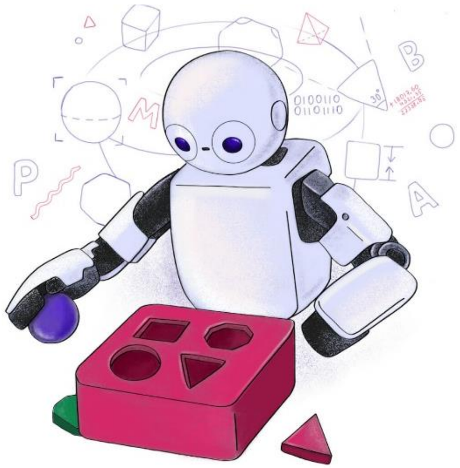
TP557 - Tópicos avançados em IoT e Machine Learning:
Medindo a precisão de um modelo de ML



Inatel

Felipe Augusto Pereira de Figueiredo
felipe.figueiredo@inatel.br

O que vamos ver?

- Neste tópico vamos ver como medir o desempenho de um modelo de aprendizado de máquina ao longo do seu processo de aprendizagem.
- Para isso, como já discutido brevemente antes, usaremos uma função chamada de função de erro ou de perda.
- Idealmente, o processo de treinamento tem como objetivo minimizar o erro e, consequentemente, aumentar a precisão do modelo.
- Além disso, veremos em breve diferentes estratégias para minimizar o erro.
- Porém, primeiro, vamos aprender como calcular o erro.

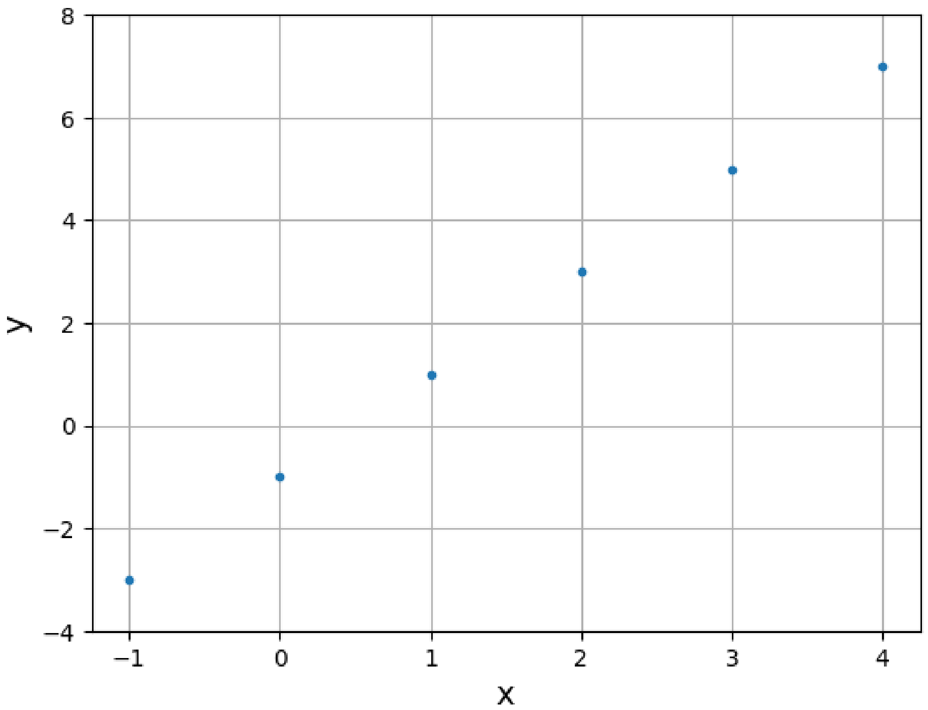
Mapeando $\boldsymbol{x}$ em $\boldsymbol{y}$

$$\boldsymbol{x} = \{-1, 0, 1, 2, 3, 4\}$$
$$\boldsymbol{y} = \{-3, -1, 1, 3, 5, 7\}$$

- Considerem esses dois conjuntos de números.
- Qual a relação entre os dois conjuntos?
  - Ou seja, qual é a função que mapeia os valores de $\boldsymbol{x}$ em $\boldsymbol{y}$?
- Nós sabemos que $\boldsymbol{y}$ é uma função de $\boldsymbol{x}$, nós só não sabemos que função é essa.
- Que tal plotarmos esse pontos?

