

Inteligência Artificial e Machine Learning II

Aula 2

Professor: Alex Pereira

Preparação dos dados para construção de Modelos de ML

Garbage-in, garbage-out

A qualidade da saída é determinada pela qualidade da entrada

Codificação de Atributos (Feature Encoding)

- Sinônimo: One-hot encoding
- Transformar atributos categóricos em numéricos

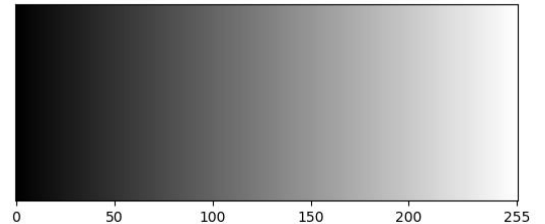
cpf	nome	salario	departamento
98654345821	Antônio Freitas	15000	Vendas
78621232185	Alessandra Rezende	10000	Produção
89632189251	Amanda Silva	8000	RH
21326478932	Anderson Pereira	12000	Produção



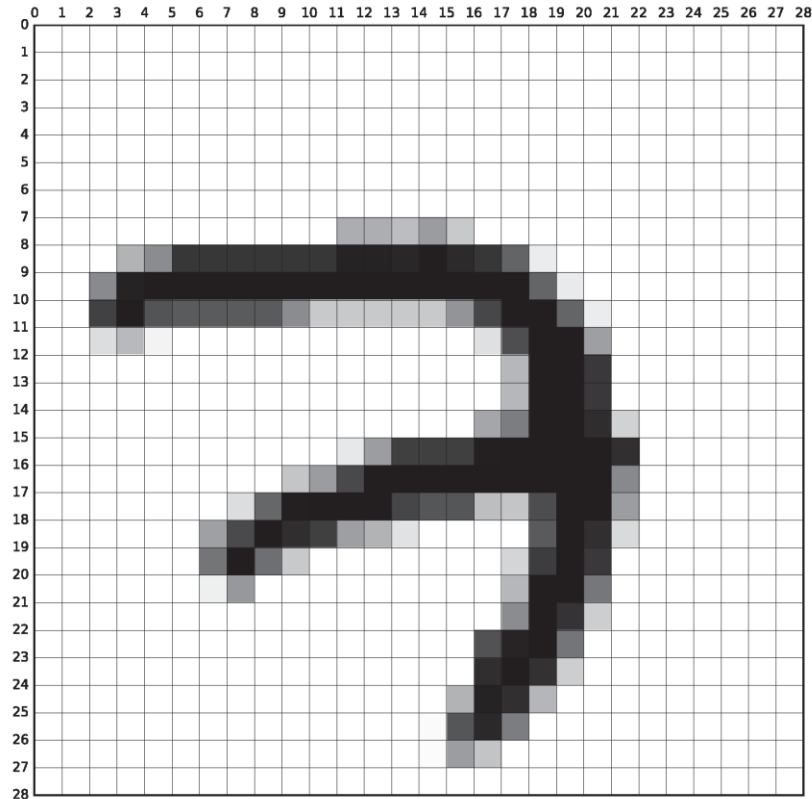
cpf	nome	salario	Vendas	Produção	RH
98654345821	Antônio Freitas	15000	1	0	0
78621232185	Alessandra Rezende	10000	0	1	0
89632189251	Amanda Silva	8000	0	0	1
21326478932	Anderson Pereira	12000	0	1	0

Dataset MNIST

- Base de dados de dígitos manuscritos
 - A MNIST contém 60.000 imagens de treino e 10.000 imagens de teste de dígitos manuscritos.
 - O conjunto de dados inclui imagens em escala de cinzentos de 28x28 pixéis.
 - As imagens são normalizadas para caberem numa caixa delimitadora de 28x28 pixels
 - ✓ níveis de escala de cinza (0 a 255)
- Em 1989, Yann LeCun (Meta)
 - Aplicou redes neurais e backpropagation neste problema
 - ✓ Que mais tarde deu origem ao MNIST.



Dataset MNIST (Modified National Institute of Standards and Technology)



(a) MNIST sample belonging to the digit '7'.

