

## Resumo Geral de Machine Learning :

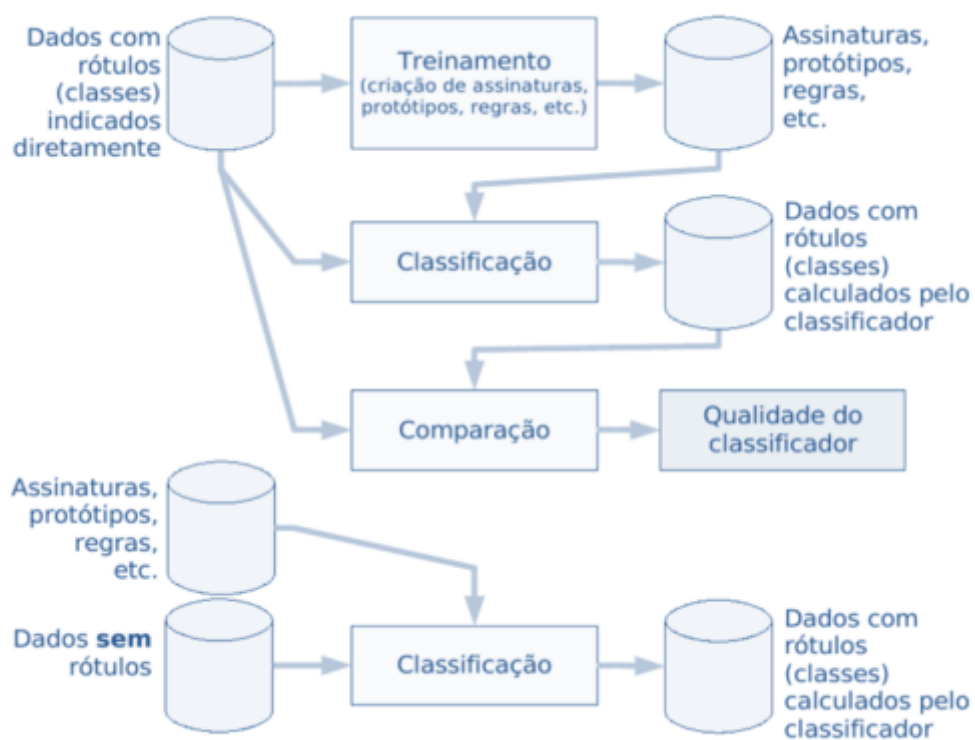
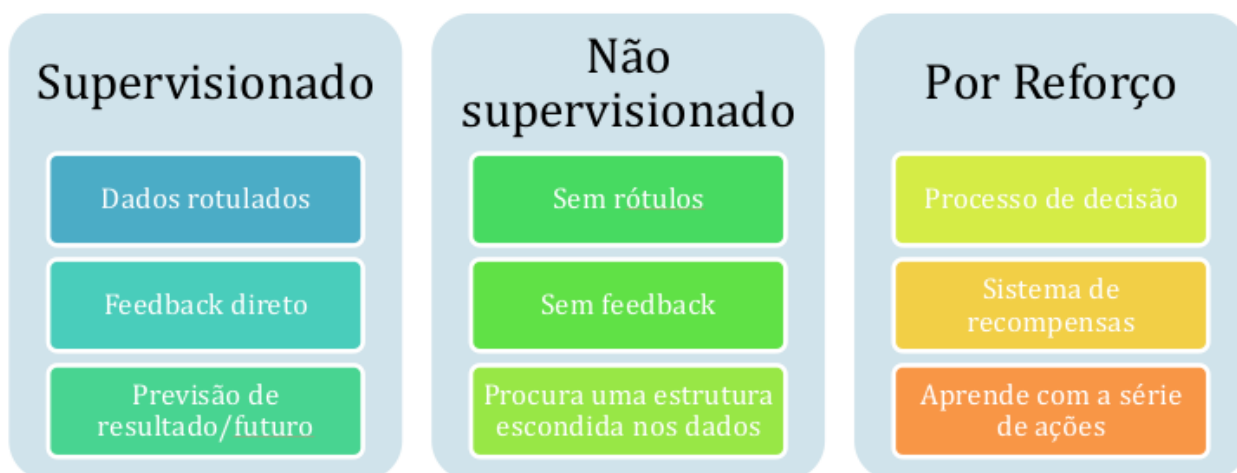


