e^{-r(t-t_0)}}$$

onde:

- $P(t)$  é a pontuação da equipe em um determinado momento  $t$ ;
- $K$  é a pontuação máxima que a equipe pode atingir;
- $r$  é a taxa de crescimento da pontuação;
- $t_0$  é o tempo de inflexão da curva, que indica o momento em que a pontuação começa a crescer mais rapidamente.



## CÓDIGO DO CHATGPT

**"Consegues usar uma  
função spline do MATLAB  
para prever o número de  
golos num ponto num  
intervalo de 30 jogos?"**

```
% Define the x and y coordinates of the data points
x = [1 2 3 4 6 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29];
y = [2 1 1 1 2 3 2 1 3 2 2 2 1 1 1 1 3 2 4 2 1 1 1 1 1 0 1];

% Define the x values where you want to predict the
number of goals
xx = 1:0.1:5;

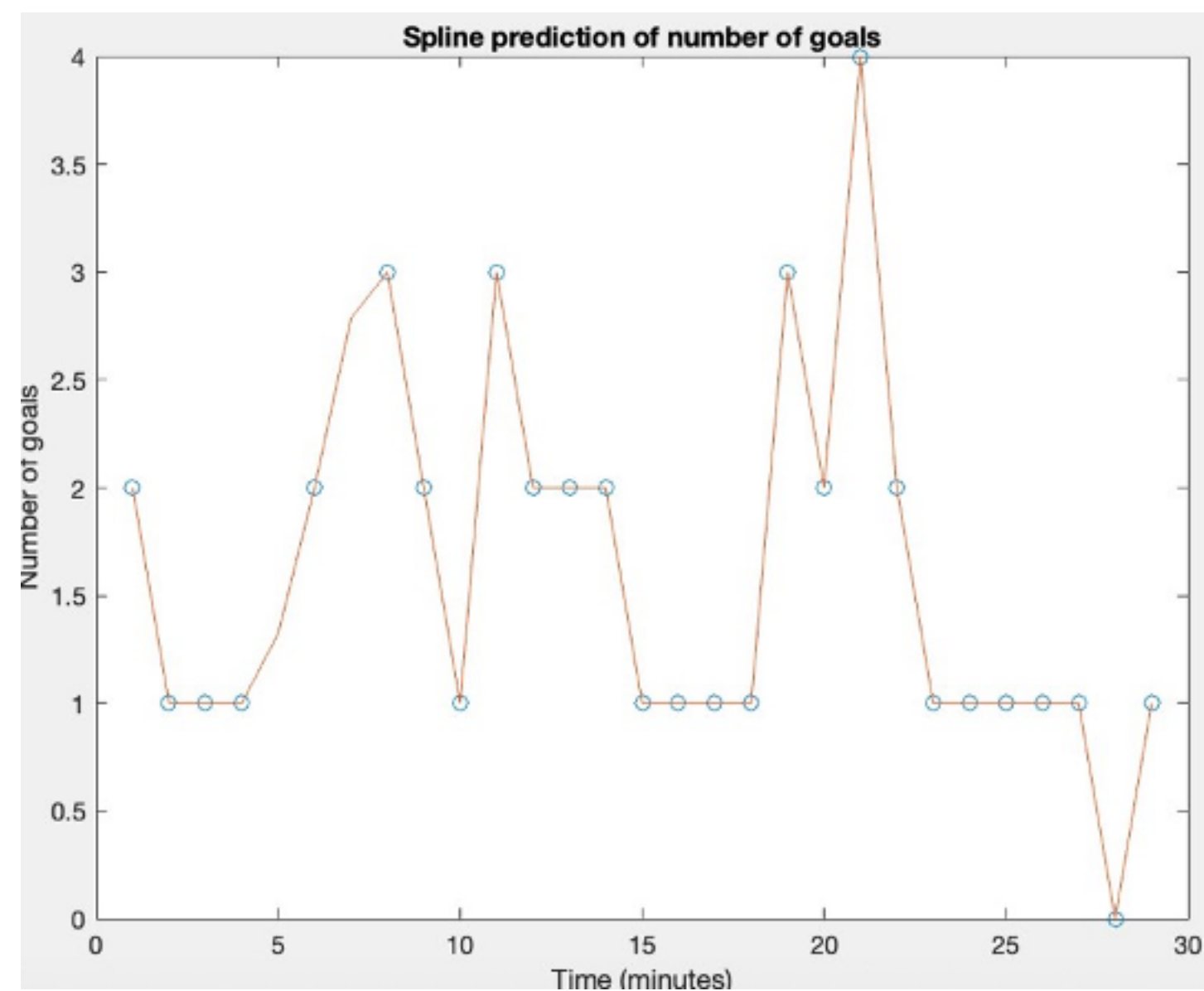
% Create the spline
yy = spline(x,y,xx);

% Plot the results
plot(x,y,'o',xx,yy);
xlabel('Time (minutes)'); ylabel('Number of goals');
title('Spline prediction of number of goals');
```