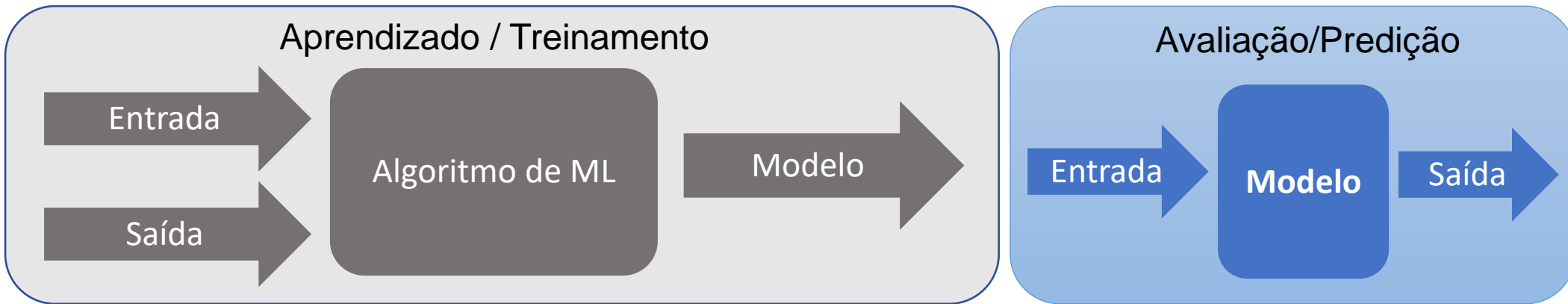


(b) 100 samples from the MNIST training set.

Rede Neural de Classificação

Do dataset MNIST

Aprendizado Supervisionado



- Treinar

- construir um modelo de predição que considere o efeito/contribuição de cada exemplo de treinamento

- ✓ Investimento

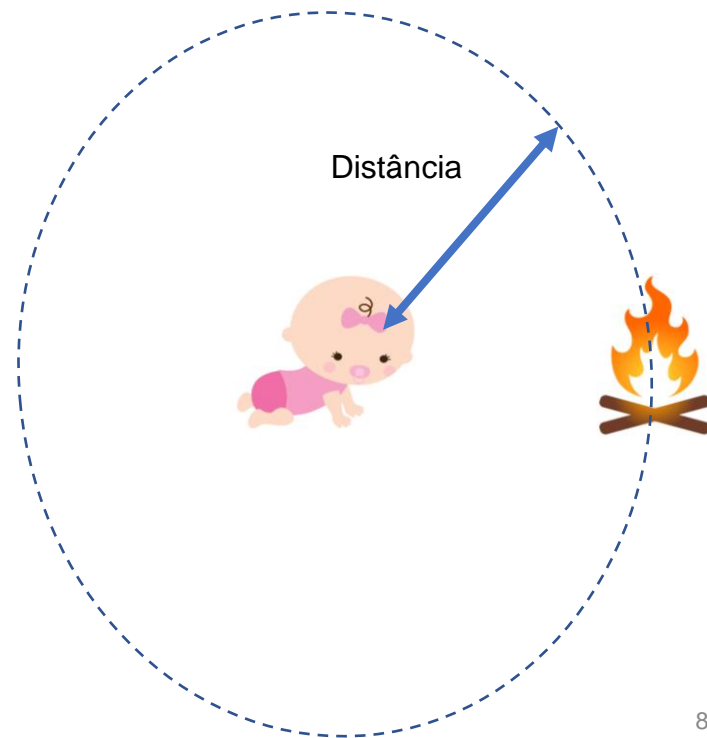
- Usa-se o modelo treinado para aplicar em situações de decisão

- Ou seja, a etapa de predição

- ✓ Retorno sobre o investimento

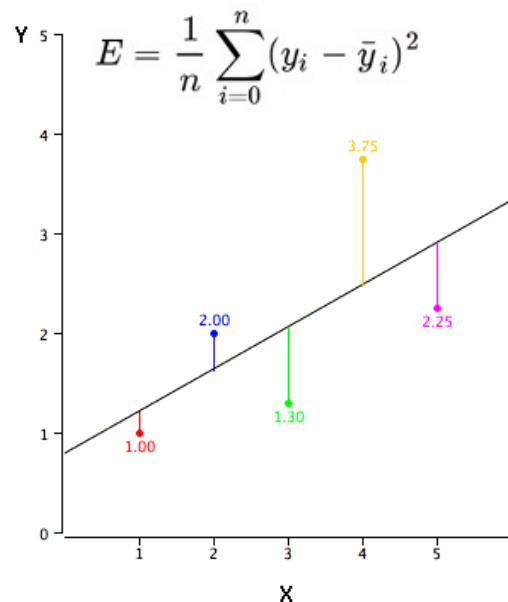
Loss/cost Function (Função de Custo): Analogia

- Loss Function
 - Seria a capacidade do bebê ter uma estimativa da sua sensação de conforto baseado na sua distância até a fogueira.
- Com o auxílio desta habilidade
 - Ele aprenderia qual distância ao fogo
 - ✓ Ihe deixa mais confortável
 - Longe demais fica frio, se perto de mais, quente
 - Aprendizado iterativo
- O ponto mínimo da função custo
 - Minimiza o desconforto
 - ✓ Maximiza o conforto do bebê.
- Loss Function => Descrever fenômenos
 - quantitativamente

