

DATA SCIENCE

AULA 4 - Modelos de Decisão

Prof. Gabriel Resende Machado



gabrielmachado@unifeso.edu.com



<https://www.linkedin.com/in/machadogabriel>



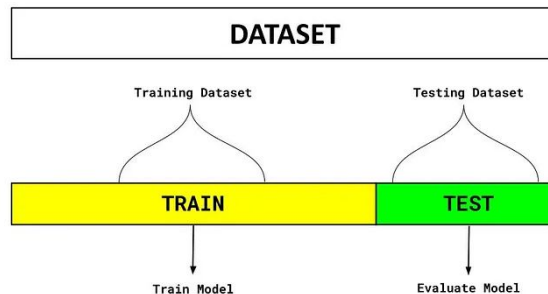
<https://github.com/UNIFESO-Gabriel/data-science>

Técnicas de amostragem dos dados

- Métodos de amostragem resultam em estimativas de desempenho mais confiáveis dos modelos de aprendizado a partir de conjuntos disjuntos de treinamento e teste;
 - os dados de treino servem para ajustar os parâmetros do modelo buscando generalização;
 - os dados de teste simulam a apresentação de novas amostras ao preditor;
 - os principais métodos de amostragem dos dados são o (i) *holdout*, (ii) amostragem aleatória, (iii) validação cruzada e (iv) *bootstrap*.

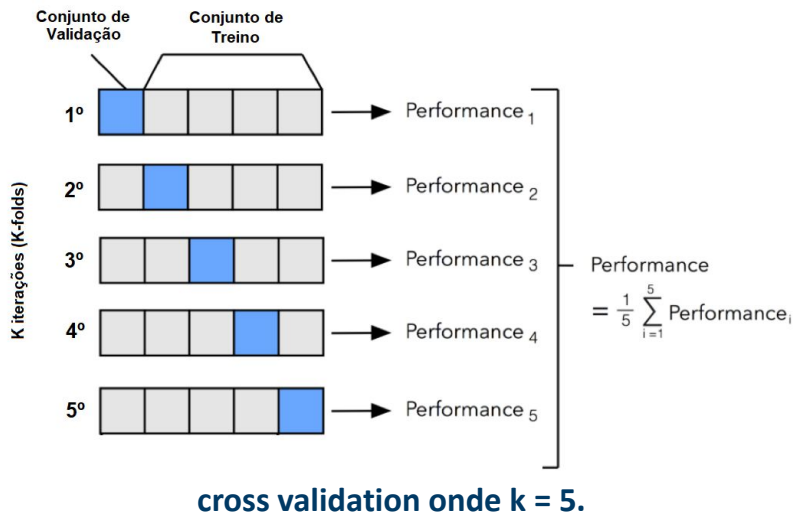
Técnicas de amostragem dos dados: *holdout*

- No caso do *holdout*, o conjunto de dados é dividido em uma proporção p para treinamento e $1-p$ para teste;
 - Normalmente, emprega-se $p = \{2/3, 0,7 \text{ ou } 0,8\}$. Em tarefas de Deep Learning, com conjuntos de dados muito grandes, emprega-se $p \geq 0,95$;
 - É comum haver também uma variação do *holdout* que realiza a separação dos dados em treino, validação e teste.
 - O conjunto de validação serve, principalmente, para avaliar o desempenho do modelo sem gerar vieses com o conjunto de teste.
- O *holdout* não permite avaliar o desempenho do modelo perante diferentes combinações, como é feito pela validação cruzada.



