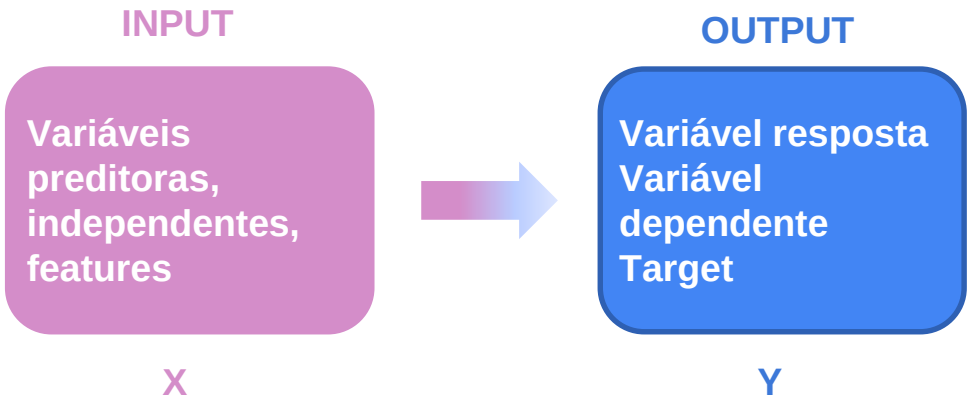
Modelos Lineares e Árvores

Professora Stella Sposito
stella.sposito@facens.br

Conceitos para modelagem

- Quando pensamos em modelagem, automaticamente pensamos em **variáveis de entrada (input) e variáveis de saída (output).**



Conceitos para modelagem

Aluno	Peso	Profissão	Idade	Horas de estudo	Tem cachorro?	Passou na disciplina
Julius	100	Dois empregos	50	2	1	1
Chris	50	Estudante	16	6	0	0
Rochelle	75	Cabelereira	45	7	0	1
Tonya	43	Estudante	13	10	1	0

Preditoras

Target

Passo a passo para a modelagem

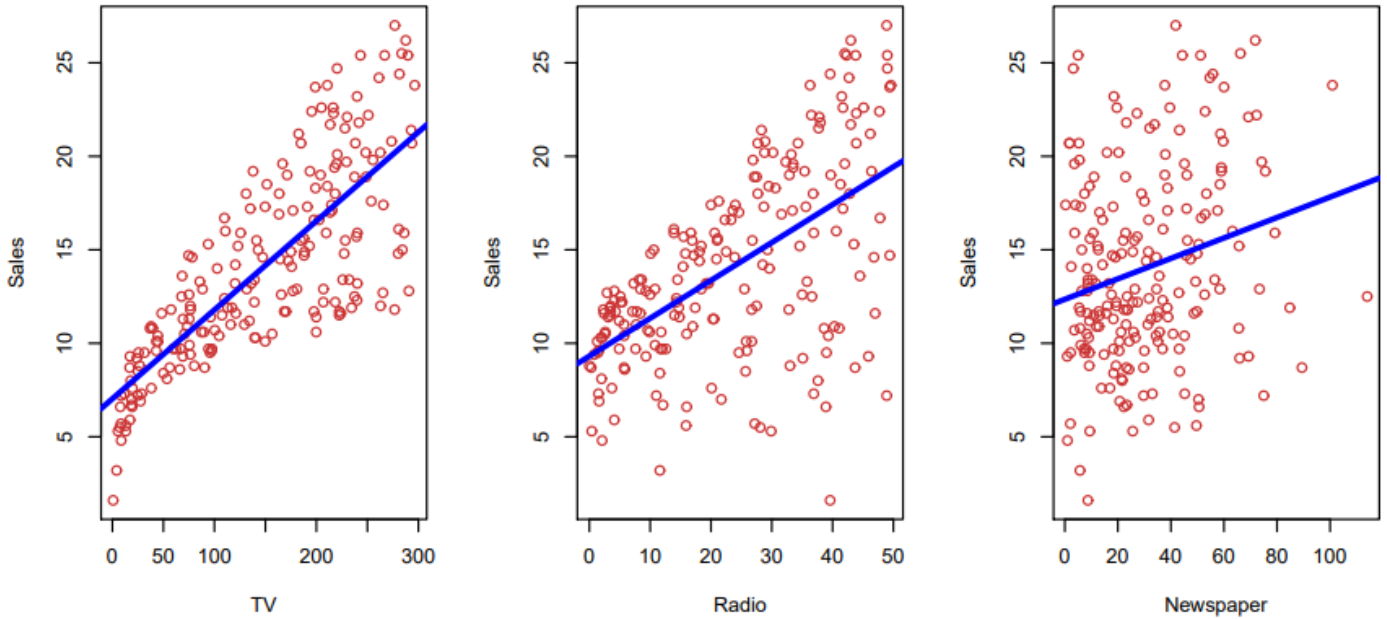
1. Definição de X e Y
2. Split - divisão em treino, teste (e validação, em alguns casos)
3. Escolha e definição do modelo
4. Treinamento (fit)
5. Otimização de hiperparâmetros
6. Realizar previsões (predict)
7. Avaliações
8. Interpretação do modelo

Passo a passo para a modelagem

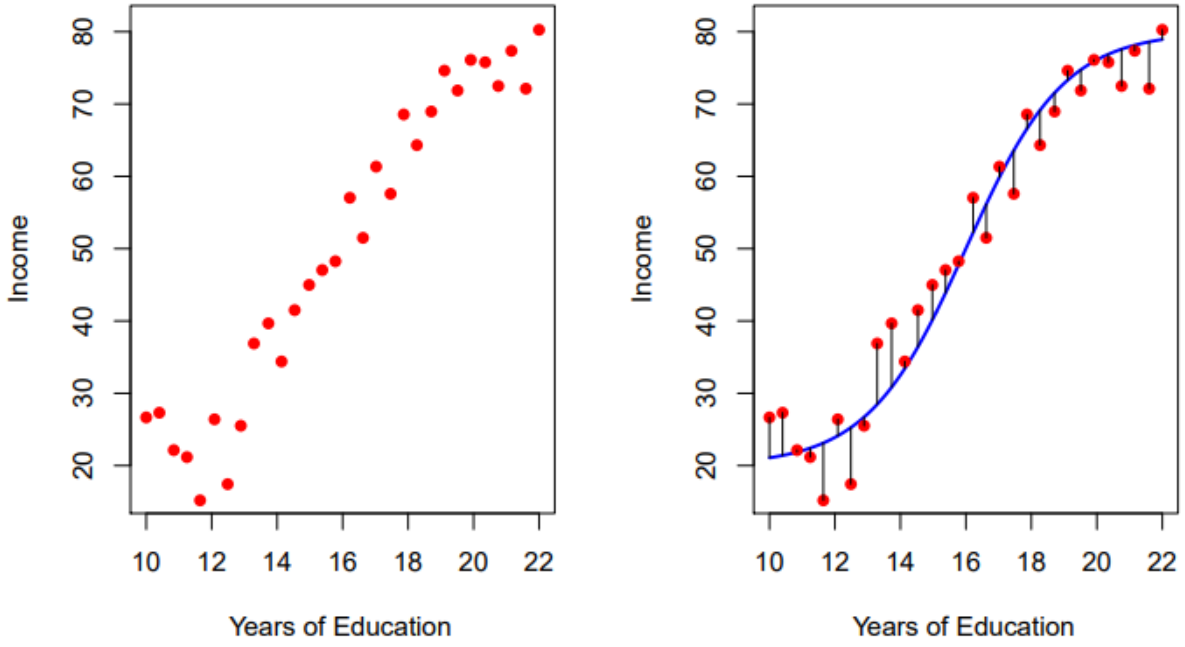
1. Definição de X e Y
2. Split - divisão em treino, teste (e validação, em alguns casos)
3. Escolha e definição do modelo
4. Treinamento (fit)
5. Otimização de hiperparâmetros
6. Realizar previsões (predict)
7. Avaliações
8. Interpretação do modelo

Conceitos para modelagem

- O plot abaixo relaciona vendas como uma função de TV, Rádio e Jornal;
- Cada linha azul representa um modelo simples que pode ser usado para prever vendas com base nas variáveis em questão.



