

# Data Science do ZERO

Capítulo 06 - Machine Learning  
**Regressão Linear**

# Regressão Linear

- Algoritmo utiliza aprendizado supervisionado.
- É uma técnica que consiste em uma equação linear que usa valores de entrada para prever valores de saída.
- Essa equação utiliza coeficientes que são aplicados para prever saídas.
- A regressão linear trabalha com dados numéricos apenas.
- Os coeficientes podem ser chamados de pesos.
- Os pesos são atualizados conforme a função que minimiza os erros.



# Regressão Linear

- Os dados de entradas são  $(x_1, x_2, x_3)$  e o valor de saída é “ $y$ ”.
- Teremos uma estrutura como a seguinte:
  - $x_1, x_2, x_3 = y$
- A representação acima diz que  $x_1, x_2$  e  $x_3$  são as **features** e  $y$  é a **classe**.
- Queremos uma função que estima o valor de  $Y$  para novos exemplos onde não temos esse valor.
- A regressão linear utiliza os pesos para aprender uma representação que se aproxime o máximo dos dados de treino.
- Para o exemplo anterior, para cada feature teríamos um peso associado. exemplo:
  - $p_0, p_1, p_2, p_3$
- O peso  $p_0$  seria usado para controlar a direção da reta.
- Esses pesos são aplicados aos dados para aprender uma representação dos dados de treino, exemplo:
  - $p_0 + p_1 * x_1 + p_2 * x_2 + p_3 * x_3 = y_i$
- O Resultado de saída é então comparado ao valor real de  $Y$ , essa diferença é chamada de erro de predição.

# Regressão Linear

## Exemplo

- Imagine que nosso objetivo é usar a regressão linear para prever o valor de fechamento da ação da Petrobras em um dia específico.
- Temos os seguintes dados e queremos prever o valor de fechamento do dia:
  - Abertura = 12,30
  - Máxima = 12,35
  - Mínima = 12,20
  - **Fechamento = ?**
- Os dados de abertura, máxima e mínima são os dados que temos, então aplicamos a regressão linear para tentar o prever o valor de fechamento.
- A regressão linear aplica os pesos nos dados, imagine que os valores dos pesos são:
  - $p_0 = 1$
  - $p_1 = 0,7$
  - $p_2 = 0,06$
  - $p_3 = 0,08$

# Regressão Linear

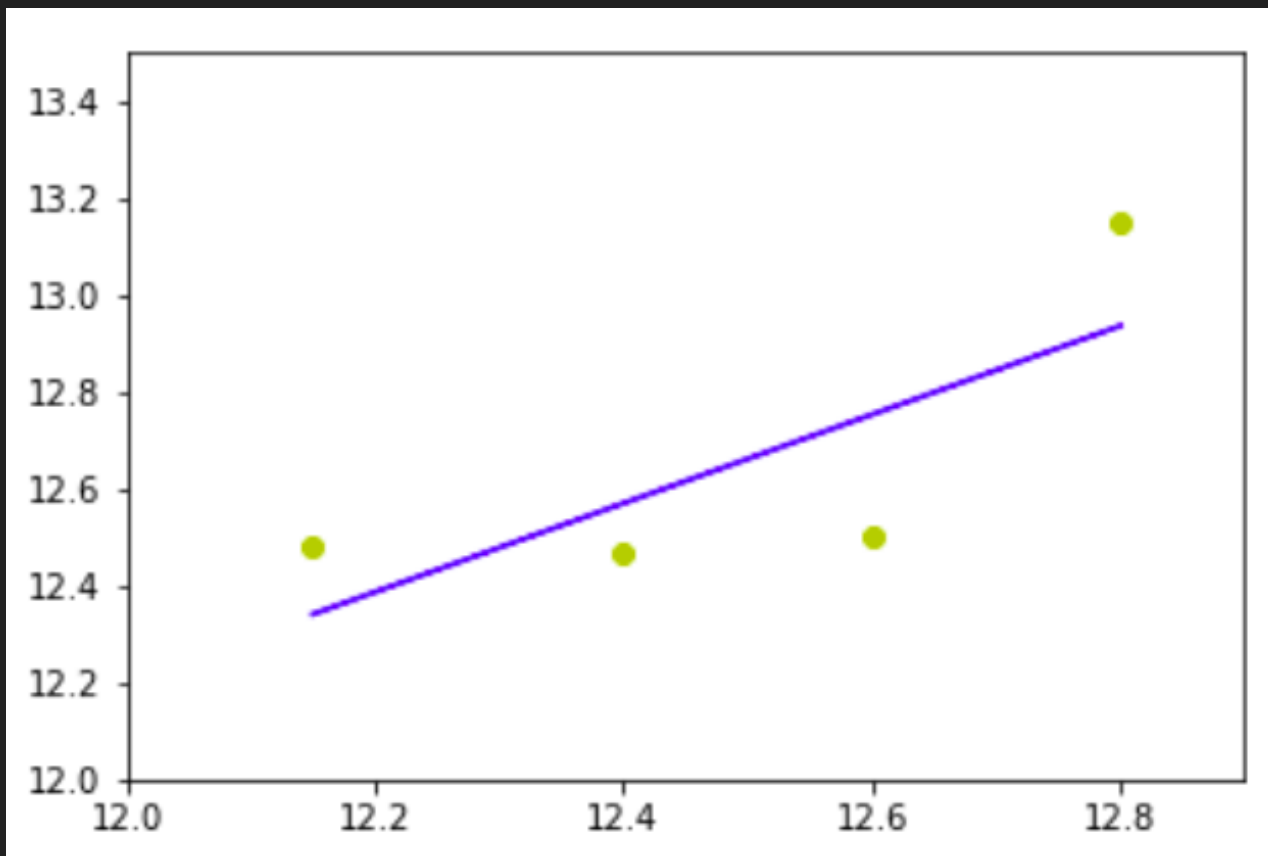
- Aplicando os valores a equação teríamos:

