Cálculo Numérico Kalimera

Participantes:

Felipe Duarte

João Fahning

Tiago Trindade

Nicholas Rodrigues

Thiago Leal

Fernando d'Ávila

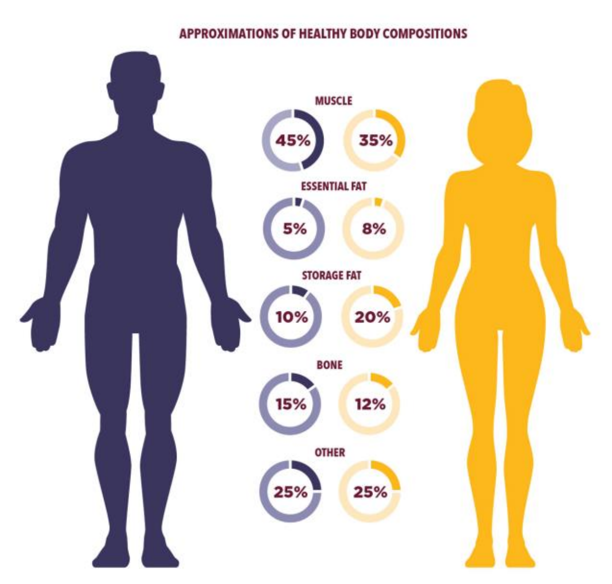
Euro Da Cunha

Renan Gondim

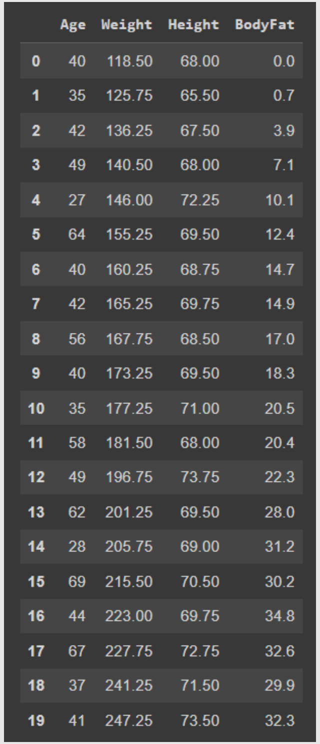
01

Introdução

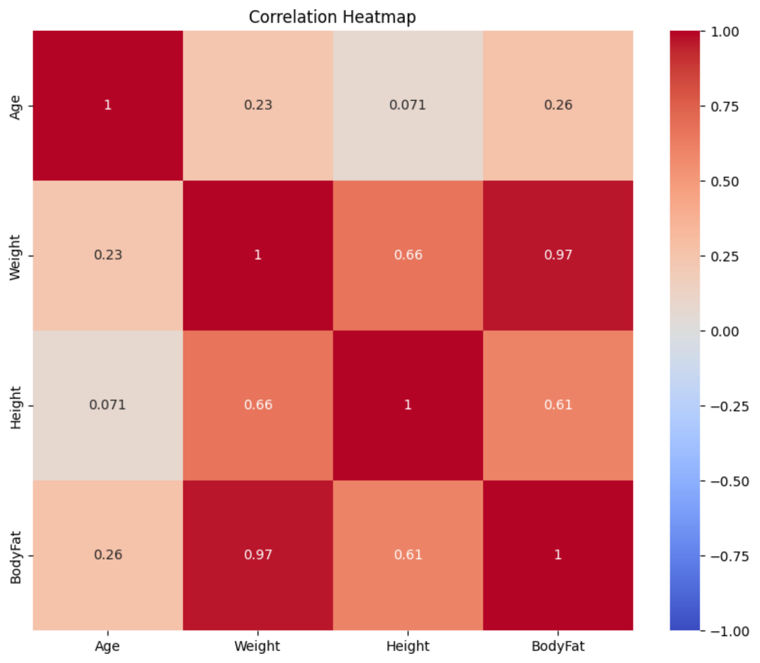
Contextualização



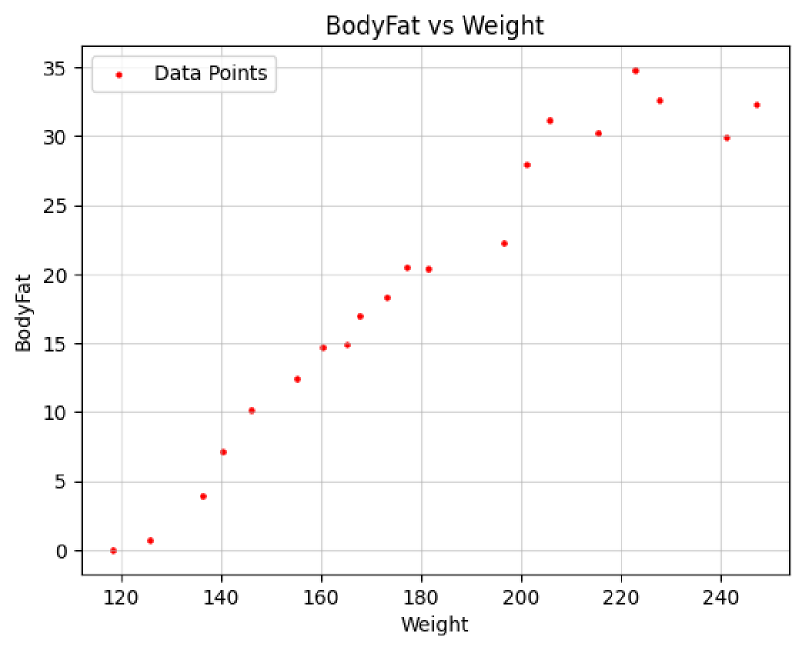
DADOS

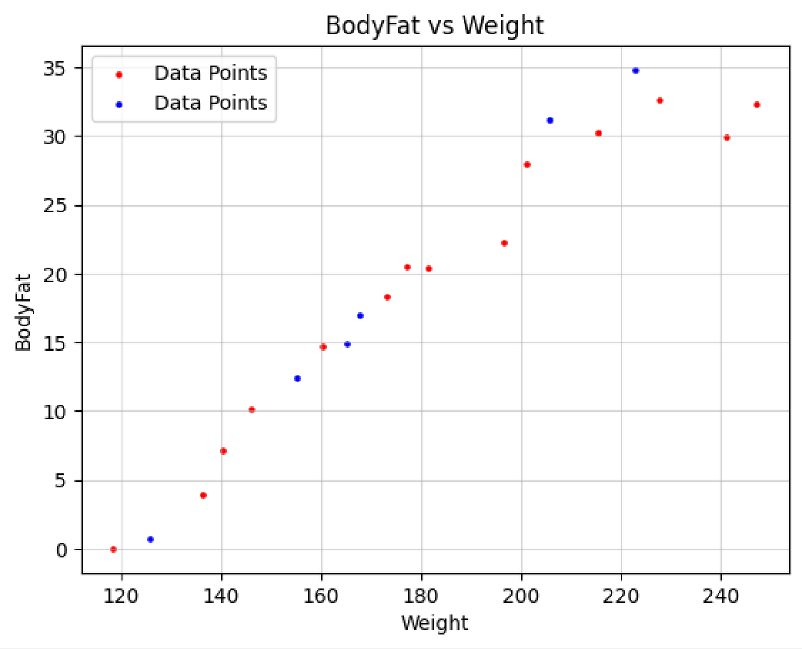


	Age	Weight	Height	BodyFat
0	40	118.50	68.00	0.0
1	35	125.75	65.50	0.7
2	42	136.25	67.50	3.9
3	49	140.50	68.00	7.1
4	27	146.00	72.25	10.1
5	64	155.25	69.50	12.4
6	40	160.25	68.75	14.7
7	42	165.25	69.75	14.9
8	56	167.75	68.50	17.0
9	40	173.25	69.50	18.3
10	35	177.25	71.00	20.5
11	58	181.50	68.00	20.4
12	49	196.75	73.75	22.3
13	62	201.25	69.50	28.0
14	28	205.75	69.00	31.2
15	69	215.50	70.50	30.2
16	44	223.00	69.75	34.8
17	67	227.75	72.75	32.6
18	37	241.25	71.50	29.9
19	41	247.25	73.50	32.3