Figura 1 - Processo de construção de um classificador

## Acurácia

Acurácia é a métrica mais simples, ela representa o número de previsões corretas do modelo. É uma ótima métrica para se utilizar quando os dados estão balanceados, vai dar uma visão geral do quanto o modelo está identificando as classes corretamente. Porém, não devemos utilizar a acurácia, quando temos classes desbalanceadas, pode causar uma falsa impressão de estamos obtendo um bom desempenho.

Por exemplo: considere um estudo em que apenas 5% da população apresenta uma determinada doença. Logo, temos um conjunto de dados desbalanceado. Se o modelo escolhido conseguir classificar corretamente todas as pessoas que não têm a doença e errar a classificação de todos os doentes, teremos uma acurácia de 95%, dando uma falsa impressão de que o modelo treinado tem uma ótima previsão. Porém, o modelo não consegue classificar corretamente a classe de interesse. A figura abaixo apresenta a fórmula para o cálculo da acurácia.

		Valor predito $\hat{Y}$	
		Negativo (0)	Positivo (1)
Valor Real	Negativo (0)	VN	FP
	Positivo (1)	FN	VP

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN}$$

Valor Preditivo Negativo (VPN) é a métrica que traz a informação da quantidade de observações classificadas como negativa (0) que realmente são negativas. Ou seja, entre todas as observações prevista como negativa (0), quantas foram identificadas corretamente. Por exemplo: entre os pacientes classificados como não doentes, quantos foram identificados corretamente.

		Valor predito $\hat{Y}$	
		Negativo (0)	Positivo (1)
Valor Real	Negativo (0)	VN	FP
	Positivo (1)	FN	VP

$$\text{VPN} = \frac{VN}{VN + FN}$$

*Precision* ou precisão, também conhecida como Valor Preditivo Positivo (VPP), é a métrica que traz a informação da quantidade de observações classificadas como positiva (1) que realmente são positivas. Ou seja, entre todas as observações identificadas como positivas (1), quantas foram identificadas corretamente. Por exemplo: entre os pacientes classificados como doentes, quantos foram identificados corretamente. A tabela abaixo apresenta a fórmula utilizada no cálculo da previsão.

		Valor predito $\hat{Y}$	
		Negativo (0)	Positivo (1)
Valor Real	Negativo (0)	VN	FP
	Positivo (1)	FN	VP

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

#### Recall (Sensibilidade)

Recall ou Sensibilidade é a proporção dos Verdadeiros Positivos entre todas as observações que realmente são positivas no seu conjunto de dados. Ou seja, entre todas as observações que são positivas quantas o modelo conseguiu identificar como positiva. Representa a capacidade de um modelo em prever a classe positiva. Por exemplo: dentre todos os pacientes doentes, quantos pacientes o modelo conseguiu identificar corretamente.

		Valor predito $\hat{Y}$	
		Negativo (0)	Positivo (1)
Valor Real	Negativo (0)	VN	FP
	Positivo (1)	FN	VP

$$\text{Recall} = \frac{VP}{VP+FN}$$

## Especificidade

Especificidade é a proporção dos Verdadeiros Negativos entre todas as observações que realmente são negativas no seu conjunto de dados. Ou seja, entre todas as observações que são negativas, quantas o modelo conseguiu prever como negativa. Representa a capacidade de um modelo em prever a classe negativa. Por exemplo: dentre todos os pacientes não doentes, quantos foram classificados corretamente.