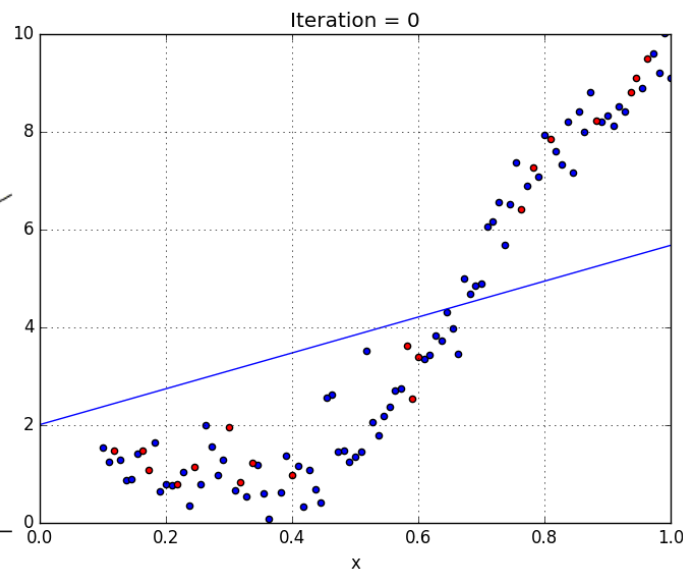


Loss/cost Function (Função de Custo)

- Tendo uma função de custo/perda
 - Que nos dê um feedback de o quão longe estamos do nosso objetivo
 - ✓ O próximo passo é encontrar os valores dos atributos/features
 - Que minimizam a função custo
- Regressão Linear
 - Ex.: Sorvete vs Temperatura
 - Função custo
 - ✓ Soma do quadrado dos erros
 - Melhor reta é a que
 - ✓ Minimiza a função custo
- Como encontrar a
 - melhor reta ?



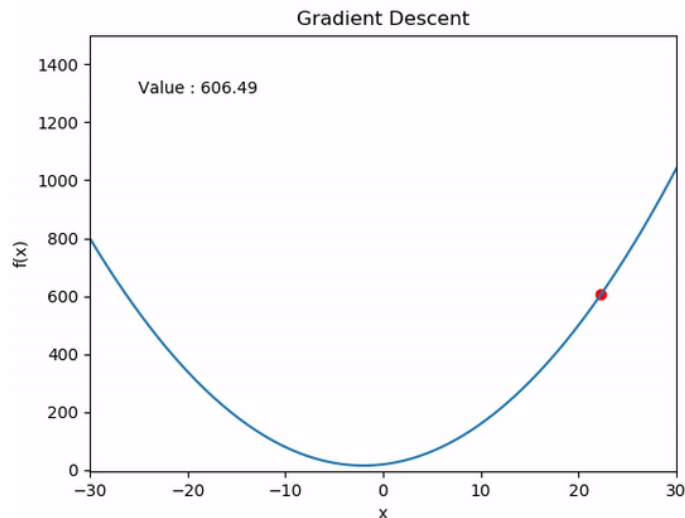
[Fonte da Imagem](#)



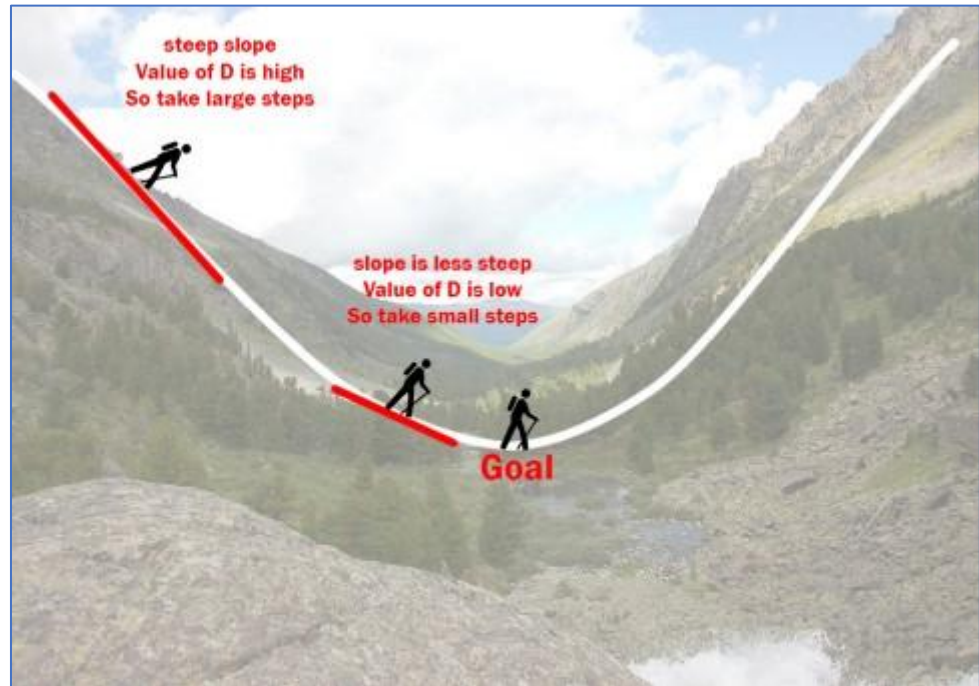
[Fonte da Imagem](#)