Conceitos para modelagem



Modelos Lineares

- Um modelo linear prevê a target usando uma **função linear dos recursos de entrada**;
- Ele assume que os dados são linearmente separáveis e tenta aprender o peso de cada feature.

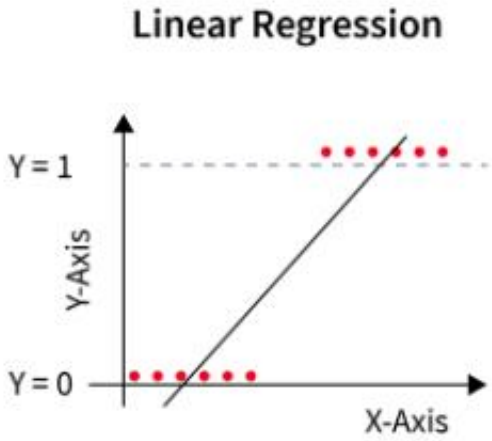
$$\hat{y}(w, x) = w_0 + w_1 x_1 + \ldots + w_p x_p$$

Variável que será prevista (target)

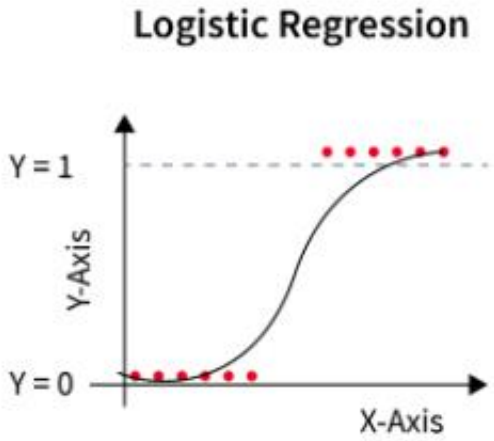
Intercepto (bias)