$$\begin{aligned}y &= p_0 + p_1 * x_1 + p_2 * x_2 + p_3 * x_3 \\1 + 0,7 * 12,30 + 0,06 * 12,35 + 0,08 * 12,20 \\10,84 + 0,74 + 0,97 &= \mathbf{12,55}\end{aligned}$$

- O resultado de predição foi **12,55**.
- Esse seria o valor de fechamento da ação predito pela regressão linear. O próximo passo é mensurar o erro de predição.
- O valor real do fechamento do dia for **12,33** temos que calcular o erro (entre o valor predito e o valor real) e **ajustar os pesos novamente**.
- Esse processo é feito durante todo o processo de treinamento do algoritmo.

# Regressão Linear

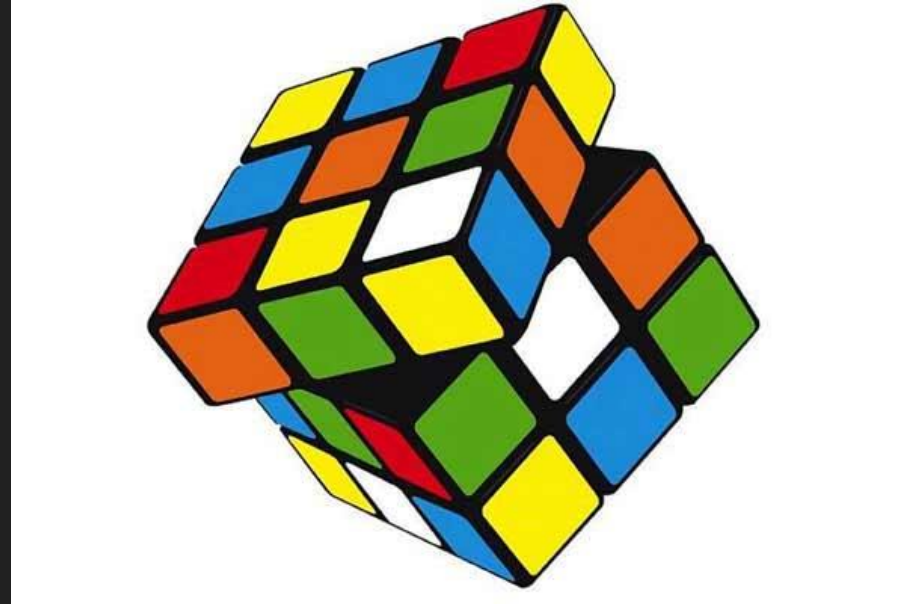
- O resultado da nossa regressão linear seria uma reta como na imagem a seguir.



