

Regressão Linear Múltipla

$$y = a_0 + a_1.X_1 + a_2.X_2 + a_3.X_3 + \dots + a_n.X_n$$

y = Variável Dependente (o valor que estamos prevendo)

a₀ = constante (ponto onde y é interceptado → x=0)

a₁ = coeficiente angular (declividade da linha)

X₁ = Variável Independente (Preditor)

n = número de características (variáveis)

Exemplo:

y = quantidade de batata

a₀ = 8 toneladas

a₁ = 1Kg de fertilizante gera +3t de batata

X₁ = quantidade de fertilizante

a₂ = +1o Celsius gera -0.5t de batata

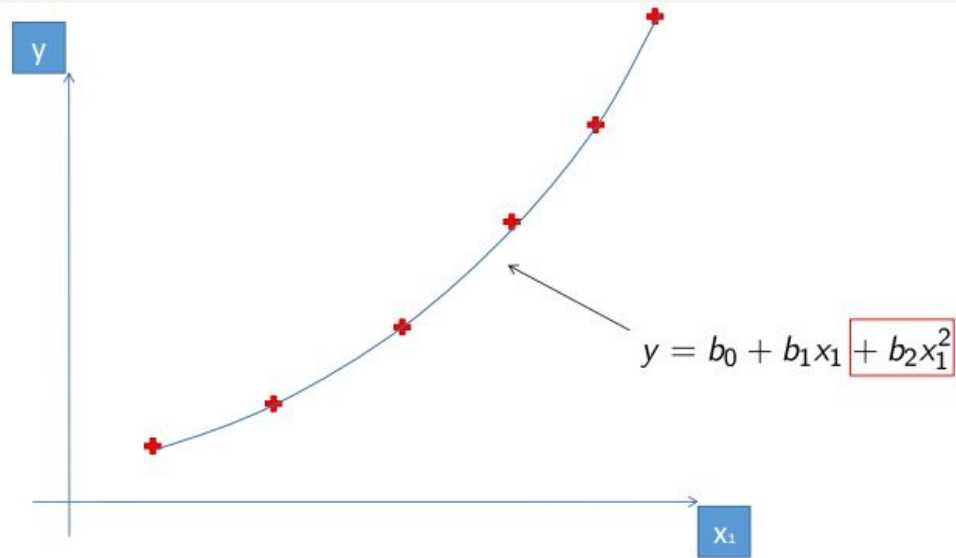
X₂ = temperatura

a₃ = +1mm de chuva gera +0.05t de batata

X₃ = qtd de chuva

Regressão Polinomial

$$y = b_0 + b_1x_1 + b_2x_1^2 + \dots + b_nx_1^n$$



Ex. Descrever como as doenças ou pandemias e epidemias se disseminam pelo território ou pela população.

Pq **Linear**?

Temos a mesma variável x_1 , mas ela está em potências diferentes.

Linear ou não-linear, se refere aos coeficientes e não às variáveis Independentes

$$y = b_0 + b_1x_1 + b_2x_1^2 + \dots + b_nx_1^n$$

```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
poly_reg = PolynomialFeatures(degree = 4)
X_poly = poly_reg.fit_transform(X_train)
regressor = LinearRegression()
regressor.fit(X_poly, y_train)
y_pred = regressor.predict(poly_reg.transform(X_test))
```

Outros tipos de Regressão

Modelos Regularizados

Regularização (L1, L2) = reduzir variância (overfit) adicionando parâmetros de controle (penalidade) → mudar a inclinação (slope) da função

$$Y = a + b * X1 + c * X2$$

Lasso (L1) => *LASSO Regularization Penalty* : $\lambda * |b| + \lambda * |c|$; $||w||$

Ridge (L2) => *Ridge Regularization Penalty* : $\lambda * b^2 + \lambda * c^2$; $||w||^2$