Features

Modelos Lineares



Tarefa de regressão:
Prevê valores numéricos contínuos



Tarefa de classificação:
Prevê classes binárias

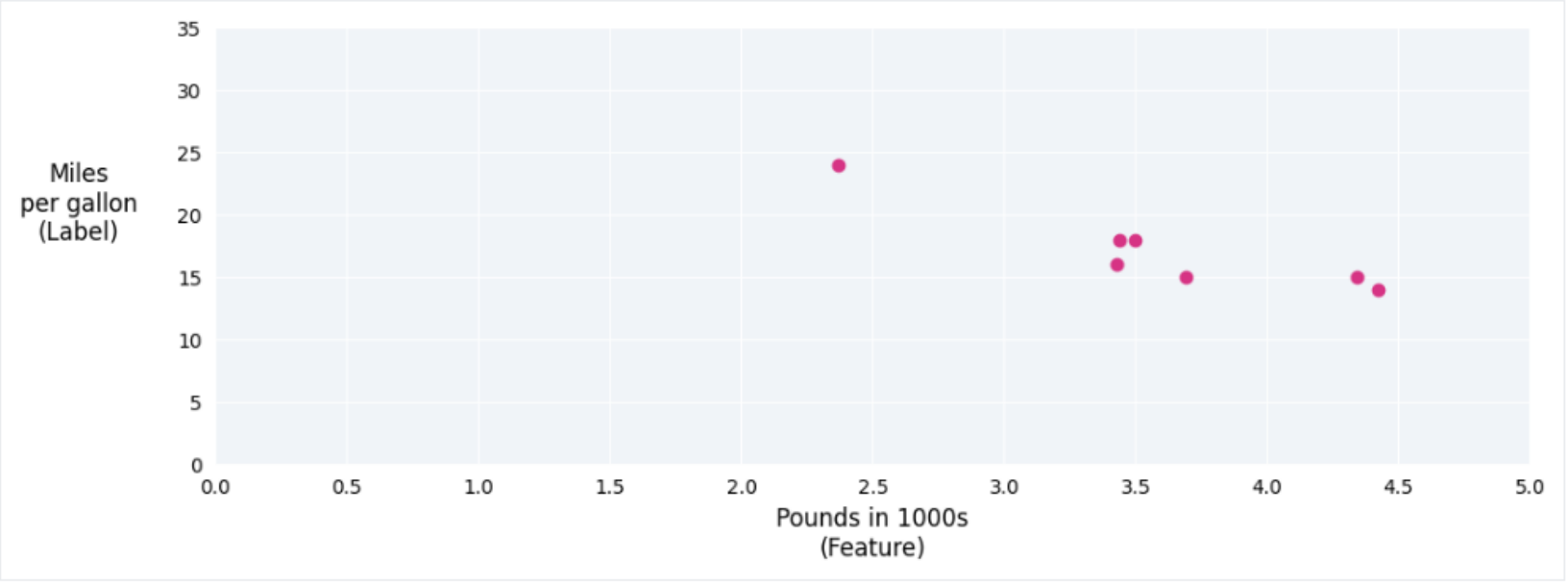
Regressão Linear

Regressão Linear

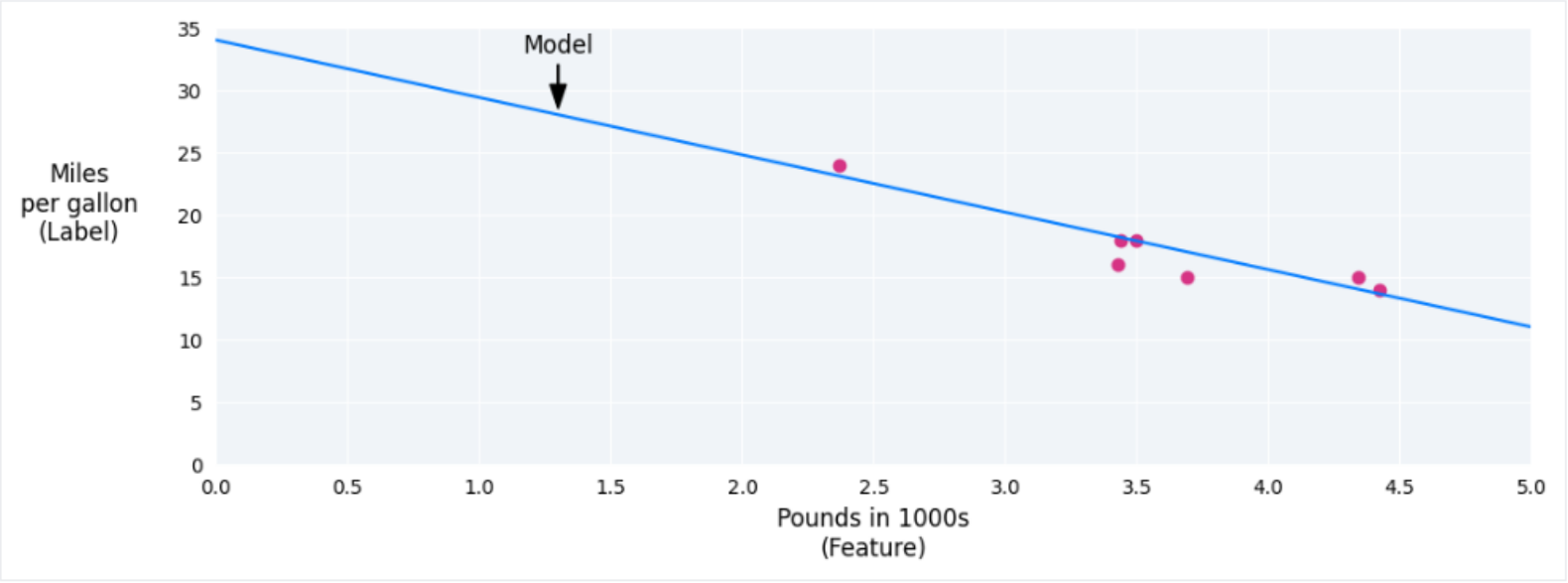
- Suponha que queremos prever a eficiência de combustível de um carro em milhas por galão (ou km/L) com base no peso do veículo.

Libras em milhares (recurso)	Milhas por galão (rótulo)
3.5	18
3,69	15
3,44	18
3,43	16
4,34	15
4,42	14
2,37	24

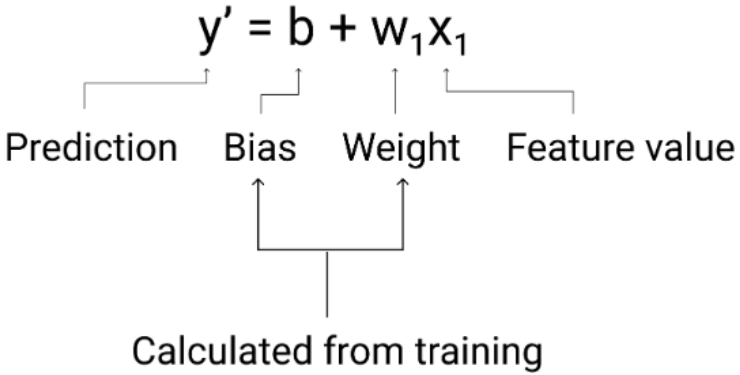
Regressão Linear



Regressão Linear



Regressão Linear



- y' - é o rótulo previsto, ou seja, a saída.
- b - é o **viés** do modelo. O viés é o mesmo conceito do intercepto y na equação algébrica de uma reta. Em ML, o viés às vezes é chamado de w0. O viés é um **parâmetro** do modelo e é calculado durante o treinamento.
- w - é o **peso** do recurso. O peso é o mesmo conceito da inclinação na equação algébrica de uma reta. O peso é um **parâmetro** do modelo e é calculado durante o treinamento.
- x - é um **atributo**, ou seja, a entrada.

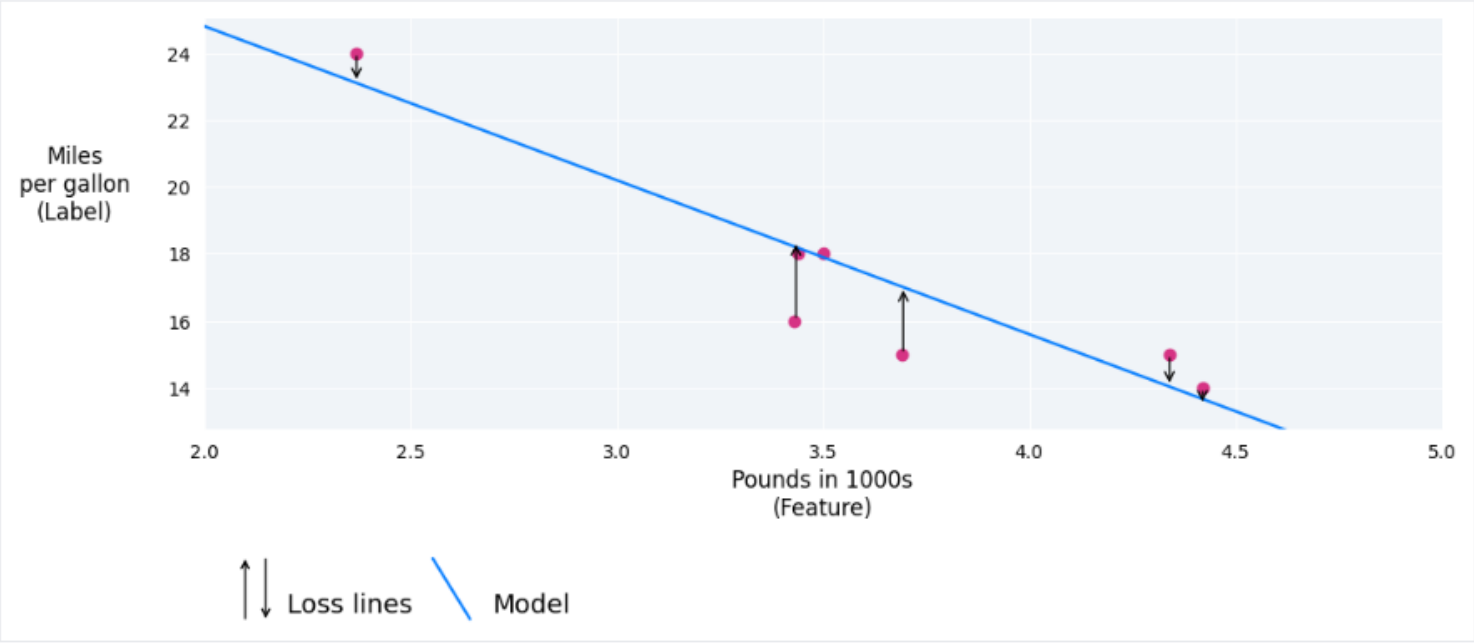
Objetivo da Regressão Linear

- Nós queremos fazer previsões (**achar o y**), e temos apenas os valores de x (**valores da nossa variável independente**).
- O treinamento de uma Regressão Linear consiste em **encontrar os valores de w (peso) e b (viés/intercepto) que minimizam o erro entre as previsões e os valores reais.**

O erro é representado pela **Função Custo**:

$$J(w,b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - (wx_i + b))^2$$

Erro


Miles
per gallon
(Label)
24
22
20
18
16
14
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Pounds in 1000s
(Feature)
Loss lines
Model

