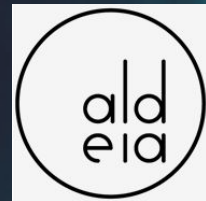


# Data Science

**Charles Adriano dos Santos**

**Turma Março/2020**



# Agenda

1 – Agenda

2 – Namorando Dados (SQL)

3 – Welcome to Python

4 – Intro Machine Learning

# Manhã

---

Horário	Assunto
09:00	Namorando Dados - SQL - Análise Exploratório Desafio Curso
10:30	Welcome to Python
12:00	Almoço

# Tarde

---

Horário	Assunto
13:00	Welcome Python - Continuação
15:00	Introdução a Machine Learning
17:30	Dúvidas Homework ETL

# Nos Episódios Anteriores...



Profissão Data Science

Desafio Agro XP

ETL

Modelagem de Dados

Banco de Dados

Queries SQL

Conceitos Estatísticos

R

# Agenda

1 – Agenda

2 – Namorando Dados (SQL)

3 – Welcome to Python

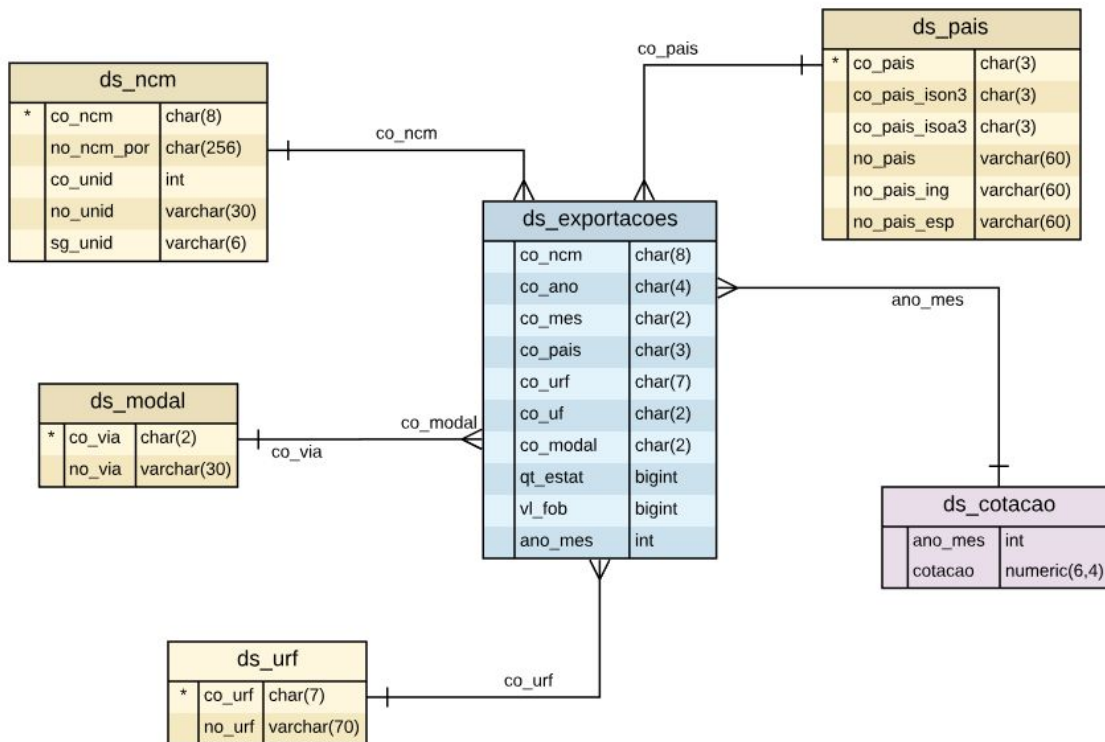
4 – Intro Machine Learning

# O Trabalho do Cientista de Dados > Desafio Curso

1. Definição do problema e levantamento de perguntas a serem respondidas ✓
2. Planejamento do processo de Data Science ✓
3. Coleta de dados ✓
4. Processamento e limpeza dos dados ←
5. Armazenamento dos dados ✓
6. Análise de dados ←
7. Construção e validação de algoritmos e modelos
8. Data Visualization
9. Disseminação da informação
10. Colocar modelo em produção