Gradient Descent (Gradiente Descendente)

- É um algoritmo que percorre uma função
 - Na direção mais íngreme da descida
 - ✓ Até encontrar seu mínimo
- Exemplo de Regressão Linear
 - Loss Function



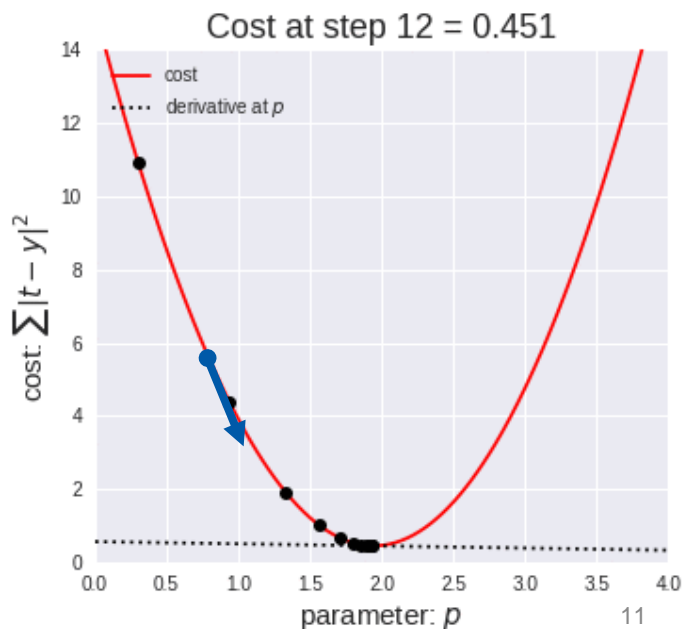
[Fonte da Imagem](#)



[Fonte da Imagem](#)

Gradient Descent (Gradiente Descendente)

- Não precisa conhecer o valor da função em todos os seus pontos
 - Basta escolher um ponto de partida e descobrir a direção descendente
 - ✓ A direção é indicada pelo Vetor Gradiente (tangente no ponto analisado)
 - A cada iteração
 - ✓ Dá-se um pequeno passo na direção
 - contrária ao vetor gradiente
 - e o mede novamente até encontrar o mínimo
 - tamanho do vetor gradiente é zero, ou próximo a zero
 - ou um número fixo de iterações.
- O cálculo que encontra a direção descendente
 - é a derivada
 - ✓ por isso há um pré-requisito da função custo
 - ser derivável



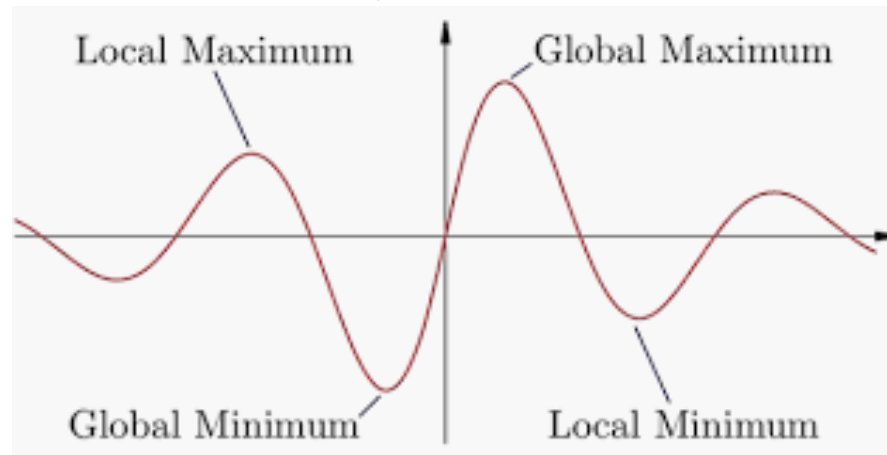
Gradient Descent (Gradiente Descendente)

- Existem funções de erro (loss functions) mais complexas

[Fonte da Imagem](#)

- Funções não puramente convexas

- ✓ Com mínimos locais e global
 - O treinamento pode convergir para
 - um mínimo local