Métodos para minimizar o erro

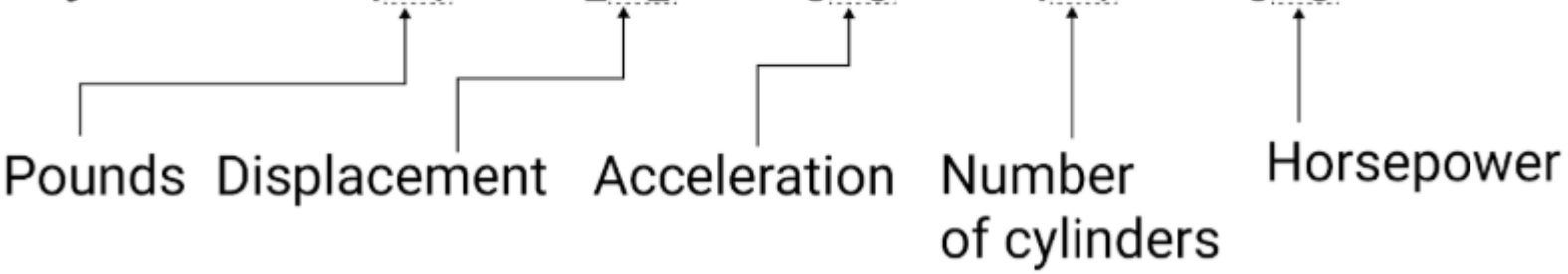
Mínimos Quadrados Ordinários → O OLS minimiza o erro de forma analítica ao derivar a função custo e encontrar as fórmulas fechadas de w e b.

Gradiente Descendente → O gradiente descendente calcula a inclinação da função de custo em relação a ***w*** e ***b***. Atualiza-se ***w*** e ***b*** em pequenas etapas para reduzir o erro.

Regressão Linear Múltipla

Quando temos mais do que uma variável independente tentando prever a variável resposta, usamos a regressão linear múltipla.

$$y' = b + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5$$



Métricas de Avaliação Regressão Linear

Métricas de Erro

Erro Quadrático Médio (MSE)
Raiz do Erro Quadrático Médio (RMSE)
Erro Absoluto Médio (MAE)
Erro Percentual Absoluto Médio (MAPE)

Métricas de Explicabilidade

R^2
R^2 Ajustado

Código

Regressão Logística

Conceitos Gerais

- A regressão logística é usada para estimar **a probabilidade** de uma instância pertencer a uma classe específica, como um classificador binário.

- Por exemplo, se a probabilidade estimada for maior que 50%, o modelo prediz que a instância pertence a essa classe (positiva, ou 1), se for menor, não pertence a essa classe (0)

Pensando em prever a probabilidade de inadimplência

P(inadimplência = sim|saldo)

y X

$$P(y = 1 \mid X)$$

Conceitos Gerais

- Os valores de P(inadimplência = sim | saldo) variam entre 0 e 1. Ou seja, **a probabilidade prevista de inadimplência para uma pessoa com determinado saldo** varia de 0% a 100%.

- Assim, para qualquer valor de saldo, é possível prever a inadimplência. Por exemplo, pode-se prever inadimplência = Sim para qualquer indivíduo cujo p(saldo) > 0,5.

- Alternativamente, se uma empresa desejar ser conservadora na previsão de indivíduos com risco de inadimplência, pode optar por usar um **limite inferior**, como p(saldo) > 0,1.

Vou classificar essa pessoa como inadimplente ou não?

Depende do meu **threshold**.

Como funciona a Regressão Logística?

- Por ser um modelo linear, a regressão logística calcula a soma ponderada das características de entrada (X, incluindo o viés), gerando a saída logística desse resultado, ou seja, uma probabilidade dada por uma **função sigmoide**.

1. A função sigmoide pega qualquer número e transforma numa probabilidade entre 0 e 1.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

2. A soma ponderada das features (igual à regressão linear) é dada por esta fórmula:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d$$

Como funciona a Regressão Logística?

3. Como a regressão logística usa a função sigmoide para estimar a probabilidade, temos:

$$p = \sigma(y)$$

4. A equação da regressão logística que transforma a saída linear em probabilidade é dada por:

$$p = \frac{1}{1 + e^{-y}}$$

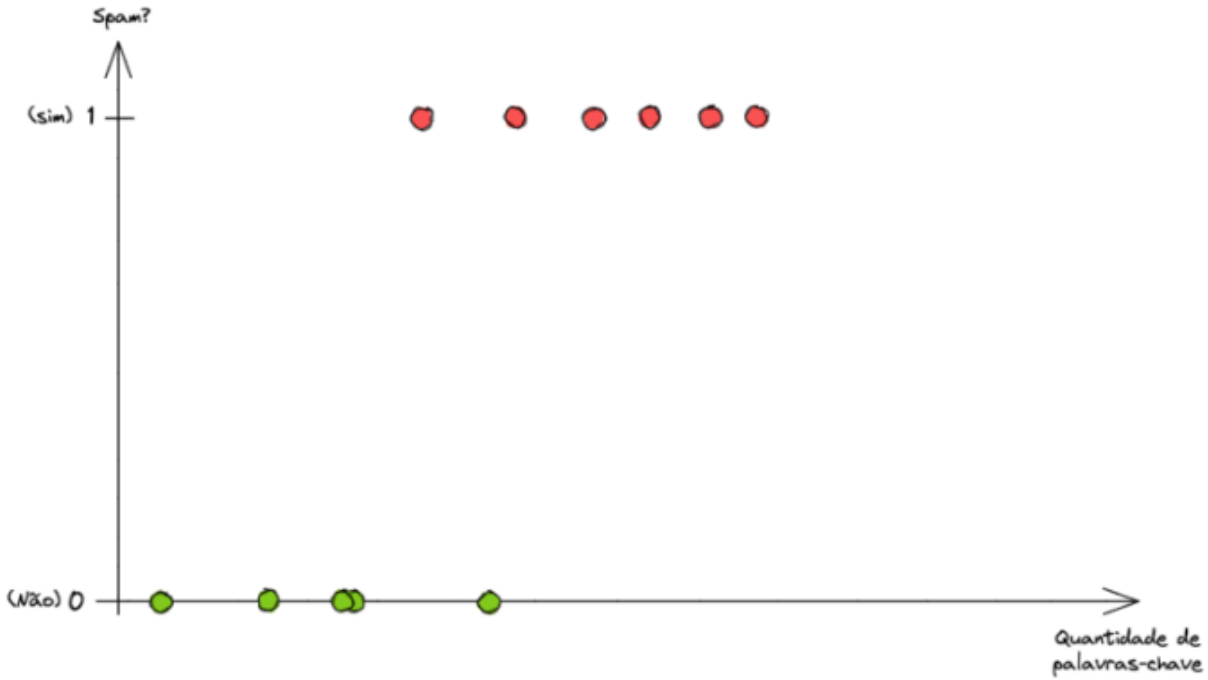
Exemplo prático

- Vamos imaginar que estamos tentando prever se um email será classificado como spam ou não.