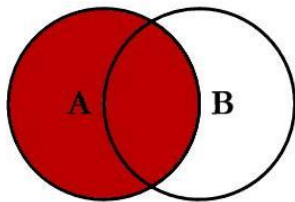
# Desafio – Modelo de Dados



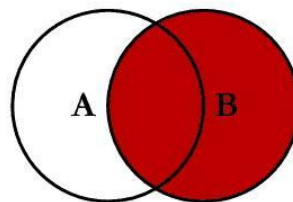


# Namorando os Dados (Queries SQL)

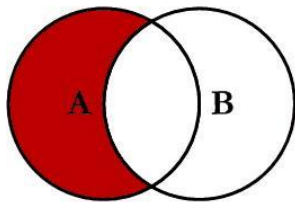
## SQL JOINS



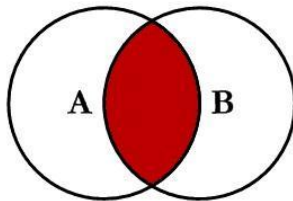
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



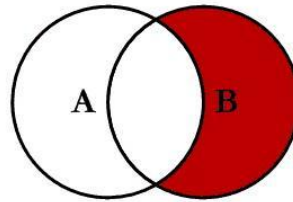
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



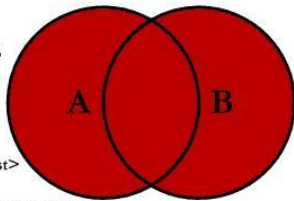
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



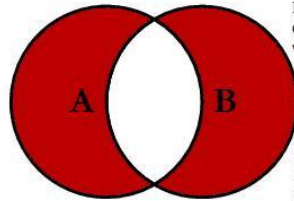
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```

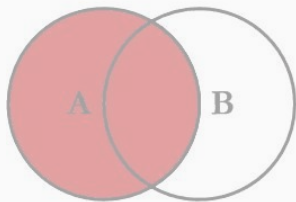


```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

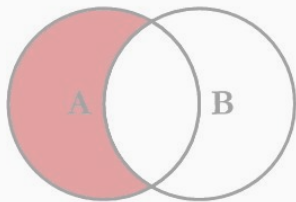
# Namorando os Dados (Queries SQL)

## SQL JOINS

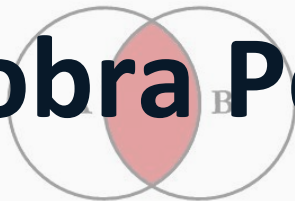
**Mãos à obra Pessoal!!!**



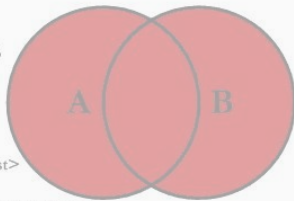
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



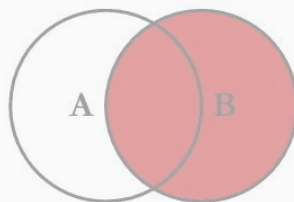
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



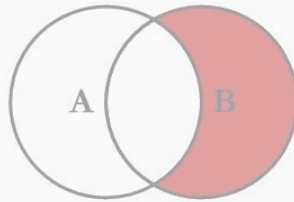
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



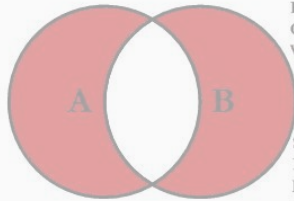
```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