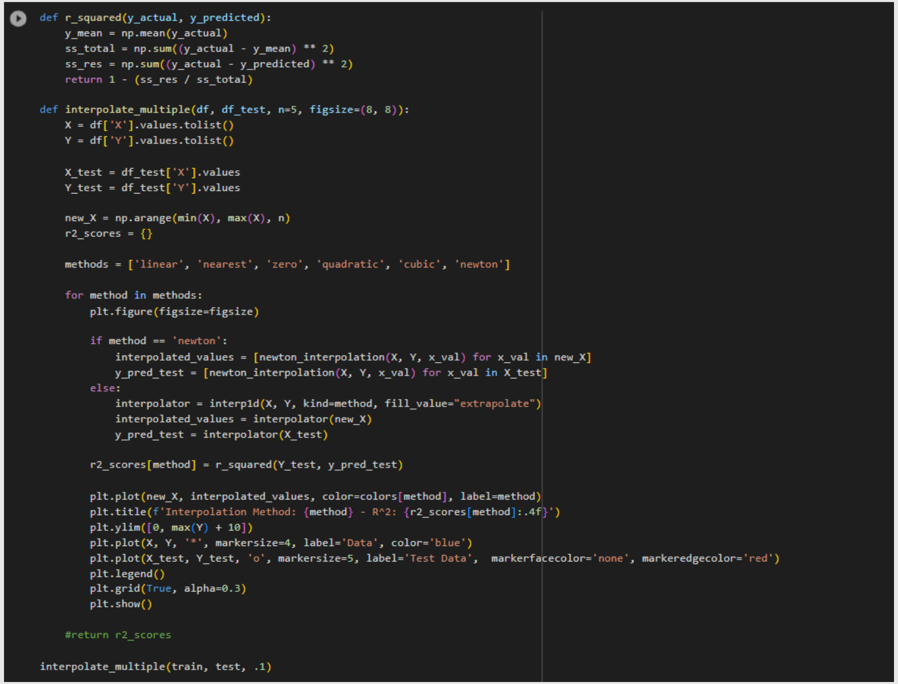
Mergulhando em Números





INTERPOLAÇÃO

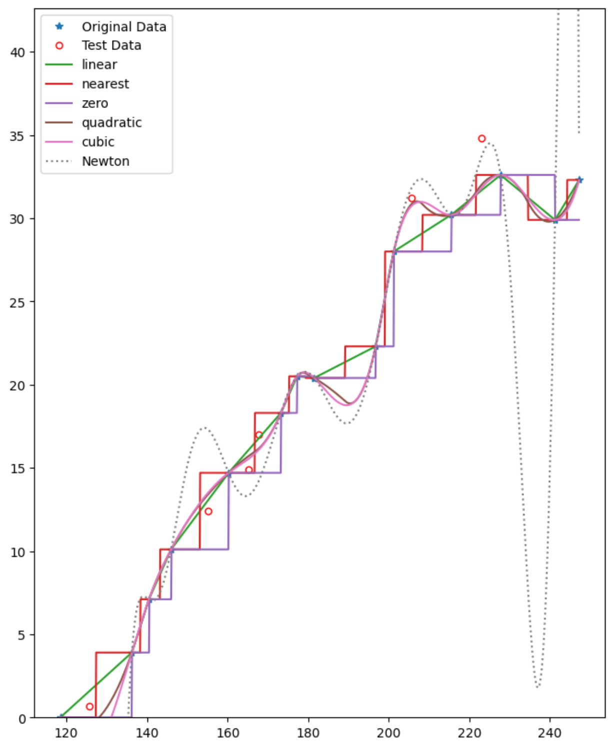
Métodos de Interpolação



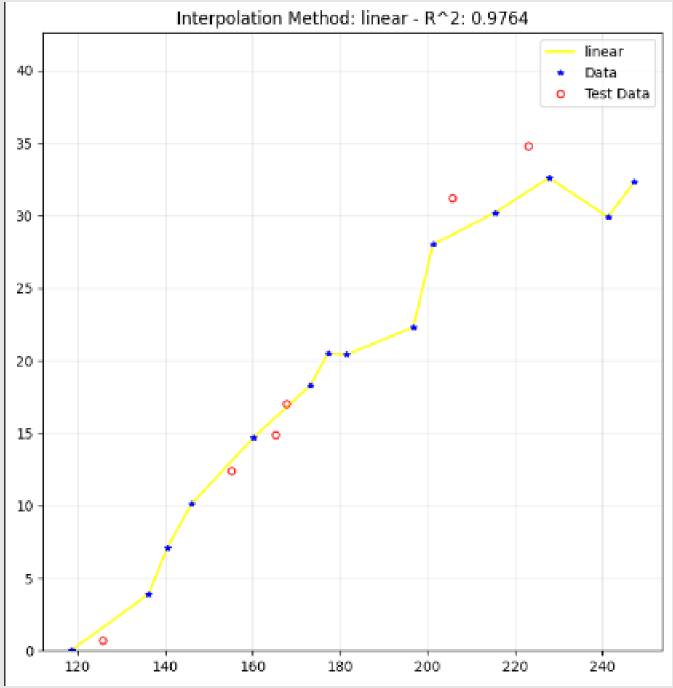
```python
def r_squared(y_actual, y_predicted):
    y_mean = np.mean(y_actual)
    ss_total = np.sum((y_actual - y_mean) ** 2)
    ss_res = np.sum((y_actual - y_predicted) ** 2)
    return 1 - (ss_res / ss_total)

def interpolate_multiple(df, df_test, n=5, figsize=(8, 8)):
    X = df['X'].values.tolist()
    Y = df['Y'].values.tolist()

    X_test = df_test['X'].values
    Y_test = df_test['Y'].values

    new_X = np.arange(min(X), max(X), n)
    r2_scores = {}

    methods = ['linear', 'nearest', 'zero', 'quadratic', 'cubic', 'newton']

    for method in methods:
        plt.figure(figsize=figsize)

        if method == 'newton':
            interpolated_values = [newton_interpolation(X, Y, x_val) for x_val in new_X]
            y_pred_test = [newton_interpolation(X, Y, x_val) for x_val in X_test]