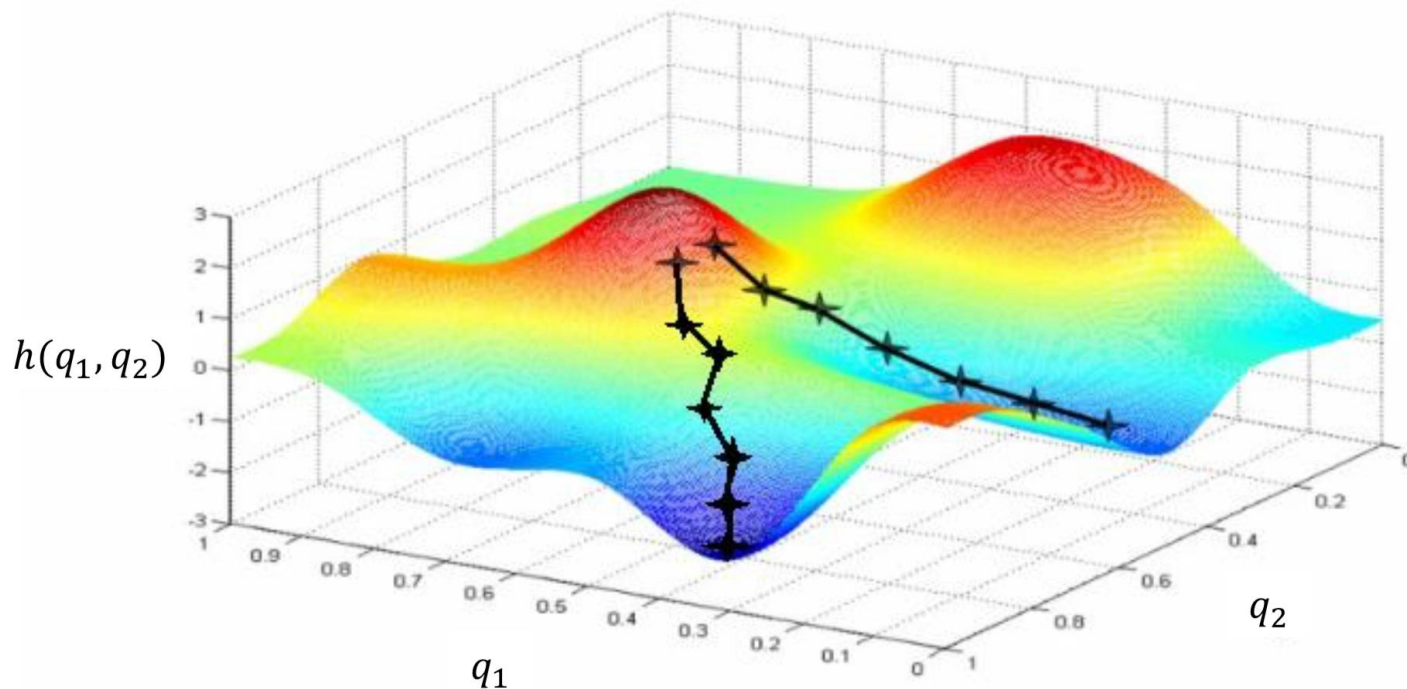
- Não visualizáveis.

- ✓ Calcular todos os pontos de uma função pode ser computacionalmente custoso
 - Melhor calcular somente nos pontos do caminho até o mínimo
 - Acostume-se com a existência de modelos mais abstratos
 - Onde o treinamento converge para um mínimo sem um auxílio gráfico.

- [Vídeo](#) ilustrativo

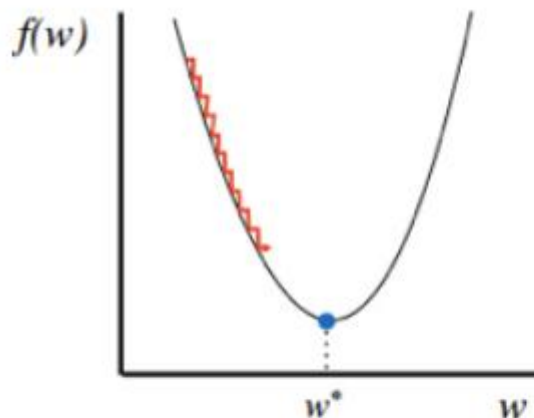
Gradient Descent (Gradiente Descendente)

- Exemplo de função não convexa
 - Pode convergir para um mínimo local
 - ✓ Dependendo do ponto de partida

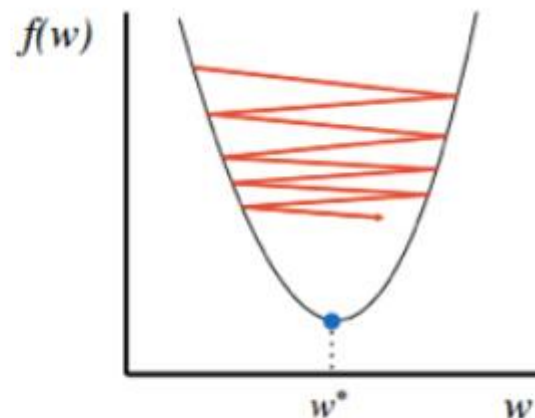


Gradient Descent (Gradiente Descendente)