# GRÁFICO RESULTANTE





## **PERGUNTAS AO CHATGPT**

- **Há alguma forma de obter uma previsão mais realista alterando os valores de interpolação?**
- **Como diminuir a diferença entre as previsões feitas e os resultados reais?**
- **Tendo em conta que se pretende, através de um conjunto de dados, prever um resultado futuro de um jogador, como é que obtemos uma previsão mais parecida com a realidade?**
- **E em termos de precisão do resultado, como é que eu sei qual é o melhor valor para usar na interpolação?**



## CÓDIGO DO CHATGPT

"Em termos de **precisão**  
do resultado, como é que  
eu sei qual é o **melhor**  
**valor** para usar na  
**interpolação?**"

```
% Define the x and y coordinates of the data points
x = [1 2 3 4 6 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29];
y = [2 1 1 1 2 3 2 1 3 2 2 2 1 1 1 1 3 2 4 2 1 1 1 1 0 1];

x_new = 30;

% Divide se os dados num grupo de treino e teste
idx = randperm(length(x));
train_idx = idx(1:floor(length(idx)*0.8)); % 80% para treino
test_idx = idx(length(train_idx)+1:end); % 20% para teste

x_train = x(train_idx);
y_train = y(train_idx);
x_test = x(test_idx);
y_test = y(test_idx);

% Testar diferentes números para a interpolação
num_points_to_test = [2, 4, 8, 16];
rmse = zeros(length(num_points_to_test), 1);

for i = 1:length(num_points_to_test) % Interpolar usando os dados de treino
    x_interp = linspace(min(x_train), max(x_train), num_points_to_test(i));
    y_interp = interp1(x_train, y_train, x_interp, 'pchip');

    % Avaliar usando os dados de teste
    y_pred = interp1(x_interp, y_interp, x_test, 'pchip');
    rmse(i) = sqrt(mean((y_pred - y_test).^2));
end

% Plotar os resultados
figure; plot(num_points_to_test, rmse, 'o-');
xlabel('Number of interpolation points');
ylabel('RMSE');
```

# CÓDIGO FINAL 1/3

```
% randperm - permuta aleatoriamente números de 1 a (length x).  
% ex: randperm(5) pode retornar o conjunto [3, 5, 1, 4, 2].  
idx = randperm(length(x));  
  
% Guarda num vetor 80% dos números do índice anterior para treinar o programa.  
train_idx = idx(1:floor(length(idx)*0.8)); % floor -> arredondado às unidades.  
  
% Guarda os restantes 20%, para testar o mesmo.  
test_idx = idx(length(train_idx)+1:end);  
  
% Guarda os valores de x do vetor train_idx.  
x_train = x(train_idx);  
  
% Guarda os valores de y do vetor train_idx.  
y_train = y(train_idx);  
  
% Guarda os valores de x do vetor test_idx.  
x_test = x(test_idx);  
  
% Guarda os valores de y do vetor test_idx.  
y_test = y(test_idx);
```

```
% Vetor com números aleatórios para testar a interpolação.  
num_points_to_test = [2,3,4,5,6,7,8,16];
```

```
% Cria um vetor 'rmse' com o tamanho igual ao de 'num_points_to_teste', preenche a  
% zero com a função 'zeros()' e através do parâmetro 1 na função, o vetor  
% criado como coluna, invés de uma linha (facilitar calculos futuros).  
rmse = zeros(length(num_points_to_test), 1);
```