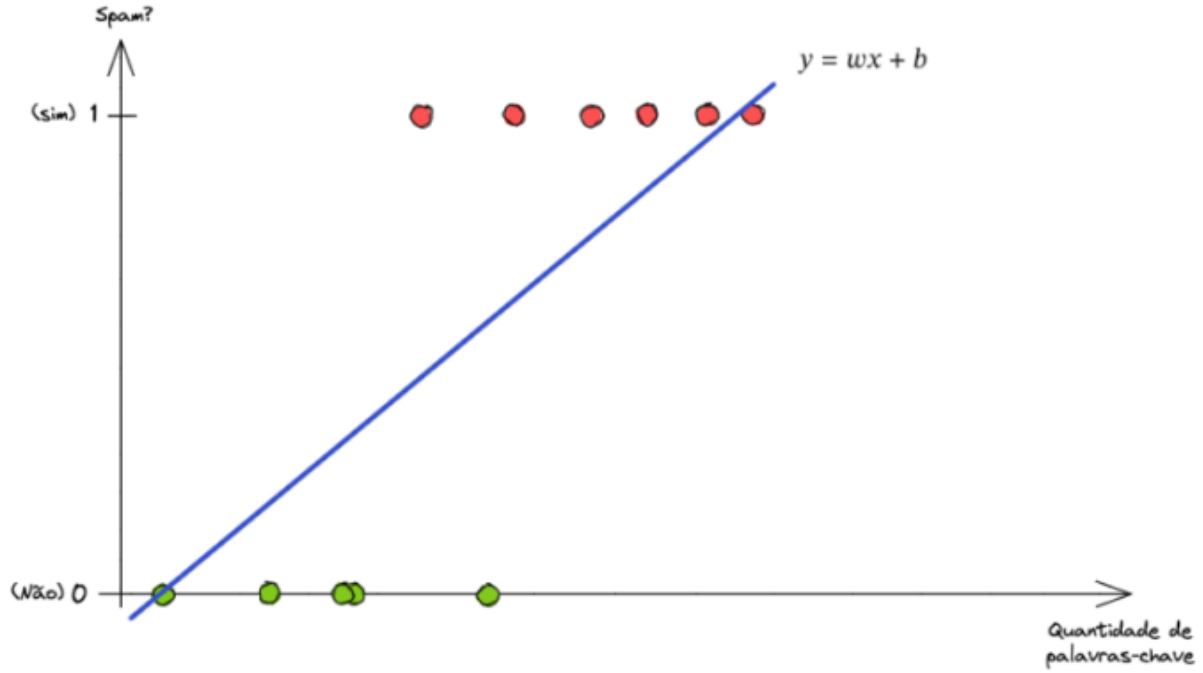
- Uma das características que nosso modelo rastreia é a quantidade de palavras-chave no corpo do e-mail (variável x). Essas “palavras-chave” são palavras comumente usadas em spams, como “promoção”, “sorteado”, “premiado”, etc.

Qtd Palavras	Spam?
7	Sim (1)
4	Não (0)
1	Não (0)
6	Sim (1)
8	Sim (1)
6	Não (0)
4	Não (0)
3	Não (0)
9	Sim (1)
5	Sim (1)
10	Sim (1)