Mapeando $\boldsymbol{x}$ em $\boldsymbol{y}$: Função hipótese

$$\boldsymbol{x} = \{-1, 0, 1, 2, 3, 4\}$$
$$\boldsymbol{y} = \{-3, -1, 1, 3, 5, 7\}$$



- Ao plotarmos os pontos, percebemos que existe uma relação linear entre eles.
- Podemos criar a ***hipótese*** de que uma reta explica esse mapeamento.
- Portanto, usamos a função de uma reta para definir o mapeamento:
$$y = a_0 + a_1 x.$$
- Agora precisamos encontrar os valores dos parâmetros (ou pesos) a_0 e a_1.

Suposição e Predição

$$\hat{y} = -1 + 3x$$

$$\boldsymbol{x} = \{-1, 0, 1, 2, 3, 4\}$$

Predições feitas pela função hipótese atual:

$$\hat{\boldsymbol{y}} = \{-4, -1, 2, 5, 8, 11\}$$

- Vamos começar atribuindo alguns valores aleatórios para os pesos, ou seja, vamos fazer uma suposição sobre os valores.
- Como medimos se essa função hipótese é boa ou ruim?
  - Usamos os valores de x para obter o mapeamento (i.e., predição) feito pela função e comparamos com os valores de saída esperados, $\boldsymbol{y}$.

O quão boa é a função hipótese?

$$\hat{y} = -1 + 3x$$

$$\boldsymbol{x} = \{-1, 0, 1, 2, 3, 4\}$$

Predições feita pela função hipótese atual:

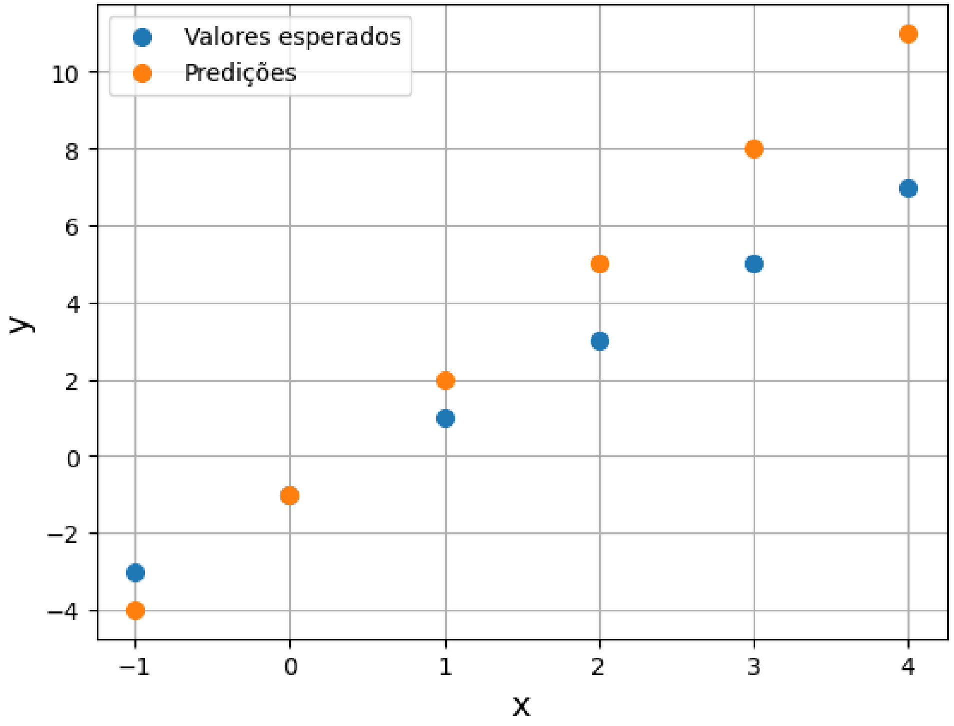
$$\hat{\boldsymbol{y}} = \{-4, -1, 2, 5, 8, 11\}$$

Valores de saída esperados:

$$\boldsymbol{y} = \{-3, -1, 1, 3, 5, 7\}$$

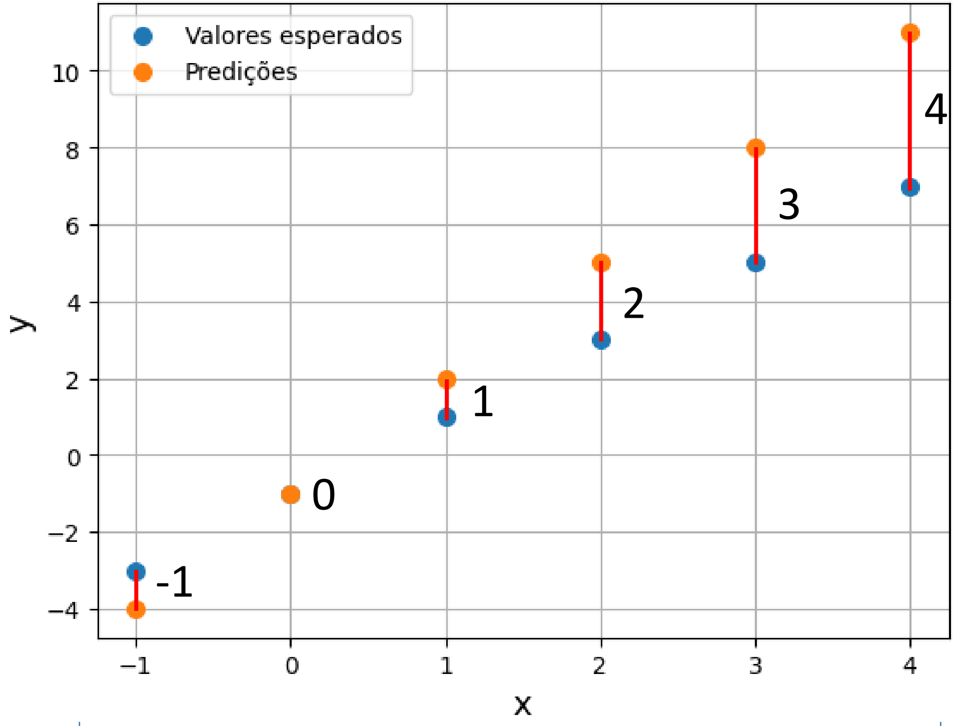
- Vemos que os valores preditos e esperados não são os mesmos.
- Os três primeiros valores até são próximos, mas os últimos já estão mais distantes.
- Existe uma maneira de formalizarmos um cálculo do quão bom ou ruim essa função e suas predições são?

Vamos medir as distâncias!



- Talvez nós possamos definir uma métrica de desempenho plotando os valores esperados e preditos.
- Na figura ao lado, temos os pontos preditos e esperados para cada valor de $\boldsymbol{x}$.

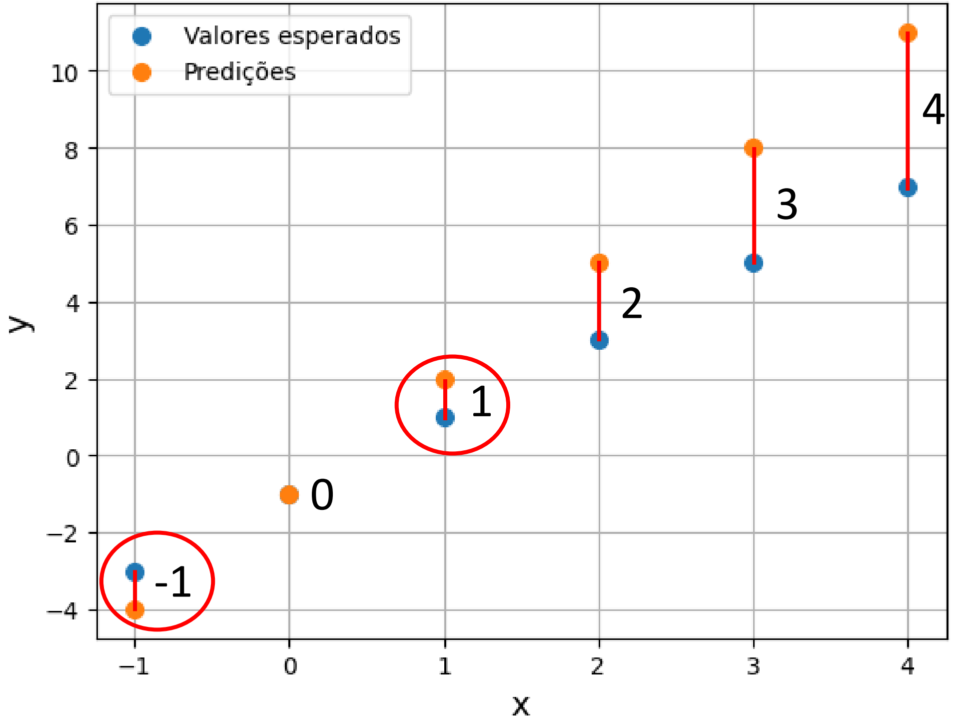
Vamos medir as distâncias!



$$\text{diff} = \hat{y} - y$$

- Podemos traçar uma reta entre cada ponto e, pelo comprimento dessas retas (ou ***diferença*** entre os pontos), descobrir se a função hipótese é boa ou não.
- O comprimento de cada reta é mostrado ao lado delas.
- Talvez nós possamos calcular a média dos comprimentos para obter o ***erro médio*** causado pela função hipótese atual.

Temos um problema!