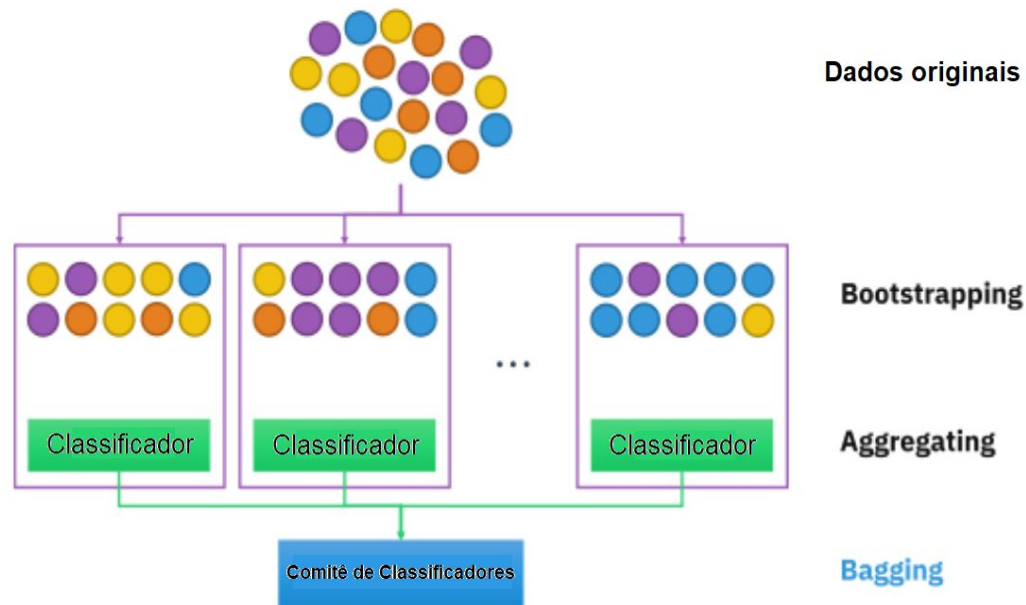
Técnicas de amostragem dos dados: *cross validation*

- Na validação cruzada, o conjunto de exemplos é dividido em k subconjuntos do mesmo tamanho.
- Em um subconjunto k , $n-1$ partições são utilizadas para treino de um modelo específico, que é avaliado na partição restante;
- O processo é repetido k vezes e o desempenho final do modelo é dado pela média dos desempenhos observados em cada subconjunto.
 - É importante ressaltar que é importante exibir o desvio padrão juntamente com a média.
- Uma variação dessa abordagem é aplicar o holdout e, após, o cross validation no conjunto de treino.



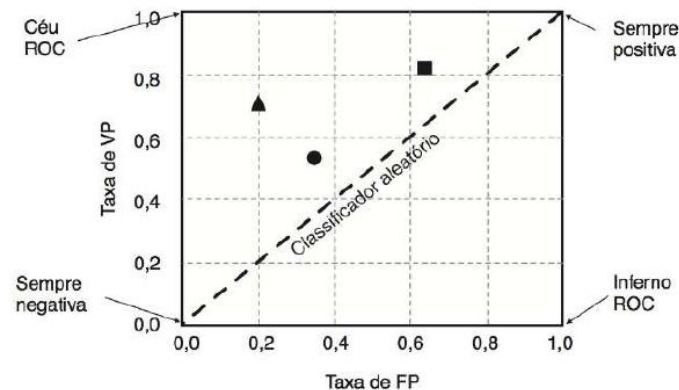
Técnicas de amostragem dos dados: *bootstrap*

- *Bootstrap* é uma técnica de reamostragem que envolve criar vários *datasets* escolhendo registros aleatoriamente com reposição;
- Com isso, um mesmo exemplo pode estar presente em um determinado subconjunto mais de uma vez. Os exemplos não selecionados compõem o conjunto de teste;
- A abordagem de treinar vários classificadores em conjuntos de dados gerados por *bootstrap* se chama **bagging**. A classificação final do comitê pode ser feita via média ou voto majoritário.