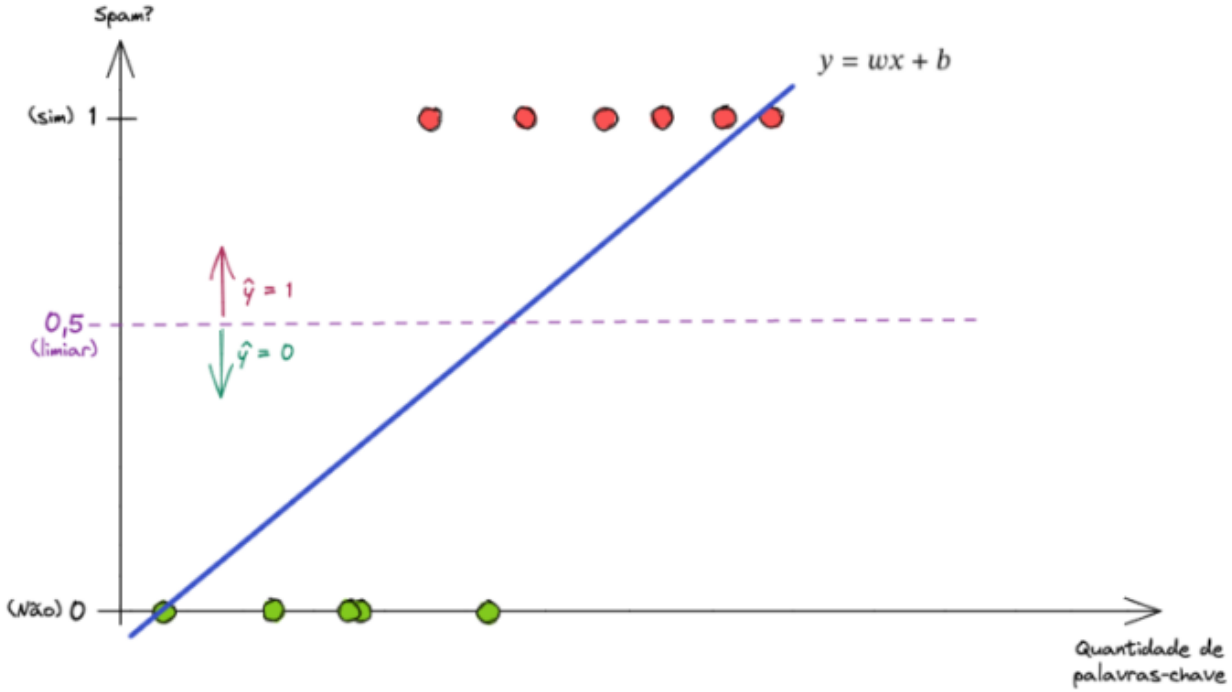
Exemplo prático



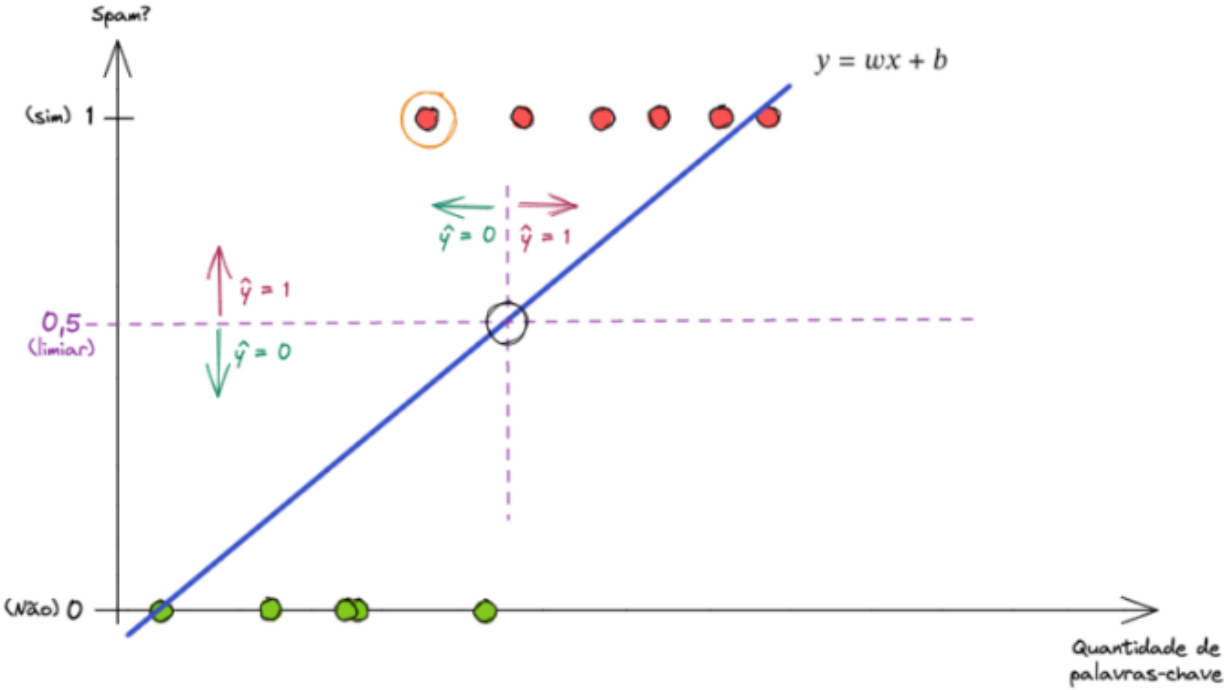
Exemplo prático



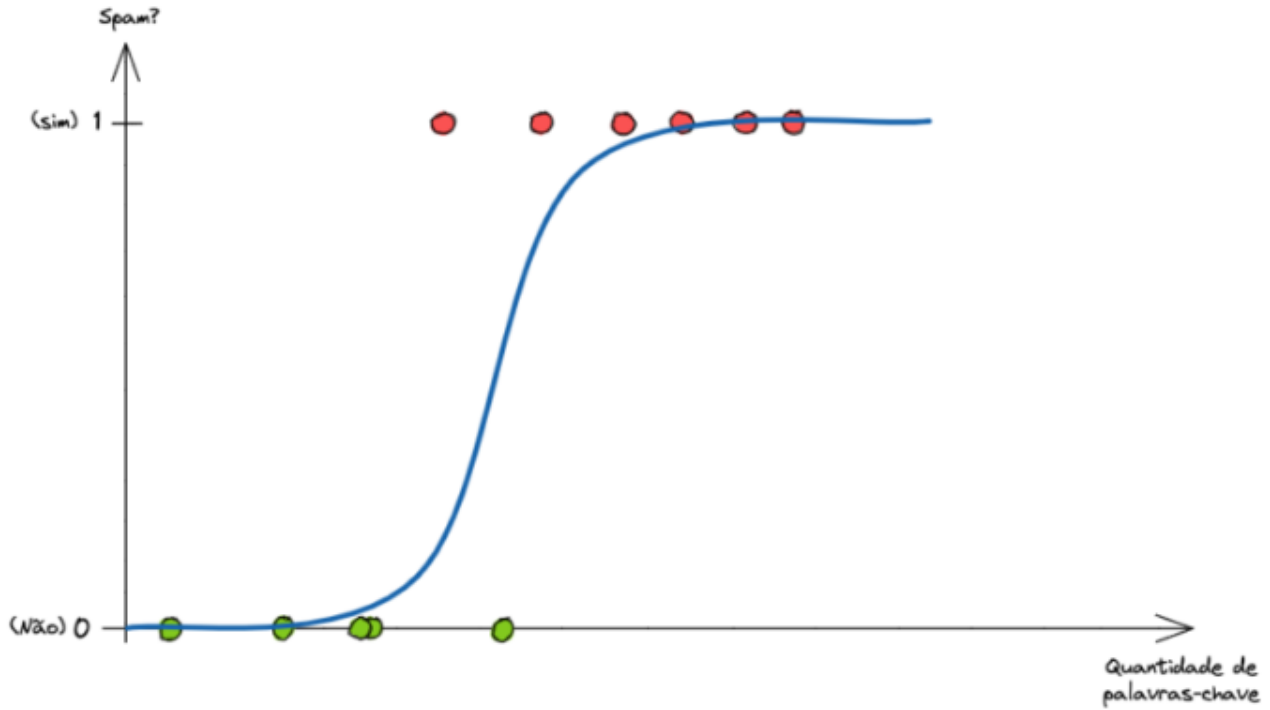
Exemplo prático



Exemplo prático

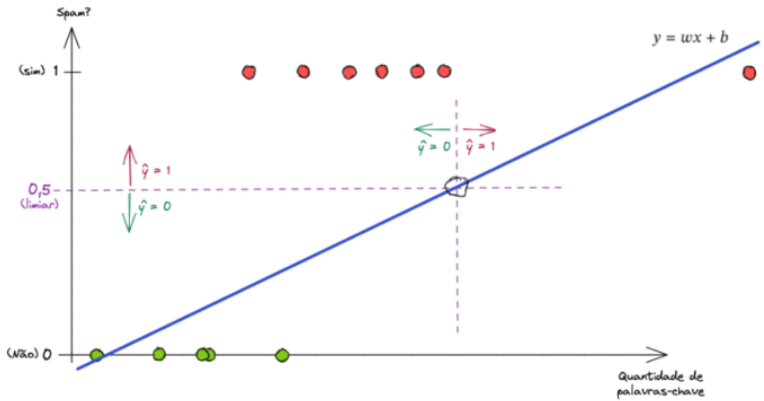


Aplicando a Regressão Logística

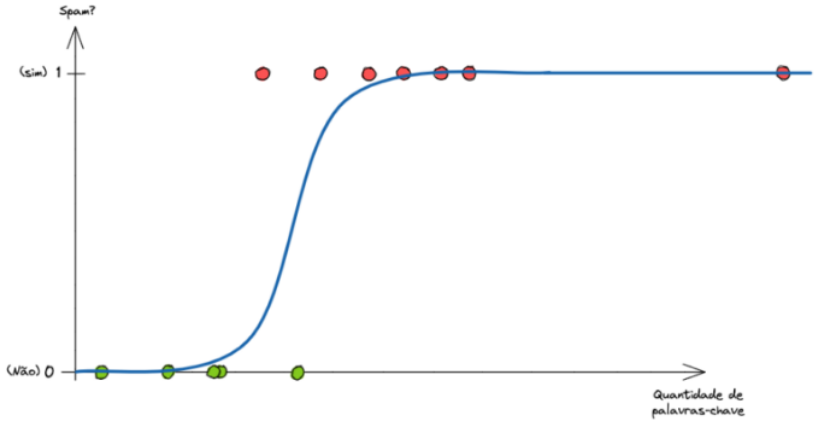


O que acontece se adicionarmos mais um dado?