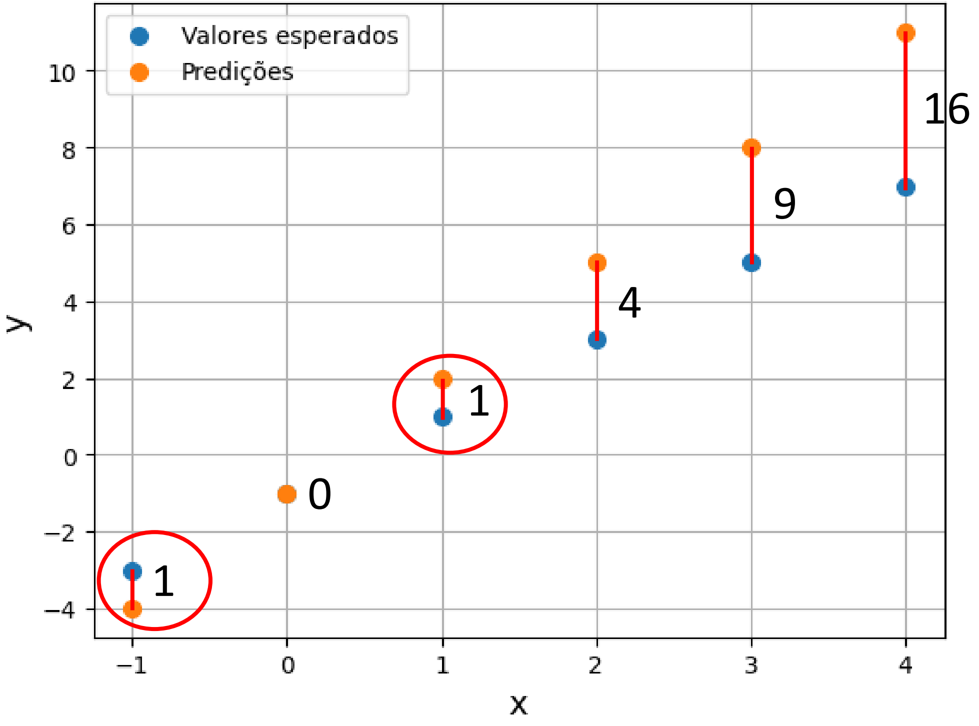


$$\text{erro médio} = \frac{\text{diff}}{6}$$

$$\text{erro médio} = \frac{-1 + 0 + 1 + 2 + 3 + 4}{6} = 1.5$$

- Ao somarmos os comprimentos, mesmo as predições para os dois pontos estando erradas, seus erros se cancelariam, afetando a medida de desempenho.
- Isso poderia sugerir que 3 de 6 predições estão corretas, o que sabemos não ser verdade.
- O que podemos fazer para resolver esse problema?

Quadrado dos comprimentos



$$\text{erro quadrático médio (EQM)} = \frac{\text{diff}^2}{6}$$

$$\text{EQM} = \frac{1 + 0 + 1 + 4 + 9 + 16}{6} = 5.17$$

- Podemos elevar ao quadrado todos os comprimentos, fazendo com que todos eles sejam positivos e não se cancelem mais.
- Isso não afeta nossa métrica, pois aplicamos a todos os comprimentos.
- Usando essa métrica de ***erro médio***, vemos que nossa função hipótese não é boa, pois o valor ainda está longe de zero, que é o menor valor possível e nosso objetivo final.
- O que devemos fazer?

Otimizando a suposição

$$\hat{y} = -2 + 2x$$

$$\boldsymbol{x} = \{-1, 0, 1, 2, 3, 4\}$$

Predições feitas pela função hipótese atual:

$$\hat{\boldsymbol{y}} = \{-4, -2, 0, 2, 4, 6\}$$

Valores de saída esperados:

$$\boldsymbol{y} = \{-3, -1, 1, 3, 5, 7\}$$

Diferença entre predições e valores esperados:

$$\text{diff}^2 = \{1, 1, 1, 1, 1, 1\}$$

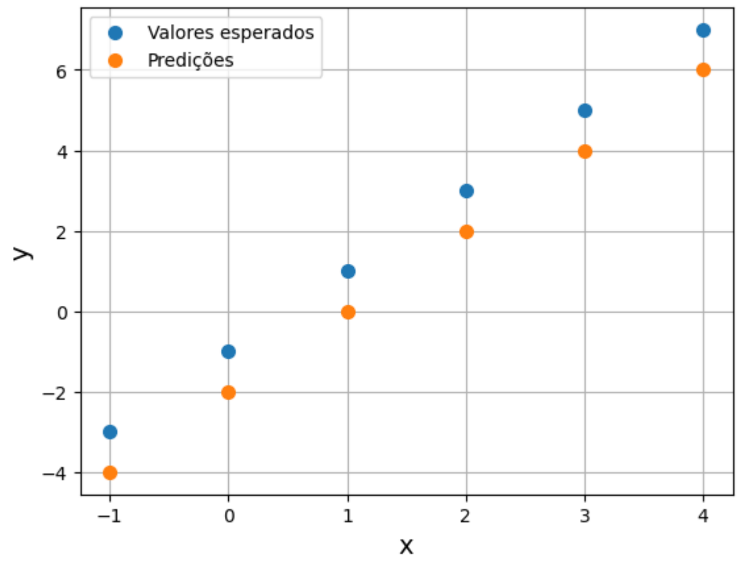
- Vamos supor outros valores para a_0 e a_1 e fazer predições com esta nova função hipótese.

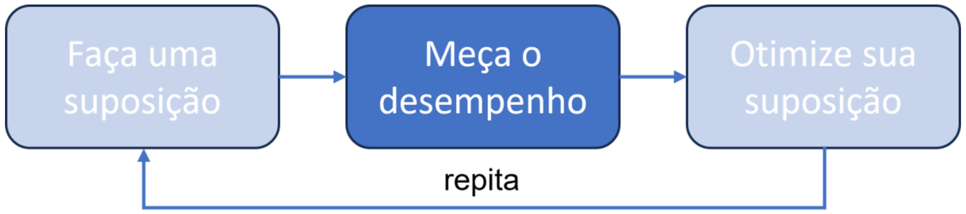
O quão boa é a nova função hipótese?

$$\hat{y} = -2 + 2x$$

$$\boldsymbol{x} = \{-1, 0, 1, 2, 3, 4\}$$

$$\mathrm{EQM} = \frac{1+1+1+1+1+1}{6} = 1.0$$





- Já temos uma função melhor, pois o erro caiu de 5.17 para 1.0.
- Estamos indo na direção correta!
- Comparando as predições com os valores esperados, notamos que as retas são paralelas, sendo a única diferença um ***deslocamento***.
- O descolamento pode ser alterado através de a_0 (coeficiente linear).

Nova suposição

$$\hat{y} = -1 + 2x$$

$$\boldsymbol{x} = \{-1, 0, 1, 2, 3, 4\}$$

Predições feitas pela função hipótese atual:

$$\hat{\boldsymbol{y}} = \{-3, -1, 1, 3, 5, 7\}$$

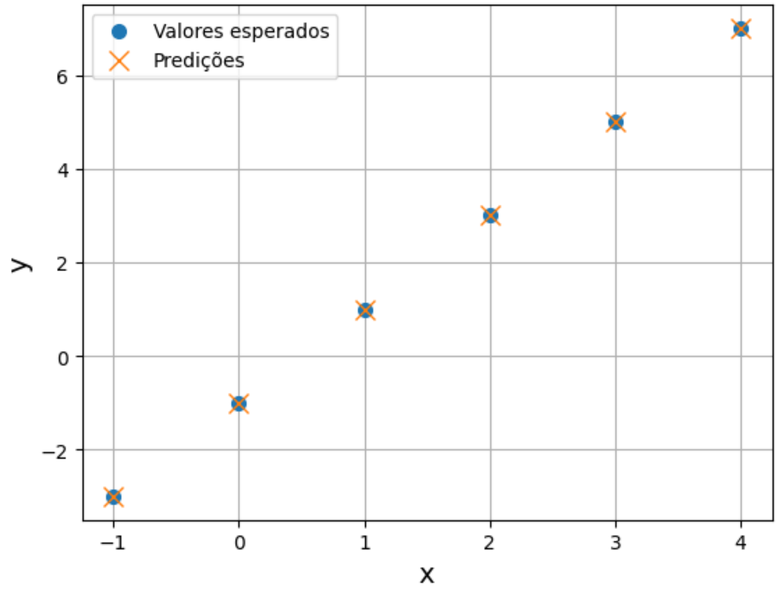
Valores de saída esperados:

$$\boldsymbol{y} = \{-3, -1, 1, 3, 5, 7\}$$

Diferença entre predições e valores esperados:

$$\text{diff}^2 = \{0, 0, 0, 0, 0, 0\}$$

$$\text{EQM} = \frac{0+0+0+0+0+0}{6} = 0.0$$



- Vamos diminuir o valor de a_0 e fazer novas predições.
- Bingo! Nossa nova suposição mapeia perfeitamente x em y, resultando em um EQM igual a 0.