# Regressão Linear

## Gradient Descent

- Algoritmo usado para **minimização do erro** dos pesos do modelo.
- Este utiliza o **erro médio quadrático** entre o valor predito e o valor real.
- Este algoritmo utiliza todos os dados de treinamento de forma **iterativa** até o menor erro possível.
- É preciso parametrizar o valor da **taxa de aprendizado**. Esse parâmetro controla o nível de aprendizado a cada iteração do algoritmo.



# Regressão Linear

## Gradient Descent

- Exemplo de como o Gradient Descent funciona:
  - Imagine os seguintes valores:
    - $p_0 = 1$  ,  $p_1 = 0.9$  ,  $x_1 = 12,30$
    - $Y = 12,33$  (valor real)
- Nossa equação agora seria:
  - Agora vamos jogar isso na equação acima:
    - $Y_i = 1 + 0.9 * 12.30$
    - $Y_i = 12,07$
  - O valor predito foi 12,07, agora vamos calcular o erro:
    - $\text{erro} = 12,07 - 12,33$
    - $\text{erro} = -0,26$
- Com o erro de predição, os pesos são atualizados para tentar minimizar novos erros..



# Regressão Linear

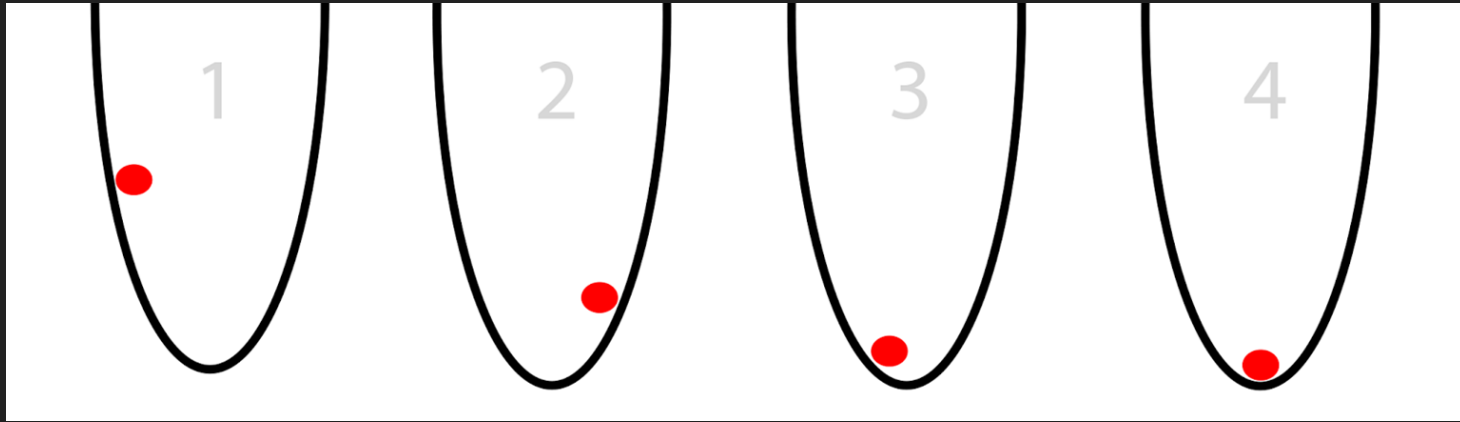
## Gradient Descent - Atualização dos pesos:

- Para a taxa de aprendizado vamos chamar de **alpha**.
- O objetivo é calcular o novo valor dos pesos  $p_0$  e  $p_1$ .
- Para  $\alpha$  no valor de 0.01 temos:
  - $p_0 = p_0 - \alpha * \text{erro}$
  - Calculando:
    - $p_0 = 1 - 0.01 * -0,26$
    - $p_0 = -0.25$
- Com o novo valor de  $p_0$  vamos agora calcular o valor de  $p_1$ .
- A única diferença é que agora o valor de entrada é usado na equação, pois, o valor de  $p_1$  deve ter influência no valor da feature associada a ele.
  - $p_1 = p_1 - \alpha * \text{error} * x_1$
  - $p_1 = 0.9 - 0,01 * -0,26 * 12,30$ 
    - $p_1 = 0,93$

# Regressão Linear

## Gradient Descent - Atualização dos pesos:

- O Gradient Descent repete esse processo a cada instância do treino até que os pesos se ajustem com o mínimo de erro.
- A cada ciclo completo damos o nome de **época**



- A imagem acima ilustra o processo, é possível observar que após várias épocas se consegue chegar no ponto mínimo de erro.

Hands on!