- Learning Rate (taxa de aprendizado)
 - É o tamanho do passo na direção contrária ao vetor gradiente
 - ✓ Simulador de [Learning Rate](#) do Google



Muito pequeno:
converge lentamente



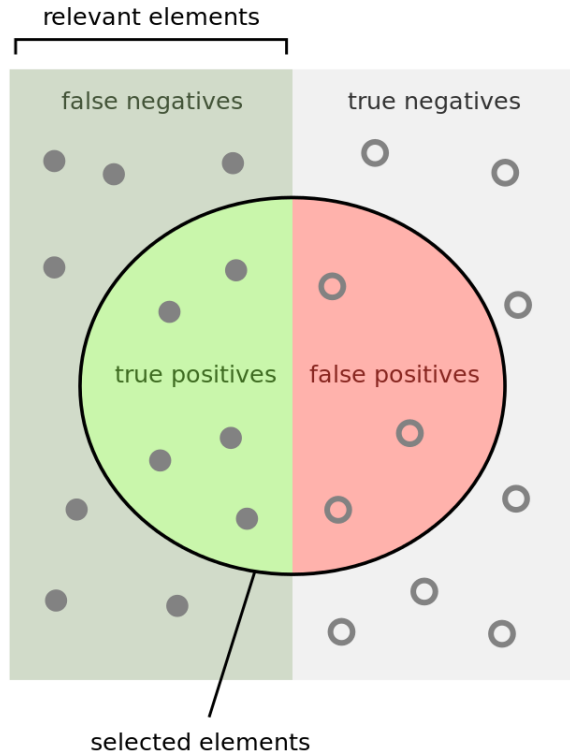
Muito grande: ultrapassa
ou até diverge

Acurácia dos Modelos: Matrix de Confusão

- TP: Classe predita foi Positivo e o valor Real também;
- FP: Classe predita foi Positivo mas o valor Real Negativo;
- FN: Classe predita foi Negativo mas o Valor Real Positivo; e
- TN: Classe predita foi Negativo e o Valor Real também.

		Predição	
		Positivo	Negativo
Real (Gabarito)	Positivo	TP=167	FN=15
	Negativo	FP=37	TN=355