Outras medidas de erro

$$\mathrm{MSE} = \frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)^2$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)^2}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{n=1}^{N}|\hat{y}_n - y_n|$$

Usadas em problemas de aproximação de curvas

$$\underbrace{\text{cross entropy} = -\frac{1}{N}\sum_{n=1}^{N} y_n \log_e(\hat{y}_n)}_{\text{Usada em problemas de classificação}}$$

- Além do EQM (Mean Squared Error – MSE), existem diversas outras métricas de erro que podemos usar
  - Raíz do Erro Quadrático Médio (Root Mean Squared Error – RMSE)
  - Erro Absoluto Média (Mean Absolute Error – MAE);
  - Entropia Cruzada (Cross-Entropia);
  - Etc.

Mean Squared Error – MSE

$$\text{MSE} = \frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)^2$$

- É a função de erro mais usada em problemas de regressão.
- Calcula a média do ***quadrado da diferença*** entre os valores preditos e esperados.
- Assim, ele ***penaliza mais erros maiores*** (devido ao quadrado), o que pode ser útil se desejarmos que o modelo seja ***mais sensível a esses erros***.
- Porém, como a diferença é elevada ao quadrado, ele ***pode amplificar*** a influência de ***outliers***.
- Por usar o quadrado da diferença, a métrica não fica na mesma escala dos dados originais.
  - Por exemplo, se estivermos predizendo o preço de casas em dólares, os erros estarão em dólares^2.

Root Mean Squared Error – RMSE

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)^2}$$

- Calcula a raiz quadrada do MSE, retornando a métrica à mesma escala dos dados originais.
- Isso é útil para interpretar o erro em unidades da variável alvo original (i.e., os rótulos), o que pode facilitar a compreensão do impacto do erro.
- Assim como o MSE, também penaliza erros maiores.

Mean Absolute Error – MAE

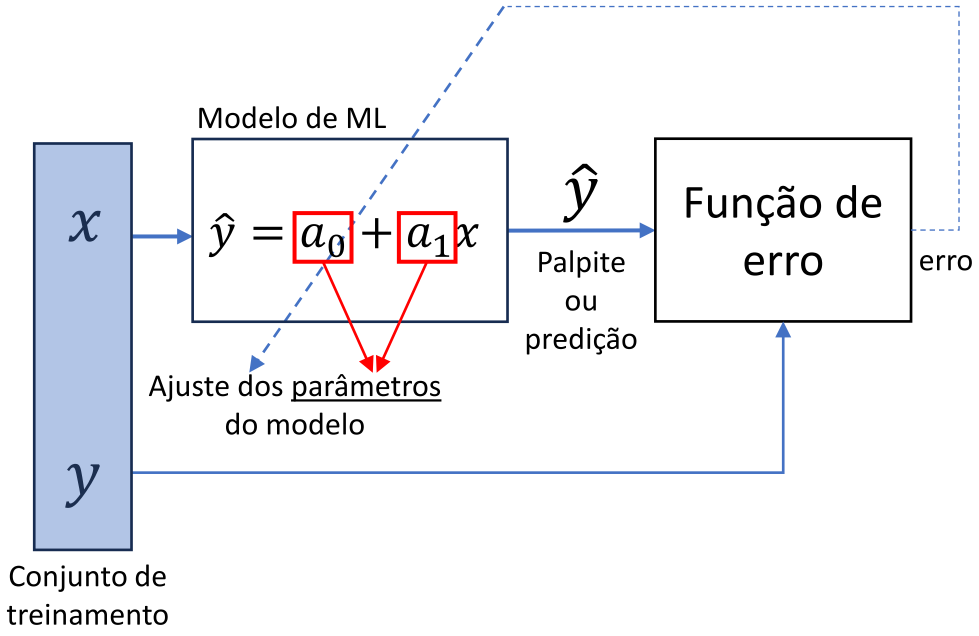
$$\text{MAE} = \frac{1}{N}\sum_{n=1}^{N}|\hat{y}_n - y_n|$$

- Calcula a ***média das diferenças absolutas*** entre as predições do modelo e os valores esperados.
  - Ou seja, penaliza os erros de maneira uniforme.
- Ao contrário do MSE e RMSE, não eleva os erros ao quadrado, o que o torna menos sensível a *outliers*.
- Representa os erros na mesma escala dos dados originais, facilitando a interpretação.
- Boa opção quando desejamos um erro mais uniforme em todas as predições e desejamos minimizar a influência de outliers.

Qual devo usar?

O recomendável é ***experimentar diferentes funções de erro*** durante a ***fase de desenvolvimento*** para determinar qual delas se ***alinha*** melhor com os ***objetivos e as características*** específicas do seu ***problema de regressão***.