        else:
            interpolator = interp1d(X, Y, kind=method, fill_value="extrapolate")
            interpolated_values = interpolator(new_X)
            y_pred_test = interpolator(X_test)

        r2_scores[method] = r_squared(Y_test, y_pred_test)

        plt.plot(new_X, interpolated_values, color=colors[method], label=method)
        plt.title(f'Interpolation Method: {method} - R^2: {r2_scores[method]:.4f}')
        plt.ylim([0, max(Y) + 10])
        plt.plot(X, Y, '*', markersize=4, label='Data', color='blue')
        plt.plot(X_test, Y_test, 'o', markersize=5, label='Test Data',  markerfacecolor='none', markeredgecolor='red')
        plt.legend()
        plt.grid(True, alpha=0.3)
        plt.show()

    #return r2_scores

interpolate_multiple(train, test, .1)
```

- LINEAR
- NEAREST
- ZERO
- QUADRATIC
- CUBIC
- NEWTON

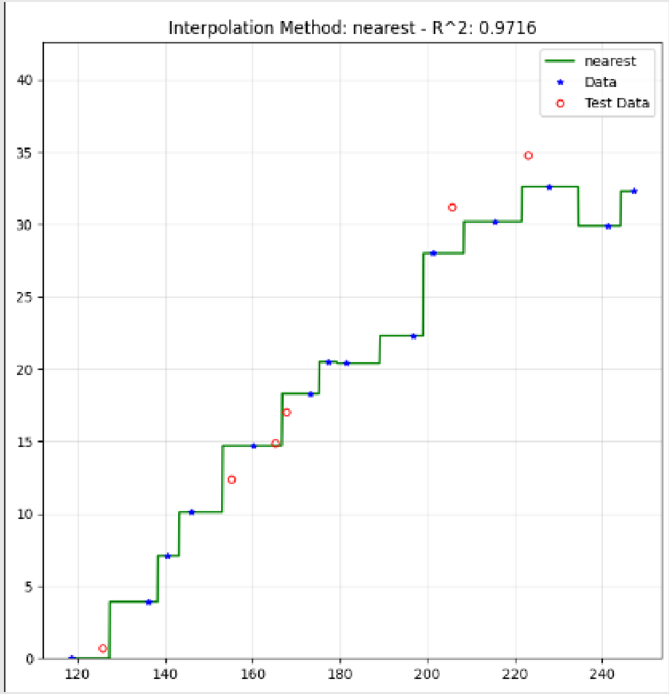
ALL


Original Data
Test Data
linear
nearest
zero
quadratic
cubic
Newton
0
5
10
15
20
25
30
35
40
120
140
160
180
200
220
240