Métricas de Acurácia



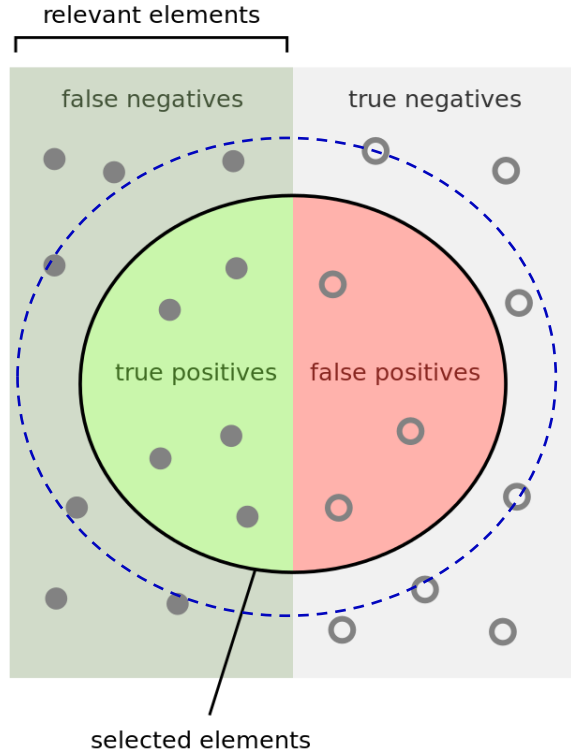
Precisão

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

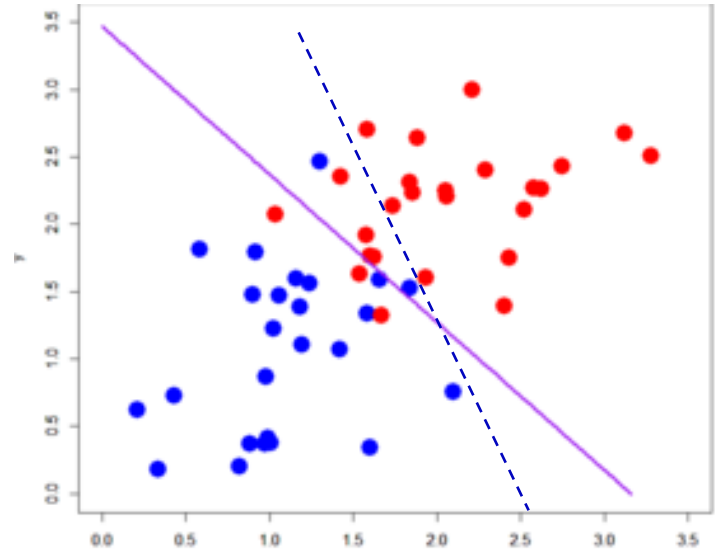
$$P = \frac{TP}{TP + FP}$$

Métricas de Acurácia

- Trade-off: ↑ Recall (Sensibilidade) e Precisão ↓



$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$
$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

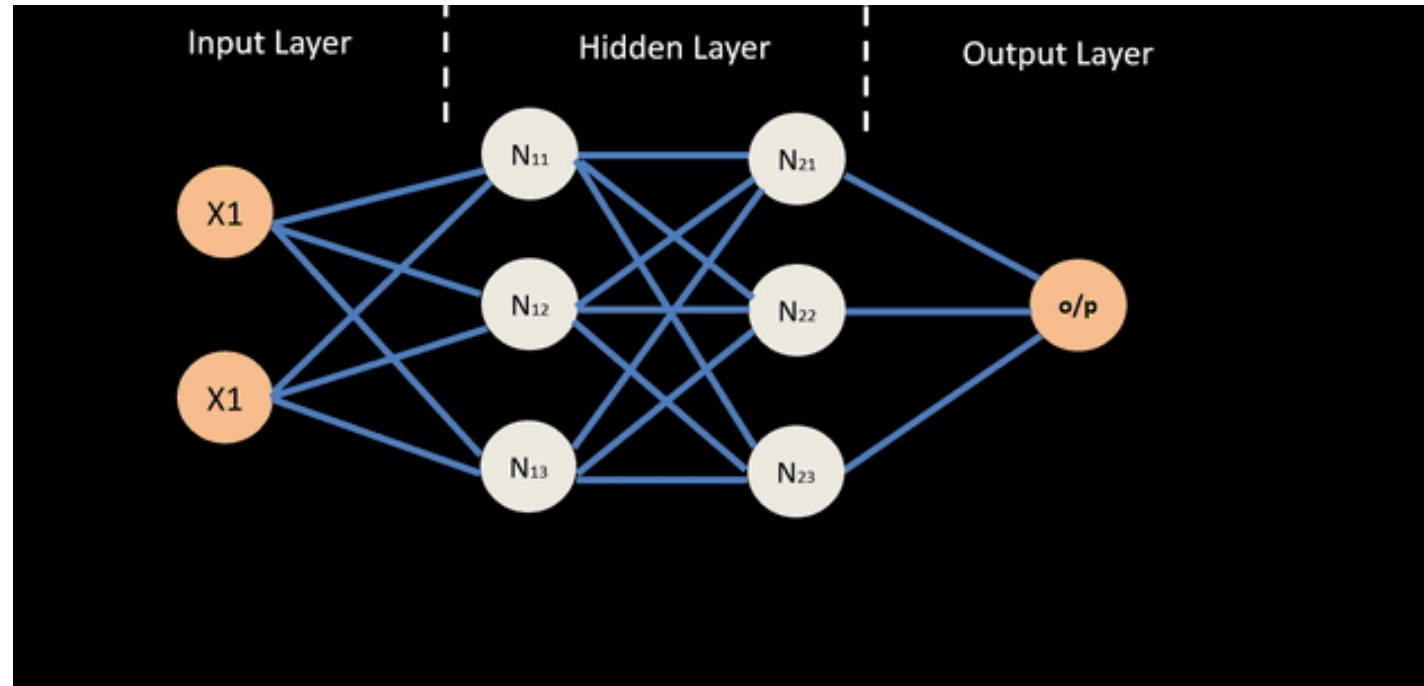
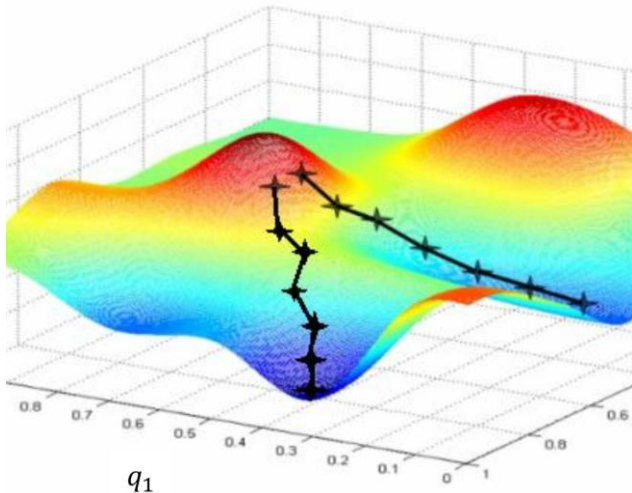