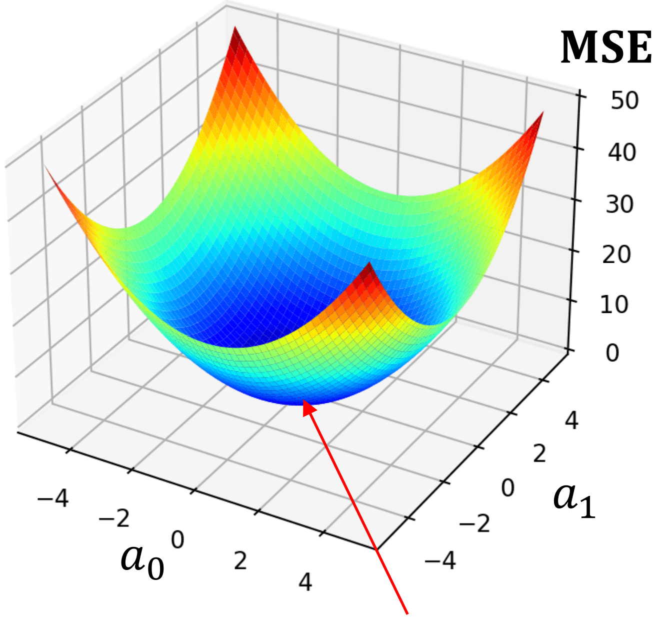
Regressão ou aproximação de funções



- O problema visto neste tópico é conhecido como ***regressão*** ou ***aproximação de funções***.
- Nele, a função com formato de reta, que definimos como ***função hipótese***, é o ***modelo de ML*** e o ***objetivo da regressão é encontrar os parâmetros (ou pesos) que minimizam a função de erro***.

OBS.: Nesse exemplo, conseguimos visualizar os dados e, facilmente, identificar a equação (ou formato) da função hipótese, mas em outros casos, além dos pesos, temos que descobrir o formato da função.

Ponto de mínimo ou solução ótima



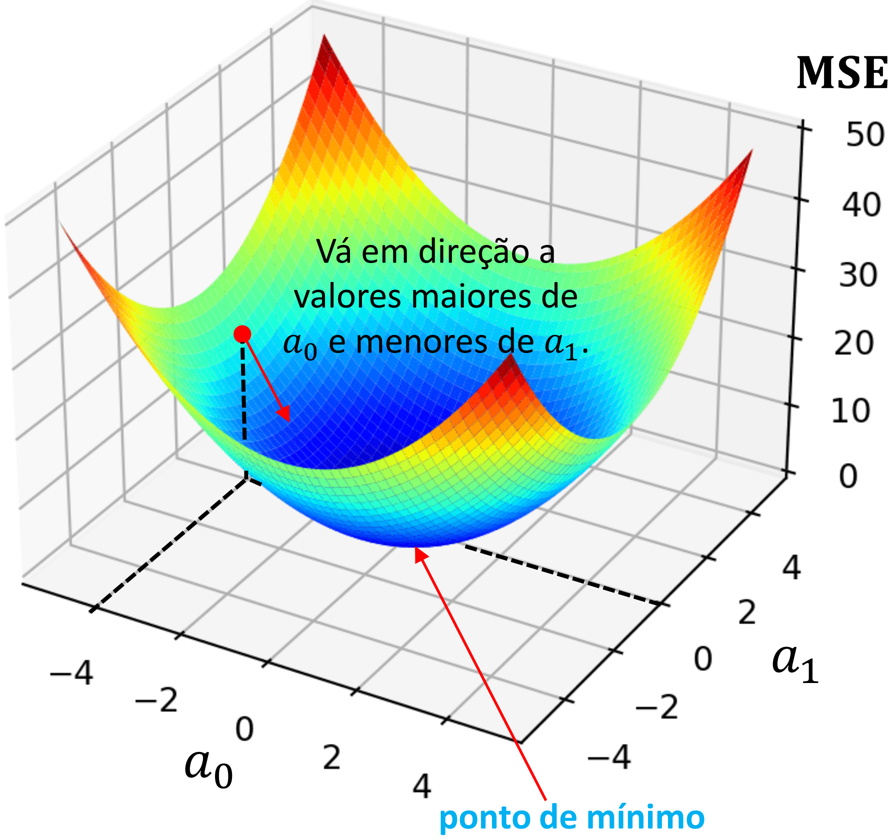
$$\text{MSE} = \frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - y_n)^2$$

$$= \frac{1}{N}\sum_{n=1}^{N}(\boxed{a_0} + \boxed{a_1}x_n - y_n)^2$$

?