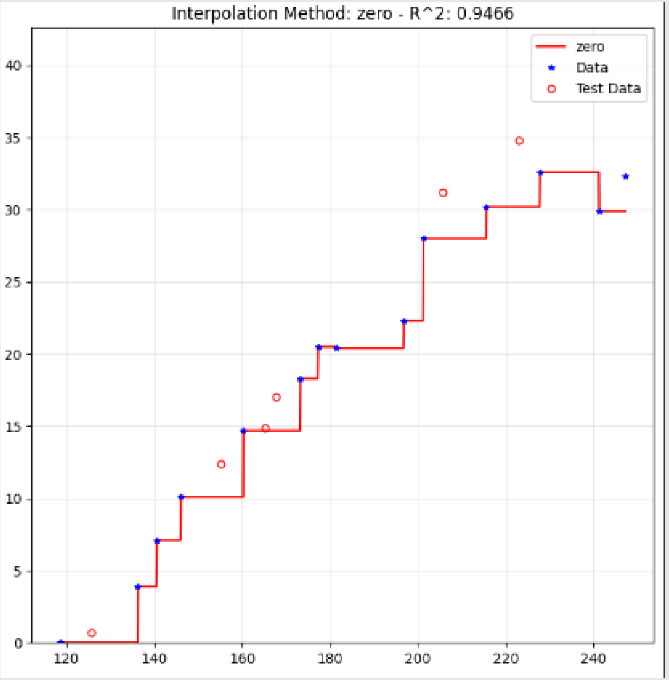
LINEAR


Interpolation Method: linear - R^2: 0.9764
linear
Data
Test Data
0
5
10
15
20
25
30
35
40
120
140
160
180
200
220
240

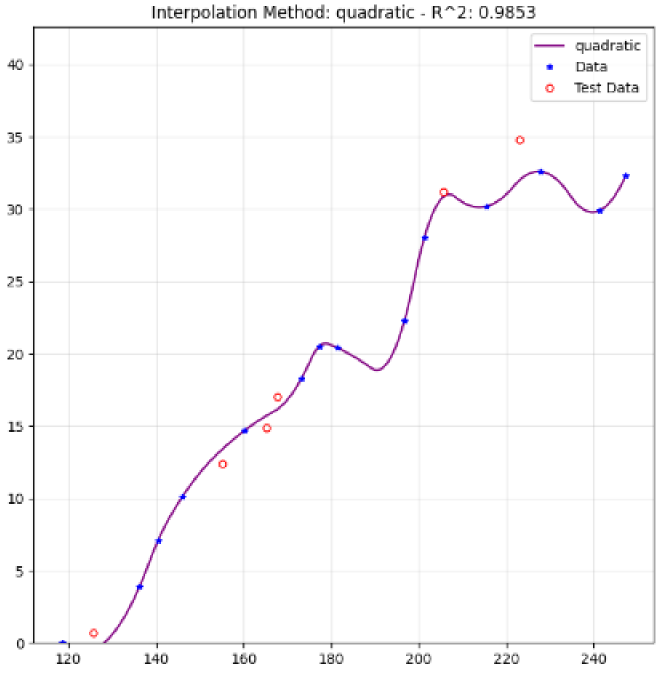
NEAREST


Interpolation Method: nearest - R^2: 0.9716
nearest
Data
Test Data
0
5
10
15
20
25
30
35
40
120
140
160
180
200
220
240