## CÓDIGO FINAL 2/3

```
% Enquanto i entre 1 e o tamanho do vetor de `num_points_to_test` aplica o código.
for i = 1:length(num_points_to_test)
    n_interp_points = num_points_to_test(i);

    % Cria um vetor, x_interp, com pontos igualmente espaçados entre o min e o max de x_train.
    x_interp = linspace(min(x_train), max(x_train), n_interp_points);

    % Calcula os valores correspondentes de y para cada ponto de interpolação x_interp.
    y_interp = interp1(x_train, y_train, x_interp, 'pchip');
    % pchip é um método de interpolação polinomial cúbica, muito útil quando os dados apresentam variações bruscas
    % pois evita que se crie oscilações ou saltos nos dados interpolados ao usar curvas suaves.

    % interp1 é usada para interpolar(encontrar) cada valor de y_interp no respectivo x_interp,
    % criando uma curva suave que passa por esses pontos.
    % Depois esta é usada para extrapolar (processo de estimar ou prever um valor desconhecido)
    % os valores da curva suave para o conjunto de valores x_test, produzindo os valores previstos y_pred.
    y_pred = interp1(x_interp, y_interp, x_test, 'pchip');

    % Cálculo do erro quadrático em cada ponto interpolado.
    rmse(i) = sqrt(mean((y_pred - y_test).^2));

    % Verifica se o número de pontos usados para a interpolação é maior que
    % o número de pontos existentes no vetor x_train, se for, sai.
    if n_interp_points > length(x_train)
        break;
    end
end
```

## CÓDIGO FINAL 3/3

```
% Plot the results
figure; % Cria janelas para mostrar os gráficos
% Plota a variável rmse no eixo y e a variável num_points_to_test no eixo
% x, marca os pontos com 'o' e conecta-os com uma linha.
plot(num_points_to_test, rmse, 'o-');
xlabel('Number of interpolation points'); % Legenda do Eixo x
ylabel('RMSE'); % Legenda do eixo y
title('Cross-validation results');
```