Backpropagation: algoritmo de treinamento de redes neurais

- Analogia com a preparação de um bolo
 - Adaptar a receita sucessivas vezes
 - ✓ Doce demais? Diminui o açúcar. Doce de menos, aumenta o açúcar.
- Numa rede neural,
 - Tentaremos pequenos ajustes em vários ingredientes ao mesmo tempo
- Aprendendo com uma medida consistente do erro/distância para o resultado ideal
 - Esse erro é chamado de perda (loss)
 - ✓ O objetivo do treinamento é minimizar o erro

Backpropagation: algoritmo de treinamento de redes neurais

- Ajusta os pesos da rede para minimizar erros
 - Em pequenos passos na direção descendente da curva de loss



Eliminação de Features

- Selecionar as features mais relevantes
 - para um modelo preditivo
- Melhorar a
 - precisão, velocidade e a generalização do modelo,
 - removendo features irrelevantes ou redundantes
 - ✓ que podem levar ao Overfitting
- Como
 - treina o modelo e remove recursivamente as features menos importantes até atingir o número desejado de features
 - ✓ Ou se a remoção de mais uma feature diminuir significativamente o desempenho do modelo

Recursive Feature Elimination (RFE)

```
base_estimator = LogisticRegression(multi_class='multinomial',  
solver='lbfgs', max_iter=500)  
rfe = RFE(estimator=base_estimator, n_features_to_select=10)
```

```
rfe.fit(X, y)
```

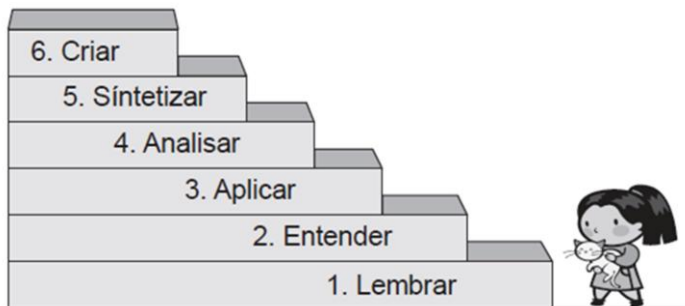
```
selected_columns = X.columns[rfe.support_]
print("\nColunas selecionadas pelo RFE:")
print(selected_columns)
```

Colunas selecionadas pelo RFE:

```
Index(['highway-mpg', 'city-mpg', 'horsepower', 'curb-weight', 'width',  
      'length', 'wheel-base', 'drive-wheels_fwd', 'fuel-system_2bbl',  
      'fuel-system_mpfi'],  
      dtype='object')
```

Se entender a piada é porque já está falando a língua dos nerds

Taxonomia de Bloom

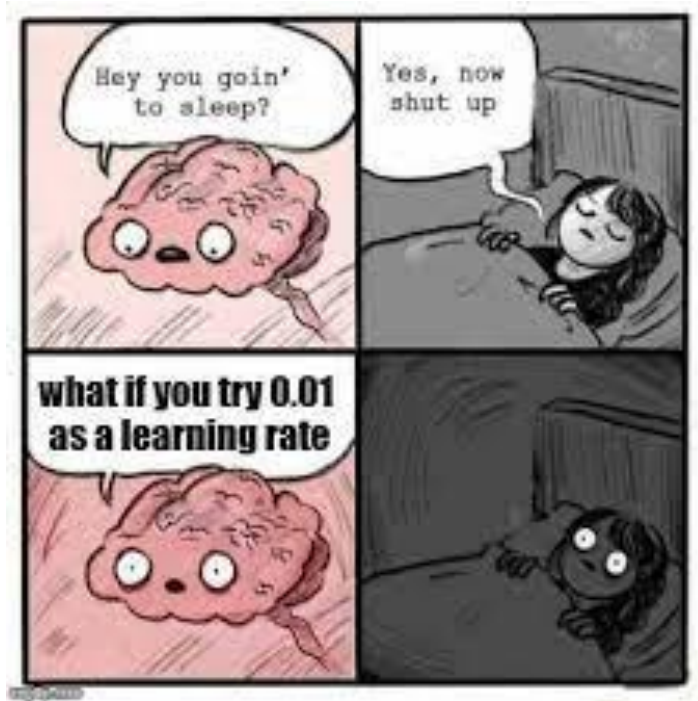


Novo teste para substituir o Teste de Turing



[Fonte da Imagem](#)

Se entender a piada é porque já está falando a língua dos nerds



[Fonte da Imagem](#)



[Fonte da Imagem](#)