ZERO

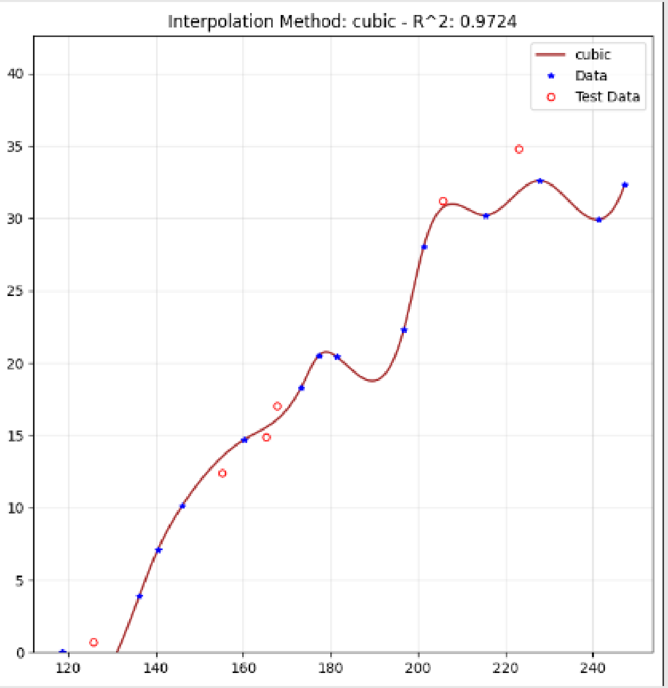

Interpolation Method: zero - R^2: 0.9466
zero
Data
Test Data
0
5
10
15
20
25
30
35
40
120
140
160
180
200
220
240

QUADRATIC

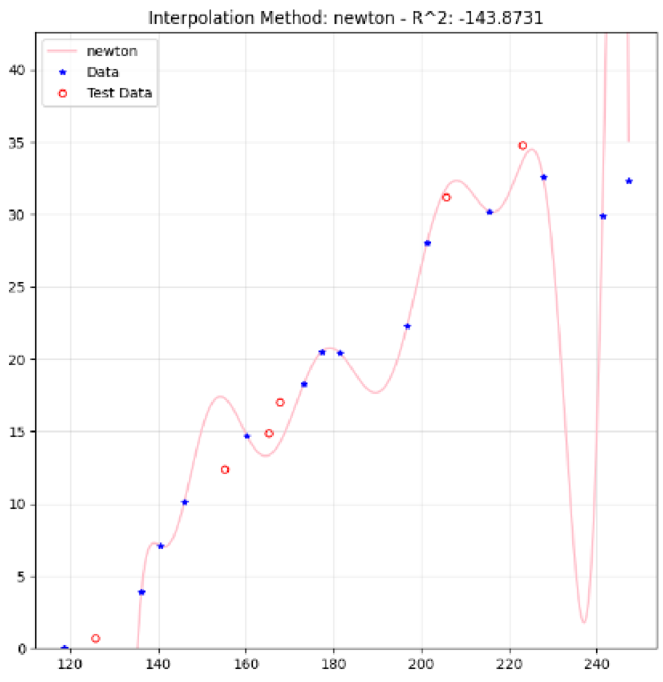


Interpolation Method: quadratic - R^2: 0.9853
quadratic
Data
Test Data
0
5
10
15
20
25
30
35
40
120
140
160
180
200
220
240