		Valor predito $\hat{Y}$	
		Negativo (0)	Positivo (1)
Valor Real	Negativo (0)	VN	FP
	Positivo (1)	FN	VP

$$\text{Especificidade} = \frac{VN}{VN+FP}$$

## F1- Score

F1-Score é a média harmônica entre o recall e a precisão (*precision*). Utilizada quando temos classes desbalanceada.

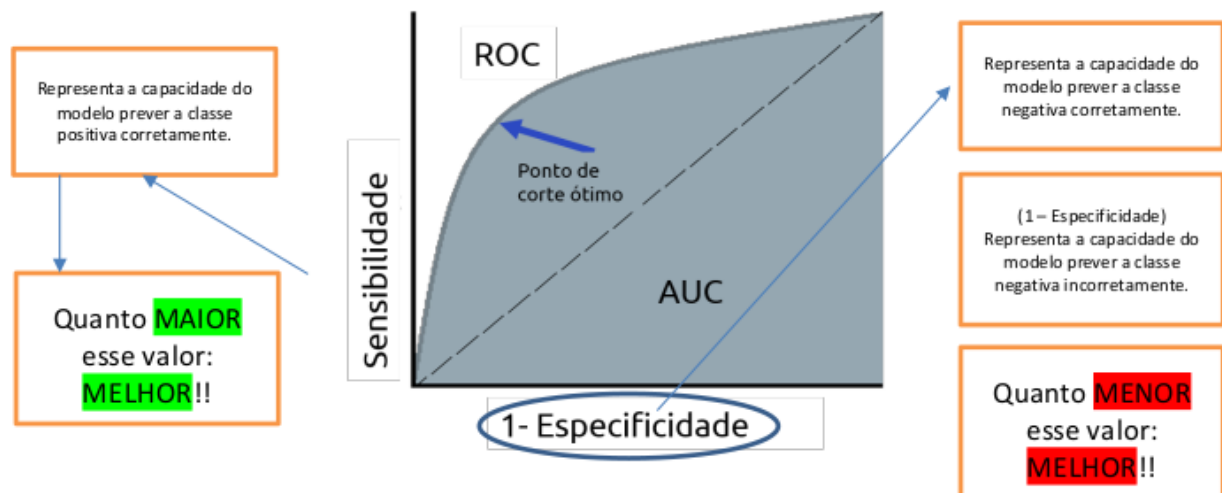
$$\text{F1-Score} = \frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}}$$

## Área sob a Curva ROC

A curva ROC (*Receiver Operating Characteristic Curve*) é a curva gerada pela taxa de verdadeiros positivos (sensibilidade) e pela taxa de falsos positivos ( $1 - \text{especificidade}$ ) para diferentes pontos de cortes ( $c$ ). A curva ROC oferece uma visão geral de um classificador e pode ser utilizada para encontrar pontos de corte ideais. O corte que deixa a curva mais próxima do vértice  $(0, 1)$  maximiza a sensibilidade conjuntamente com a especificidade.

Uma medida decorrente da curva ROC é o AUC (*Area Under the Curve*), que nada mais é que a área abaixo da curva. O AUC varia entre 0 e 1 e quanto maior o AUC melhor o modelo.

Utilizada quando temos classes desbalanceada e sua principal vantagem é poder escolher o melhor ponto de corte para otimizar o desempenho do modelo.



Todas essas métricas variam no intervalo  $[0,1]$  e quanto mais próximos de 1, melhor é o modelo.