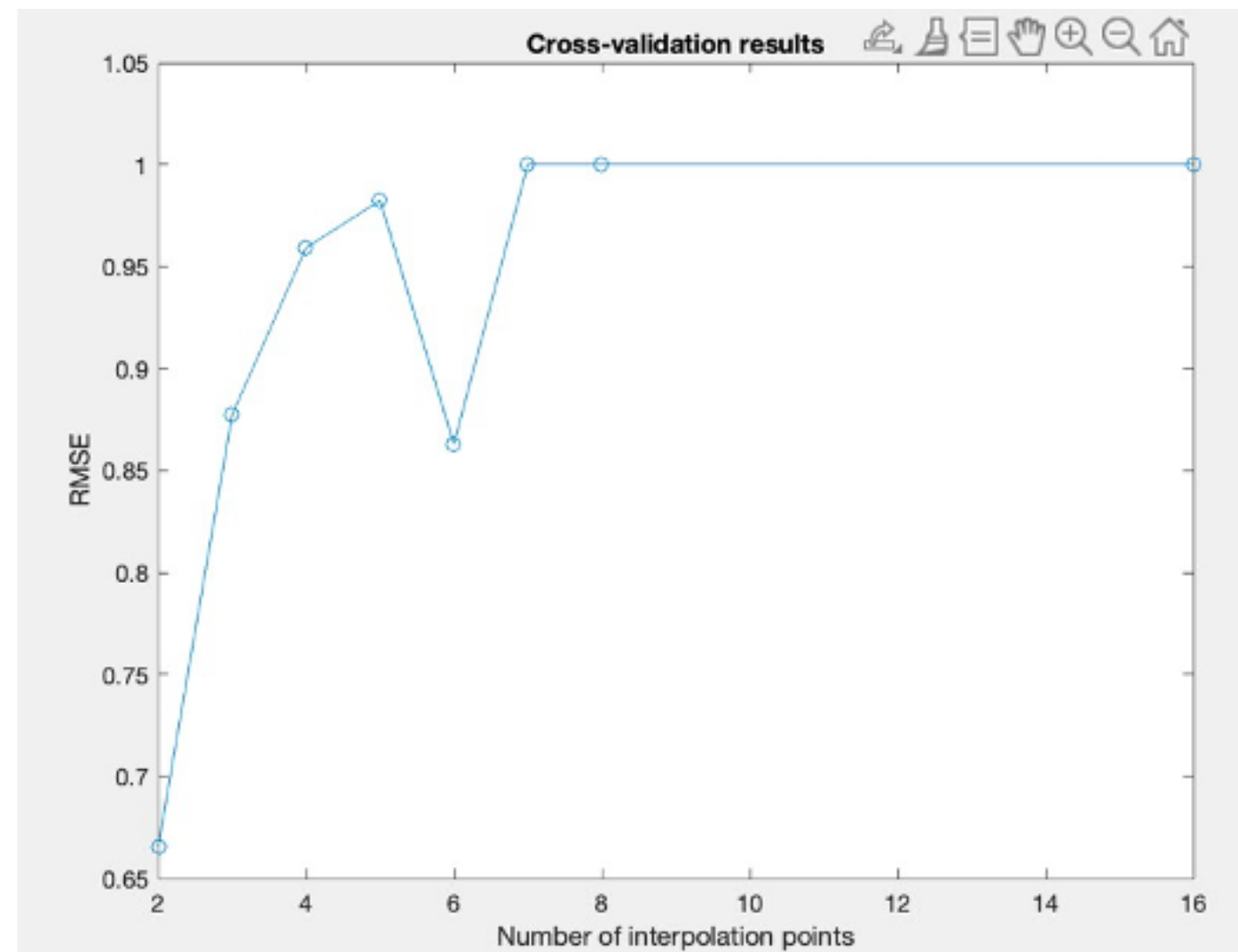


# SCRIPT AUXILIAR

```
x = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29];  
y = [0 1 1 1 3 0 3 0 0 1 0 0 0 0 0 1 0 1 0 2 0 0 0 1 0 0 0 0 1];  
  
% Jornada a prever o número de golos.  
x_new = 30;  
  
% Gera 8 pontos (entre o min e o max de x) igualmente espaçados para interpolação.  
x_interp = linspace(min(x), max(x), 8);  
  
y_interp = interp1(x, y, x_interp, 'pchip');  
  
y_new = interp1(x_interp, y_interp, x_new, 'pchip');  
  
% disp - Converte a frase dada como parâmetro numa string para devolver como output.  
disp(['The predicted number of goals at ' num2str(x_new) ' is ' num2str(round(y_new))]);
```



## GRÁFICO RESULTANTE



```
>> mnl2
```

The predicted number of goals at 30 is 1



# OUTROS EXEMPLOS DE APLICAÇÃO

Diogo Leite Número de alívios no jogo 21.

JOGO	1	2	3	4	5	6	7	8	9	10
ALÍVIOS NO JOGO	5	5	0	2	0	2	4	6	8	2

JOGO	11	12	13	14	15	16	17	18	19	20
ALÍVIOS NO JOGO	1	5	8	5	4	3	1	3	1	6

Jogo 21: 3 alívios  
Previsão: 6 alívios  
Erro: 3

# OUTROS EXEMPLOS DE APLICAÇÃO

Rafael Leão Número de golos no jogo 22.

JOGO  
ALÍVIOS NO JOGO

1	2	3	4	5	6	7	8	9	10
2	0	0	0	0	0	0	1	0	1

JOGO  
ALÍVIOS NO JOGO

11	12	13	14	15	16	17	18	19	20	21
1	0	0	0	0	1	0	2	0	1	0

Jogo 22: 0 alívios  
Previsão: 0 alívios  
Erro: 0



A wide-angle, low-perspective shot of a soccer stadium at night. The pitch is a vibrant green, with white lines marking the field. In the foreground, the corner of a goal is visible on the right. The stands are filled with spectators, and the stadium is illuminated by a long row of bright floodlights along the top edge of the seating area. The sky above is dark and cloudy, with some light breaking through. A dark green rectangular box is superimposed over the center of the image, containing white text.

**DIFICULDADES ENCONTRADAS**



A wide-angle photograph of a soccer stadium at night. The pitch is a vibrant green, and the stands are filled with spectators. Bright stadium lights illuminate the scene, creating a dramatic atmosphere. A goal is visible on the right side of the frame. A dark green rectangular box is superimposed over the center of the image, containing the word 'CONCLUSÃO' in white, bold, uppercase letters.

# CONCLUSÃO