# Agenda

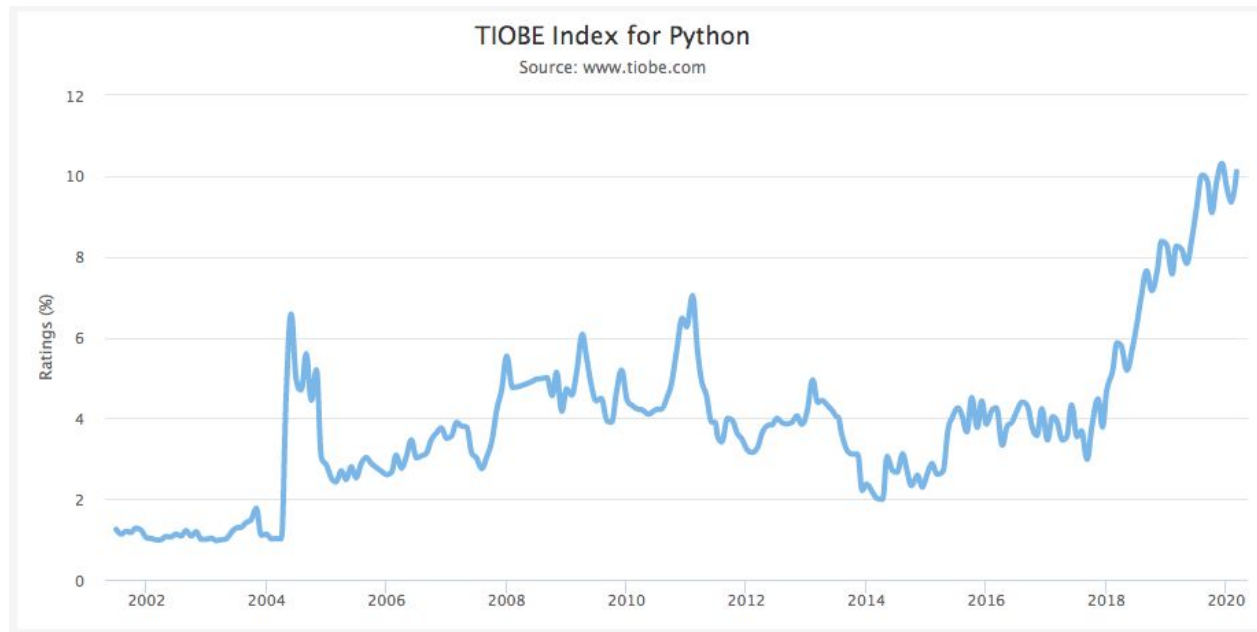
1 – Agenda

2 – Namorando Dados (SQL)

**3 – Welcome to Python**

4 – Intro Machine Learning

# Python



Fonte: <https://www.tiobe.com/tiobe-index/>



# Python – Me Dê Motivos

**Linguagem em forte ascensão** ([2ª linguagem mais amada](#) Stack Overflow)

**Curva de Aprendizado Baixa**

**Free** (Licença GLP)



**Estável** (1ª versão 1991) / Última Versão 3.8.0 (Out/2019)

**Multiplataforma** (Windows, Linux, MacOS e etc.)

**Comunidade**

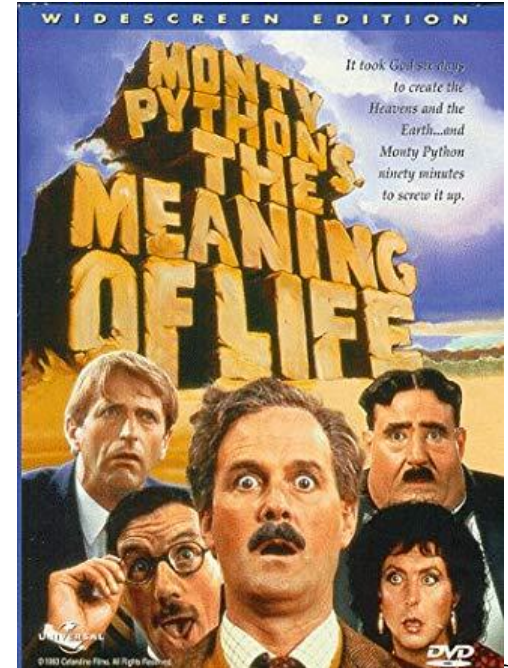
**Data Science** → Ótimos pacotes

# Python – História

Pai do Python →  
[Guido van Rossum](#)