## Algoritmos de Classificação

- |                       |                                |
|-----------------------|--------------------------------|
| ❖ Regressão Logística | ❖ Máquinas de Vetor de Suporte |
| ❖ Naive Bayes         | ❖ XGBoost                      |
| ❖ Árvore de decisão   | ❖ LightGBM                     |
| ❖ Random Forest       | ❖ CatBoost                     |
| ❖ KNN                 | ❖ Redes Neurais Artificiais    |

### 1.4.2 Algoritmos Supervisionados

Neste tipo temos os atributos alvo rotulados e do ponto de vista da máquina, esse processo é uma rotina de "conectar os pontos" ou achar similaridades entre as observações e o que ela representa. Para "alimentar" o algoritmo determinamos que tipo de resultado é desejado ("sim/não", "verdadeiro/falso", a projeção do valor das vendas, a perda líquida de crédito ou o preço da habitação).

**Regressão linear:** muito utilizado para prever resultados numéricos contínuos, como preços de casas ou ações, umidade ou temperatura de um local, crescimento populacional.

**Regressão logística:** É um classificador popular utilizado especialmente no setor de crédito para prever inadimplências de empréstimos.

**k-Nearest Neighbors:** KNN é um algoritmo usado para classificar as observações em duas ou mais categorias (*cluster*) amplamente usado na separação como preços de casas em grupos, por exemplo com base em preço, área, quartos e toda uma gama de outros preditores.

**Support Vector Machines:** SVM é um classificador popular utilizado na detecção de imagens e faces, além de aplicativos como reconhecimento de manuscrito.

---

<sup>3</sup>Assim como análise da linguagem, chamada de NLP - *Natural Language Processing*

**Tree-Based Algorithms:** Algoritmos baseados em árvores, como *Random Florest* (florestas aleatórias) ou *Decision Tree* (árvores de decisão), são usados para resolver problemas de classificação e regressão.

**Naive Bayes:** Utiliza um modelo matemático de probabilidade para resolver problemas de classificação.

### 1.4.3 Algoritmos de Aprendizagem por Reforço

Tratam de situações onde a máquina começa a aprender por tentativa e erro ao atuar sobre um ambiente dinâmico. Desta maneira, não é necessário novos exemplos ou um modelo a respeito da tarefa a ser executada: a única fonte de aprendizado é a própria experiência do agente, cujo objetivo formal é adquirir uma política de ações que maximize seu desempenho geral.

**Q-Learning:** é um algoritmo de aprendizado baseado em valores. Esses tipos atualizam uma função com base em uma equação (particularmente neste caso de *Bellman*) matemática, ou então, com base em políticas, estima a função de valor para uma política gananciosa obtida a partir do último aprimoramento. Q-learning é um algoritmo de políticas. Significa que aprende o valor ideal, independentemente das ações do agente. Por outro lado, um aprendiz de política aprende o valor que está sendo executada pelo agente, incluindo as etapas de exploração e encontrará uma política ideal, leva em consideração a exploração inerente dessa.

**Temporal Difference:** TD é um agente que aprende por meio de episódios sem conhecimento prévio desse ambiente. Isso significa que a diferença temporal adota uma abordagem de aprendizado sem modelo ou supervisão. Podemos considerar isso como uma tentativa e erro.

**Monte-Carlo Tree Search:** MCTS é um método geralmente usado nos jogos para prever o caminho (movimentos) que a política deve seguir para alcançar a solução final vencedora. Jogos como Cubo de Rubik, Sudoku, Xadrez, Go, ou um simples Jogo da Velha têm muitas propriedades em comuns que levam ao aumento exponencial do número de possíveis ações que podem ser executadas. Esses passos aumentam exponencialmente à medida que o jogo avança. Idealmente, podemos prever todos os movimentos possíveis e seus resultados que podem ocorrer no futuro e assim aumentarmos a chance de ganhar.

**Asynchronous Advantage Actor-Critic:** A3C é um dos algoritmos mais recentes a serem desenvolvidos no campo de Reforço Profundo. Foi desenvolvido pelo *DeepMind* do Google e implementa um treinamento no qual vários *workers*, em ambiente paralelo, atualizam independentemente uma função de valor global - portanto "assíncrona". Um dos principais benefícios de ter atores assíncronos é a exploração eficaz e eficiente do espaço de estados.