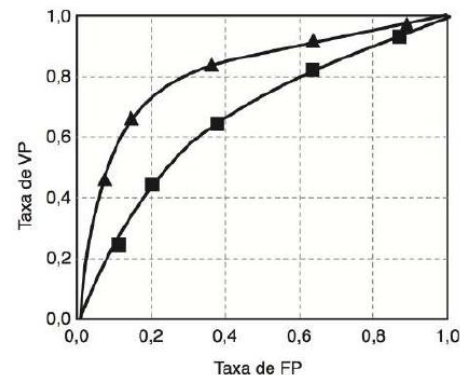


A Curva ROC e AUC

- O gráfico ROC é uma forma alternativa de se avaliar classificadores binários, onde os eixos X e Y representam as taxas de falsos positivos e verdadeiros positivos, respectivamente;
- Um gráfico ROC possui uma **linha diagonal** que representa o desempenho aleatório de um classificador. Qualquer classificador abaixo dessa linha possui um desempenho pior do que um classificador aleatório.
- O **AUC** é o valor da área abaixo da curva ROC que varia de 0 (pior) a 1 (melhor), plotada a partir da escolha de diferentes limiares para um determinado classificador.



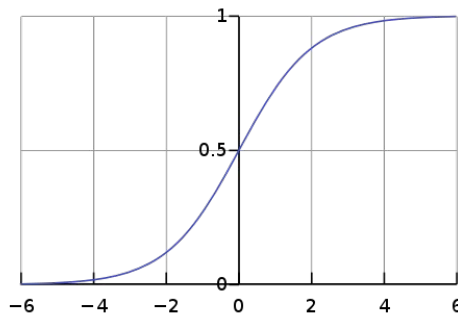
▲ Algoritmo A ■ Algoritmo B ● Algoritmo C



▲ Algoritmo A ■ Algoritmo B

Regressão Logística

- Ao contrário da Regressão Linear, a Regressão Logística é voltada para **problemas de classificação**;
- É um algoritmo recomendado para problemas de **classificação binária**;
- Produz como saída **uma probabilidade de uma amostra pertencer à uma determinada classe**;
 - **Utiliza a função sigmoide para realizar o cálculo das probabilidades:** $\frac{1}{1 + e^{-(\beta_0 + \vec{\beta}_1 x)}}$
 - β_0 e $\vec{\beta}_1$ são atualizados via gradiente descendente e o rótulo é atribuído por meio de um limiar (geralmente 0,5).



Regressão Logística: Gradiente Descendente

- Objetivo: encontrar os parâmetros β_0 e $\vec{\beta}_1$ tal que minimizem a função de custo J;

$$J(\beta_0, \vec{\beta}_1) = -\frac{1}{m} \sum_{i=1}^m y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

- Aplicar iterativa e simultaneamente o gradiente descendente à função de custo J:

- $\beta_0 = \beta_0 - \alpha \frac{\partial}{\partial \beta_0} J(\beta_0, \vec{\beta}_1)$

- $\beta_{1_j} = \beta_{1_j} - \alpha \frac{\partial}{\partial \beta_{1_j}} J(\beta_0, \vec{\beta}_1)$

- Onde, tem-se:

- $\frac{\partial}{\partial \beta_{1_j}} J(\beta_0, \vec{\beta}_1) = \frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)}) x^{(i)}$

- $\frac{\partial}{\partial \beta_0} J(\beta_0, \vec{\beta}_1) = \frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})$











Exercício 1

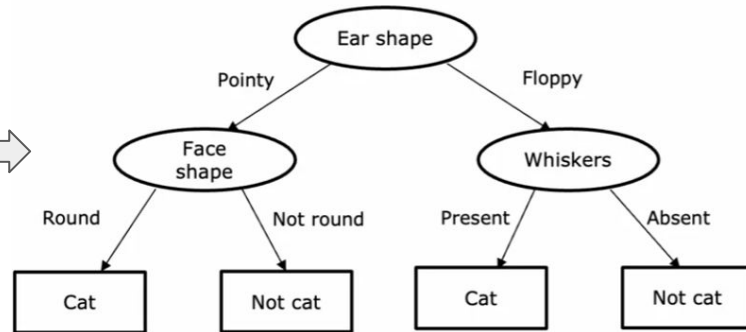
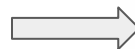
- Classifique o estado de um carro a partir de seus atributos. Utilize a Regressão Logística para isso. Exiba ao final a curva ROC do modelo.