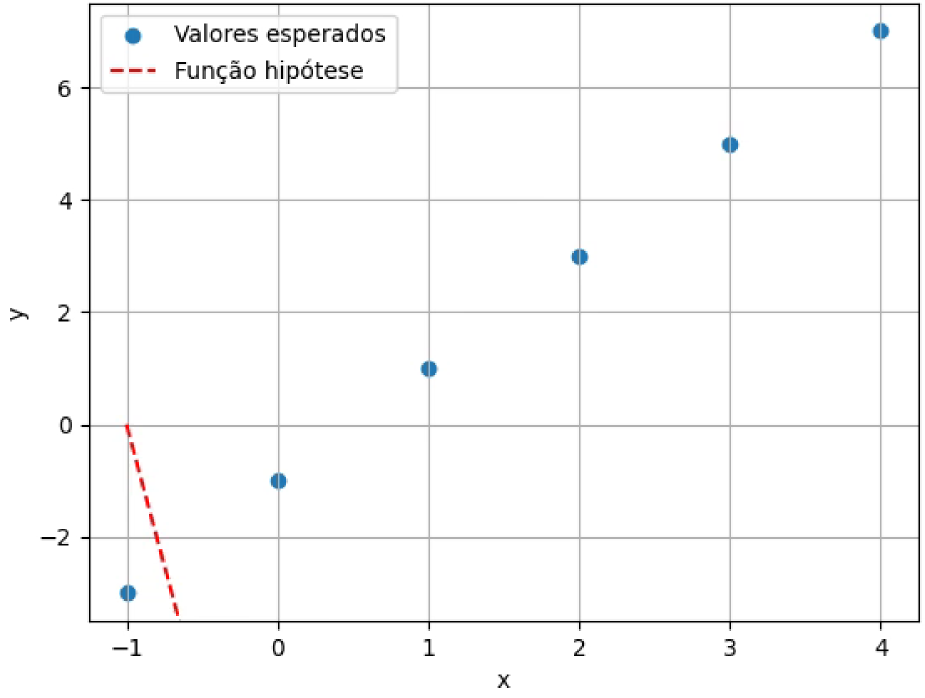
- Se notarmos que a ***função de erro*** é ***função dos pesos*** do modelo, nós podemos dizer que o ***objetivo da regressão é encontrar o ponto de mínimo***, especificado pelos pesos, da função, ou seja, ***seu ponto mais baixo***.
- Esse é um ***problema de minimização***.
  - ***Qual é o conjunto de pesos que resulta no menor erro possível?***

Otimização (treinamento) do modelo



- O processo que tem como objetivo minimizar o erro é chamado de ***otimização*** e, no contexto de ML, é conhecido como ***treinamento do modelo***.
- O treinamento/atualização do modelo se baseia em informação obtida a partir da função de erro.
- Essa informação ***aponta para a direção do ponto de mínimo*** ***(i.e., conjunto de pesos ótimo)***.

Gradiente descendente



- Como veremos em breve, de posse dessa informação, o algoritmo de otimização, conhecido como ***gradiente descendente (GD)***, consegue ***caminhar iterativamente*** até o ***ponto de mínimo*** (i.e., conjunto de pesos ótimo).
- A figura ao lado ilustra o processo iterativo de otimização realizado pelo GD.
  - A partir de um ponto inicial (i.e., uma suposição aleatória de pesos), o algoritmo extrai a informação de direção do ponto de mínimo a partir da função de erro e a usa para atualizar o ponto atual (i.e., otimizar a suposição atual).
  - Em geral, o novo ponto (i.e., conjunto de pesos atualizado) resulta em um erro menor do que o anterior.
  - Esse processo é repetido a partir do novo ponto até que o erro seja minimizado.

Próximos passos

- Nesse tópico nós vimos o que é o erro (ou perda) e definimos uma forma de medi-lo.
- Agora, o próximo passo envolverá a definição de um processo que utilize a informação do erro para gerar o próximo palpite (ou suposição) de forma que ele seja melhor do que o atual.
- Esse processo é chamado de ***otimização*** e será abordado em breve.
- Mas primeiro, vamos dar uma olhada em um exemplo que nos dará uma ideia sobre como funciona esse processo de otimização.

jupyter CO Explorando a função de erro

Atividades

- Quiz: “***TP557 - Medindo a precisão de um modelo de ML***”.
- Exercício: Encontre os pesos da função hipótese.

Perguntas?

Obrigado!


Faça uma suposição
Meça o desempenho
Otimize sua suposição
repita


Faça uma suposição
Meça o desempenho
Otimize sua suposição
repita


Faça uma suposição
Meça o desempenho
Otimize sua suposição
repita