CUBIC

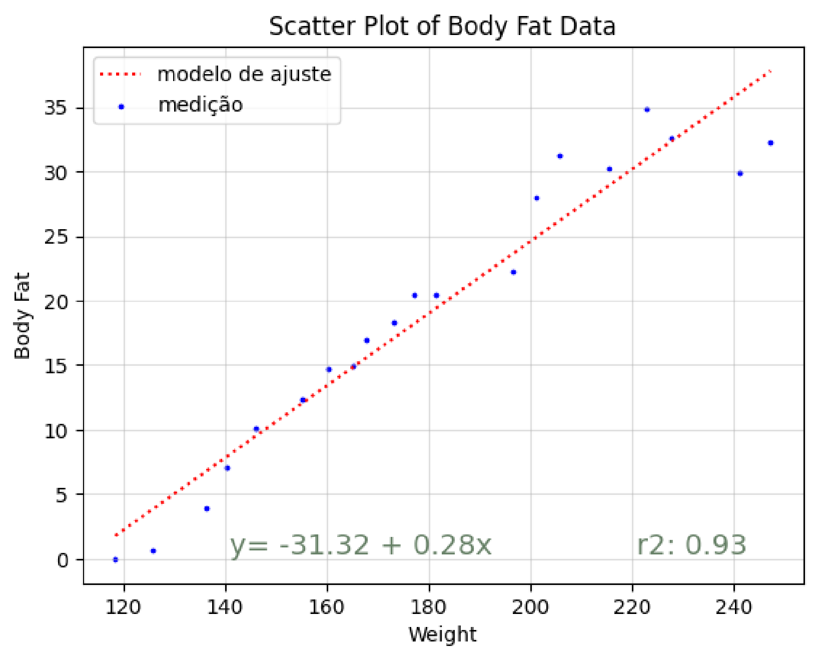

Interpolation Method: cubic - R^2: 0.9724
cubic
Data
Test Data
0
5
10
15
20
25
30
35
40
120
140
160
180
200
220
240

NEWTON


Interpolation Method: newton - R^2: -143.8731
newton
Data
Test Data
0
5
10
15
20
25
30
35
40
120
140
160
180
200
220
240

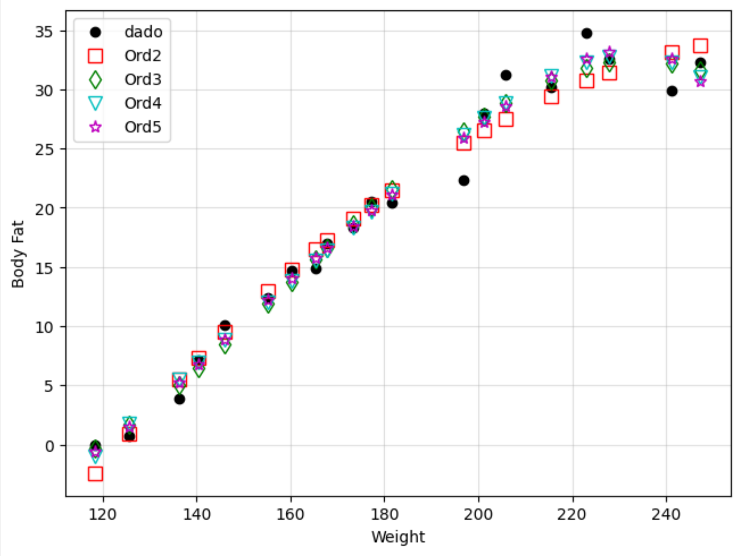
REGRESSÃO LINEAR

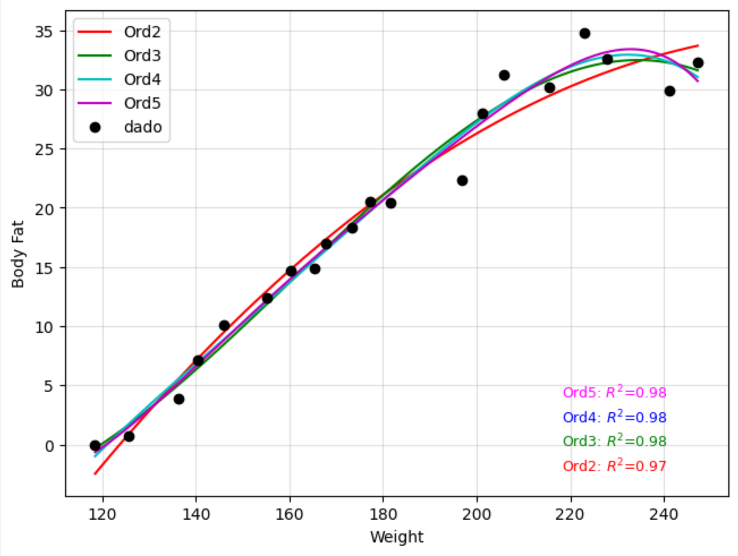
LINEAR REGRESSION



AJUSTE POLINOMIAL LINEAR

POLYNOMIAL LINEAR FIT





Fim

Referências

- Dr. A. Garth Fisher
- https://www.kaggle.com/datasets/fedesoriano/body-fat-prediction-dataset