Árvores de Decisão

- Modelo de aprendizado supervisionado de caixa-branca, formado por uma estrutura hierárquica que consiste em um nó raiz, ramos, nós internos e nós folhas;
- O objetivo é criar um modelo que realiza a predição de um valor para uma variável alvo a partir de regras de decisão simples, geradas a partir dos dados de treino;
- Os algoritmos de árvores de decisão mais conhecidos são o C4.5 e o CART (este implementado no Scikit-Learn).

	Ear shape	Face shape	Whiskers	Cat
	Pointy	Round	Present	1
	Floppy	Not round	Present	1
	Floppy	Round	Absent	0
	Pointy	Not round	Present	0
	Pointy	Round	Present	1
	Pointy	Round	Absent	1
	Floppy	Not round	Absent	0
	Pointy	Round	Absent	1
	Floppy	Round	Absent	0
	Floppy	Round	Absent	0

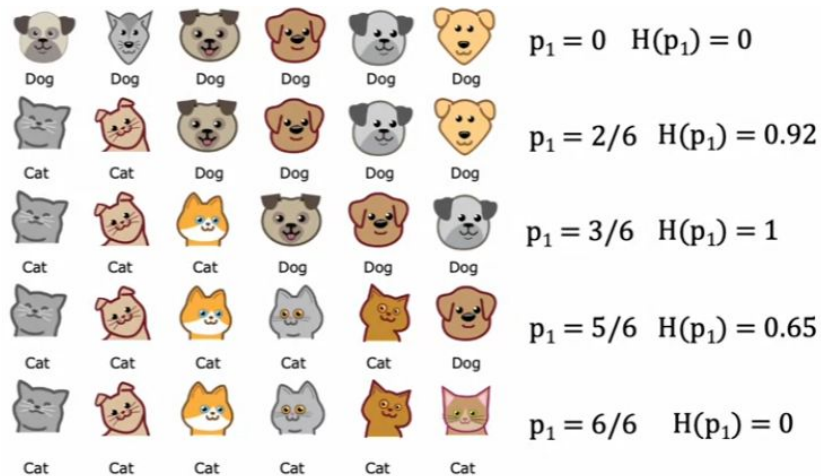
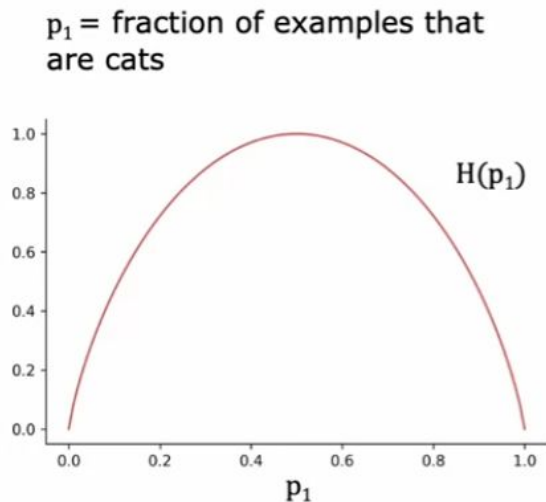


Árvores de Decisão: Questões

- **Pergunta 1: Qual característica escolher para um determinado nó da árvore de decisão?**
 - Aquela que maximiza a taxa de pureza (diminui a entropia).
- **Pergunta 2: Quando finalizar o processo de repartição da árvore em nós folhas?**
 - Quando um nó é composto por amostras pertencentes a apenas uma classe;
 - Quando a repartição de um nó em nós folhas exceder o limiar de profundidade máxima;
 - Quando a taxa de pureza tiver um ganho menor do que um limiar pré-definido;
 - Quando o número de amostras em um nó estiver abaixo de um limiar pré-definido.