A inspiração do nome →



# Python – História

Versão 2 (2.7) x Versão 3 (3.7.1)

3/4 Paradigmas de Programação:

- **Programação Imperativa** → Ações/Comandos de um programa
- **Programação Orientada o Objeto** → Abstração, Encapsulamento, Herança e Polimorfismo
- **Programação Funcional** → Soluções como problemas de funções



Interpretada

# Python – Hands-on





# Python – Versão 2 x Versão 3

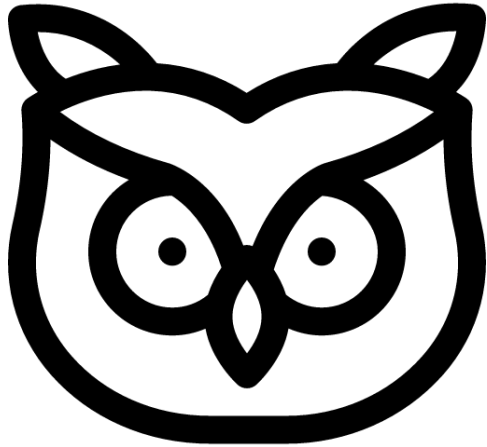


Python 2.X	Python 3.X
There's ASCII <code>str</code> type and <code>unicode</code> type, but no separate type to handle bytes of data	All strings ( <code>str</code> ) are Unicode strings; two byte classes are introduced: <code>bytes</code> and <code>bytearray</code>
Two types of integers: C-based integers ( <code>int</code> ) and Python long integer ( <code>long</code> )	All integers are long but referred to by the <code>int</code> type
Return type of division is <code>int</code> if operands are integers: <code>5 / 4</code> gives 1; <code>4 / 2</code> gives 2	Return type of division is <code>float</code> even if operands or result are integers: <code>5 / 4</code> gives 1.25; <code>4 / 2</code> gives 2.0
<code>round(16.5)</code> returns a float of value 16.0	<code>round(16.5)</code> returns an int of value 16
Unorderable types can be compared	Comparison of unorderable types raises a <code>TypeError</code>
<code>print</code> is a statement: <code>print "Hello World!"</code>	<code>print()</code> is a built-in function: <code>print("Hello World!")</code>
<code>range()</code> returns a list of numbers while <code>xrange()</code> returns an object for lazy evaluation	<code>range()</code> returns an object for lazy evaluation similar to Python 2 <code>xrange()</code> ; and <code>range()</code> method <code>__contains__</code> speeds up lookups
Functions/methods <code>map()</code> , <code>filter()</code> , <code>zip()</code> , <code>dict.items()</code> , <code>dict.keys()</code> , <code>dict.values()</code> return lists	These function/methods return objects for lazy evaluation
<code>raw_input()</code> returns input as strand <code>input()</code> evaluates the input as a Python expression	<code>input()</code> will return a string similar to Python 2 <code>raw_input()</code>
Raising exceptions: <code>raise IOError("file error")</code> or <code>raise IOError, "file error"</code>	Raising exceptions: <code>raise IOError("file error")</code>
Handling exceptions: <code>except NameError, err:</code> or <code>except (TypeError, NameError), err:</code>	Handling exceptions: <code>except NameError as err</code> or <code>except (TypeError, NameError) as err</code>
On generators, a method or function call: <code>g.next()</code> or <code>next(g)</code>	On generators, only a function call: <code>next(g)</code>
Loop variables in a comprehension leak to global namespace	Loop variables are limited in scope to the comprehension