Regressão Linear

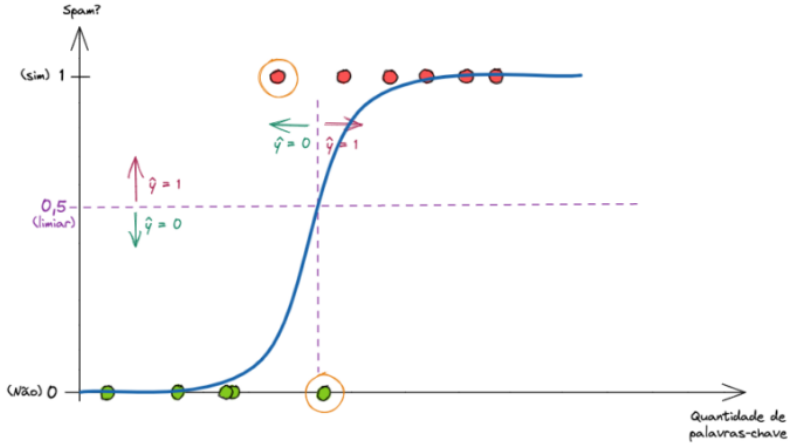


Regressão Logística

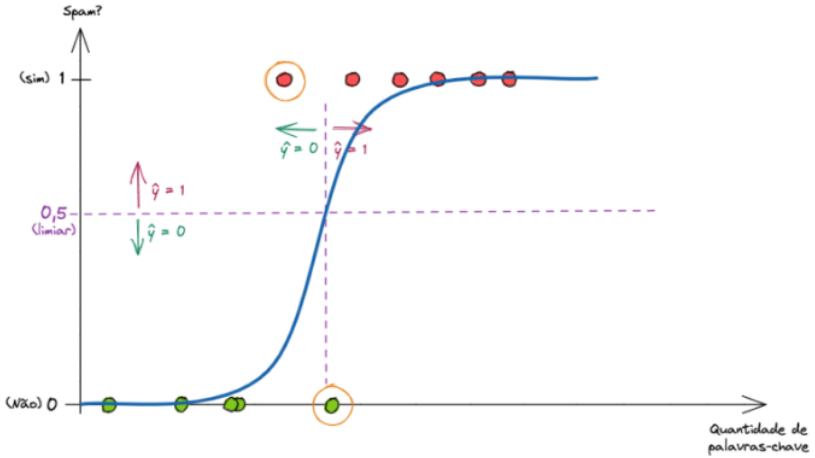


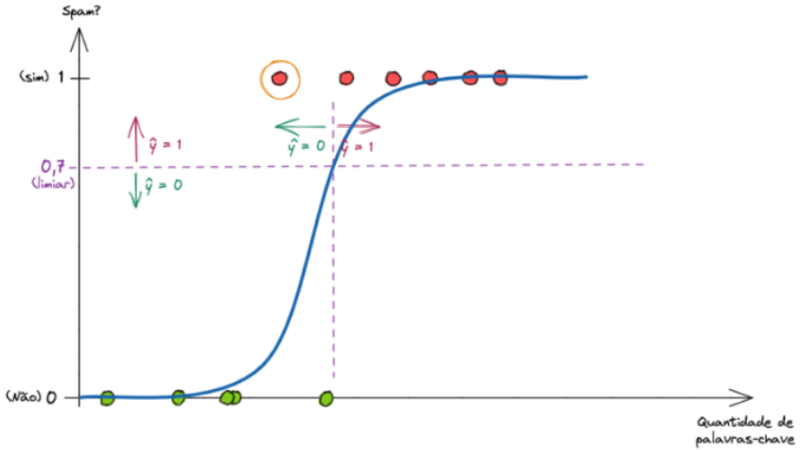
Interpretação da RegLog

- A saída de um modelo de Regressão Logística é a probabilidade de que a classe y será igual a 1 dado uma certa entrada x.
- No nosso problema: é a probabilidade do email ser spam dada uma quantidade de palavras-chave.
- Considerando um limiar de 0.5, imagine que o modelo nos dê os seguintes valores previstos: ŷ = 0.6, 0.3, 0,9.



Qual limiar acerta mais?



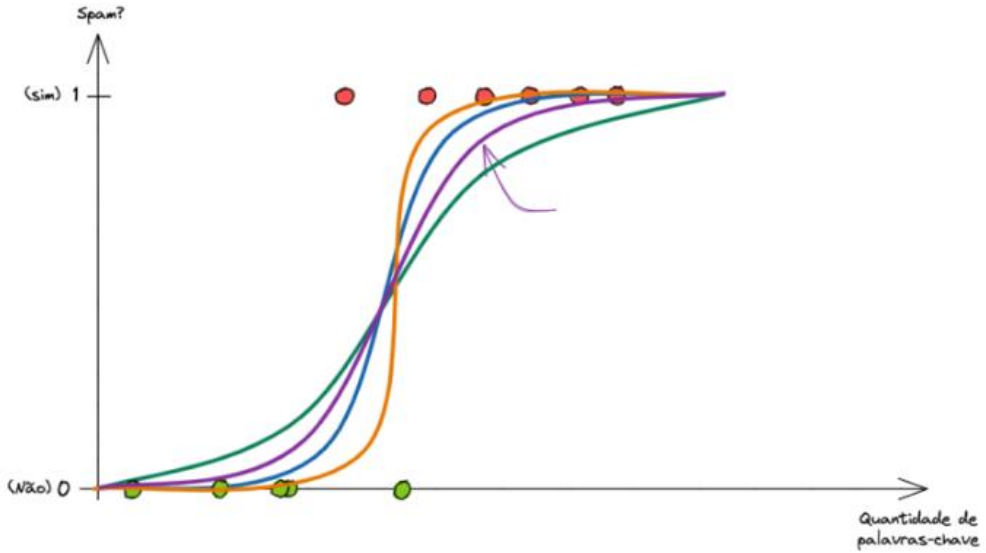


Como encontrar a melhor sigmoide?

- Encontrar a melhor sigmoide significa **ajustar o modelo para aprender o padrão geral dos dados**. Isso é o que permite prever corretamente casos novos, que não vimos no treinamento.

- Assim como na regressão linear, no treinamento, os parâmetros w e b serão aprendidos e irão gerar diferentes Funções Logísticas.

- Para encontrarmos qual dupla de parâmetros tem o melhor desempenho, precisamos saber qual erra menos.



Métodos para minimizar o erro em RegLog

Logaritmo da Perda (Log Loss ou Cross-Entropy)

É a função de erro que mede o quão ruim é a probabilidade prevista. Se o modelo dá **probabilidade alta** para a classe certa – **erro baixo**. Se o modelo dá **probabilidade baixa** para a classe certa – **erro altíssimo**.

$$\text{Log Loss} = \begin{cases} -\log(1-\hat{y}) & \text{se } y = 0 \\ -\log(\hat{y}) & \text{se } y = 1 \end{cases}$$

Classe real	Probabilidade prevista	Fórmula	Custo
1	p = 0.2	−log(0.2)	**1.609**
1	p = 0.005	−log(0.005)	**5.298**