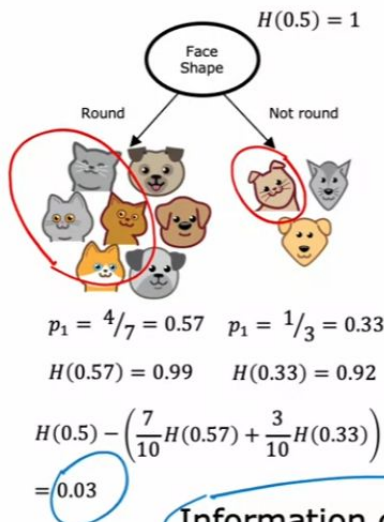
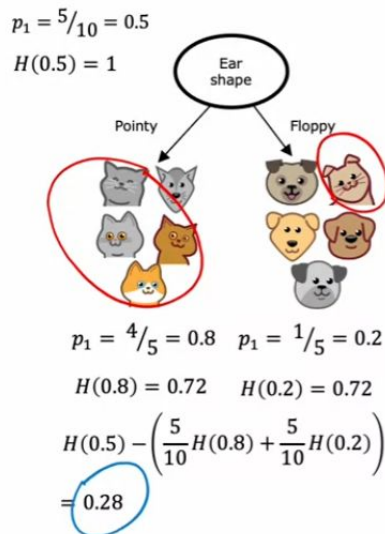
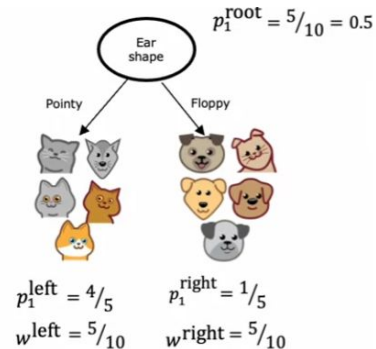
Árvores de Decisão: Entropia

- Calcula a taxa de impureza em um nó da árvore;
- A entropia é calculada a partir da equação $H(p_1) = -p_1 \log_2(p_1) - (1 - p_1) \log_2(1 - p_1)$
 - Considerar $0 \log_2(0) = 0$.

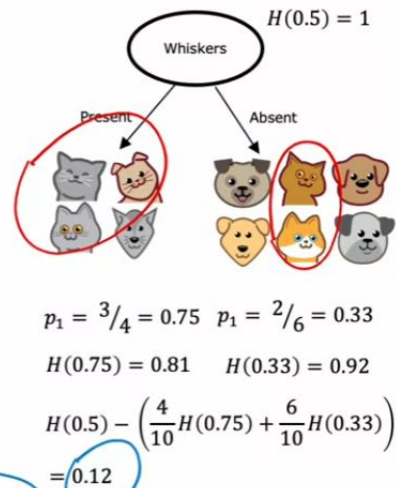


Árvores de Decisão: Repartição do Nó

- O processo de repartir um nó é definido via *Information Gain* (IG);
- O *Information Gain* mede a redução na entropia após uma repartição;
- A Equação para o cálculo do IG é: $H(p_1^{root}) - (w^{left} H(p_1^{left}) + w^{right} H(p_1^{right}))$



Information gain



Árvores de Decisão: Etapas

1. Iniciar com todas as amostras no nó raíz;
2. Calcular o IG para todas os atributos, e selecionar aquele com o maior valor de *Information Gain*;
3. Repartir o dataset de acordo com o atributo selecionado para o nó, e criar os ramos da esquerda e direita da árvore;
4. Continuar o processo de repartição até que algum dos critérios de parada seja satisfeito:
 - Quando um nó estiver homogêneo, *i.e.*, contendo apenas amostras referentes a uma classe;
 - Quando a repartição de um nó resultar em uma profundidade maior que o limiar pré-definido;
 - O IG for menor que um determinado limiar;
 - Quando o número de amostras em um nó estiver abaixo de um limiar.

Exercício 2

- Utilize uma árvore de decisão para prever se a renda salarial de uma pessoa excede \$50k por ano. Exibir a árvore de decisão gerada ao final do processo.



DATA SCIENCE

AULA 4 - Modelos de Decisão

Dúvidas e/ou perguntas?



gabriel.rmachado10@gmail.com



<https://www.linkedin.com/in/machadogabriel>



<https://github.com/UNIFESO-Gabriel/data-science>