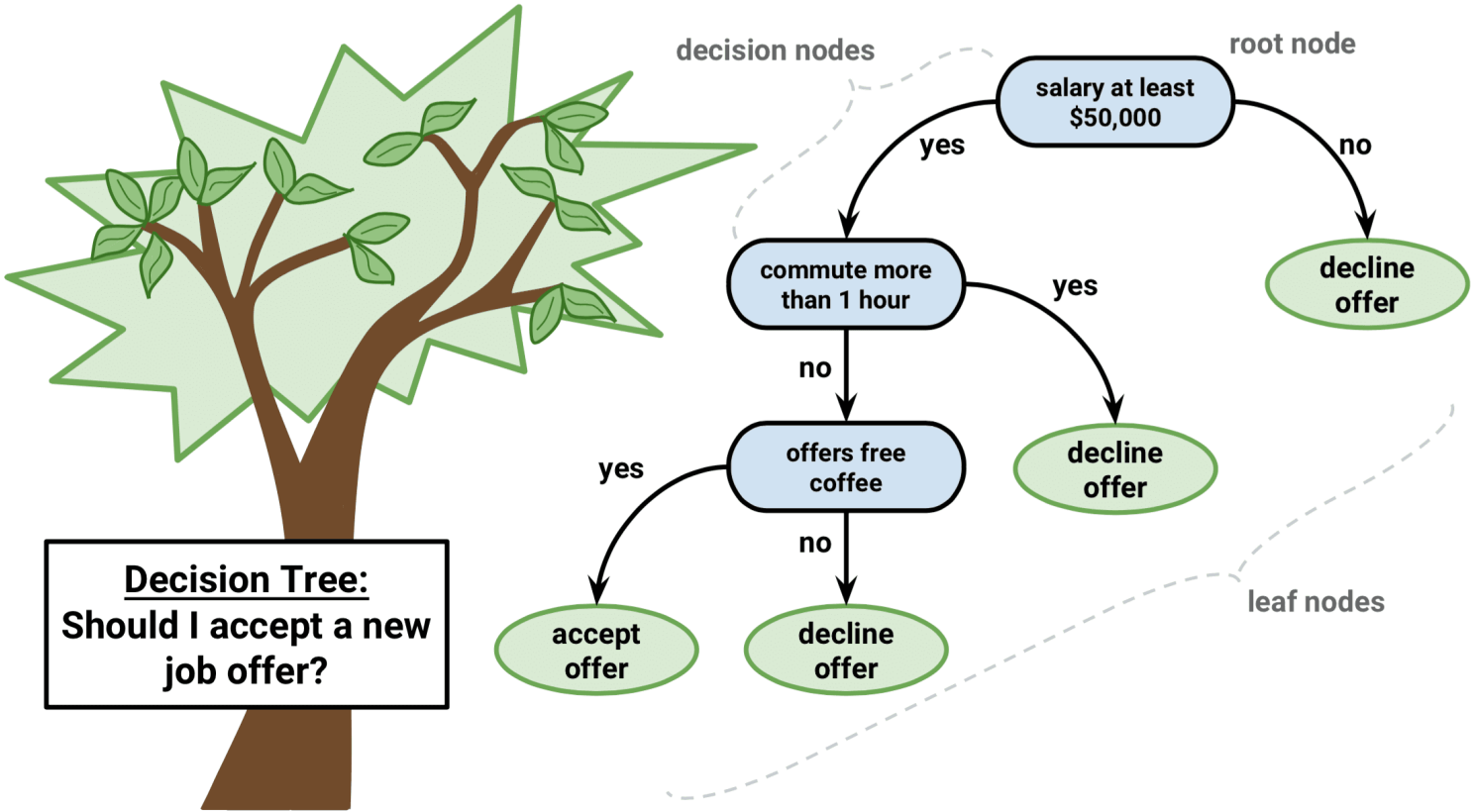
Como avaliar Regressão Logística?

Da mesma forma que avaliamos modelos de classificação, no geral.

- Classification report (acurácia, recall, precisão, f1-score)
- Matriz de Confusão
- Curvas ROC-AUC
  AUC-PR
- Gain Chart

Árvores de Decisão

Árvore de Decisão



Conceitos Gerais

- Neste algoritmo, vários pontos de decisão serão criados (nós). Cada nó gera dois caminhos possíveis para uma decisão, e esses caminhos são os ramos da árvore.

Dia	Sol?	Vento?	Vou para praia?
1	Sim	Sim	Não
2	Sim	Sim	Não
3	Sim	Não	Sim
4	Não	Não	Não
5	Não	Sim	Não
6	Não	Sim	Não

Como definir os nós e os ramos?

- Existem duas formas de definir a melhor estrutura para uma árvore de decisão: entropia e gini.

Os métodos utilizados pelos algoritmos irão buscar as variáveis que possuem maior relação com a variável target, colocando-as no topo da árvore, em seus nós principais.

Neste exemplo, a árvore não precisaria conferir se existe vento ou não, pois independente desta informação o resultado seria o mesmo.

Entropia

- A entropia vem da teoria da informação e mede o nível de impureza ou incerteza de um conjunto.

- ***Entropia alta*** ➡ classes muito misturadas;
- ***Entropia baixa*** ➡ classes mais “organizadas”, mais puras.

Uma árvore de decisão tenta **escolher a variável que mais reduz a entropia** após o split.
O nome disso é **Information Gain** (ganho de informação).

Partindo da entropia, o algoritmo confere o ganho de informação de cada variável. Aquela que apresentar maior ganho de informação será a variável do primeiro nó da árvores.

```
model = DecisionTreeClassifier(criterion='entropy', random_state=42)
model.fit(X, y)
```

Índice de Gini

- O **Gini** é uma outra forma de medir impureza. Ele é mais simples computacionalmente e costuma ser o padrão no Scikit-Learn.

- *Gini = 0* ➡ nó completamente puro;
- *Gini alto* ➡ muita mistura de classes.

Ele mede a probabilidade de escolher uma amostra aleatória e classificá-la **erradamente** se seguíssemos a distribuição do nó.

A variável preditora com o menor índice Gini será a escolhida para o nó principal da árvore, pois um baixo valor do índice indica maior ordem na distribuição dos dados.

```
model = DecisionTreeClassifier(criterion='gini', random_state=42)
model.fit(X, y)
```

Como avaliar Árvores de Decisão?

Da mesma forma que avaliamos modelos de classificação, no geral.

- Classification report (acurácia, recall, precisão, f1-score)
- Matriz de Confusão
- Curvas ROC-AUC
  AUC-PR
- Gain Chart

Classification report

	precision	recall	f1-score	support
malignant	0.97	0.97	0.97	63
benign	0.98	0.98	0.98	108
accuracy			0.98	171
macro avg	0.97	0.97	0.97	171
weighted avg	0.98	0.98	0.98	171

1. Class Name
2. Each Class Metrics
3. Actual Instance of each class
4. Overall Accuracy Metric
5. Metrics Average

Próxima aula...

- Aprofundamento em técnicas de avaliação de modelos;
- Interpretação de modelos;
- Outros modelos de classificação...

Tarefa de aula + tarefa de casa

Tarefa de aula

- Divisão em grupos e resolução das perguntas do notebook.
- Construção de um modelo de **Regressão Logística** para resolver o problema de classificar novos pacientes com câncer.

Tarefa de casa

- Individual, tentar resolver o mesmo problema mas agora usando **Árvores de Decisão.**
- Comparar o resultado do modelo construído em aula com o modelo feito em casa;


Facens
AQUI TEM ENGENHARIA