Fonte: <https://devopedia.org/python-2-vs-3>



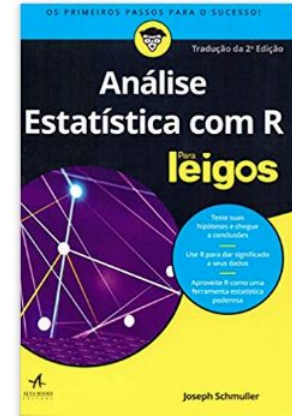
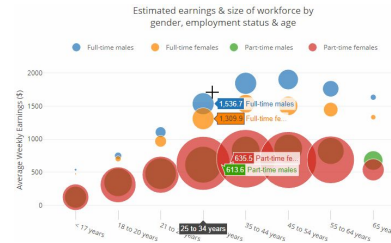
# Quero Saber Mais...



## Data Manipulation for Machine Learning with Pandas

Os 35 Melhores Cursos de Python gratuitos disponíveis pra você

**Towards Data Science**  
Sharing concepts, ideas, and codes



# Agenda

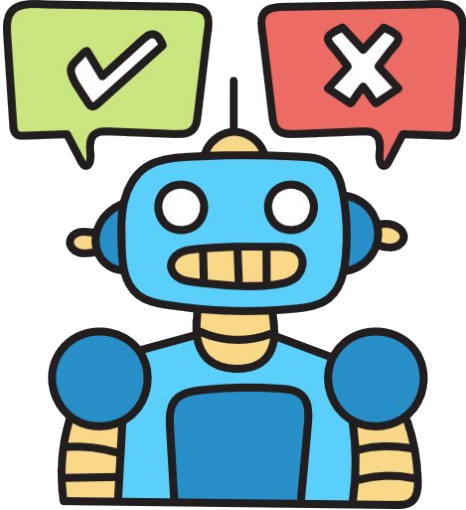
1 – Agenda

2 – Namorando Dados (SQL)

3 – Welcome to Python

**4 – Intro Machine Learning**

# Machine Learning - Conceito

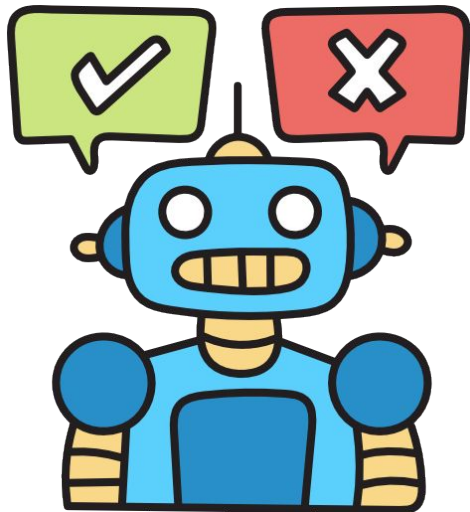


A máquina, através de algoritmos, obter padrões sobre características extraídas dos dados para, com um modelo gerado/criados, classificar as observações futuras de novos dados.

No conceito cada vez menos intervenção humana (conceito).

Pré-processamento e análise dos dados, além de realizar “grid” de valores para treinamento obterem maior acurácia (na prática)

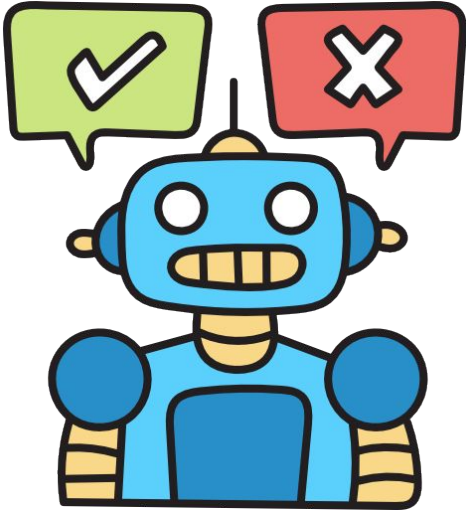
# Machine Learning - História



**1950 - IA: Computadores com habilidade de “pensar”**  
-Teste de Turing. Em 2014 chatbot enganou 10/30 juízes



# Machine Learning - História

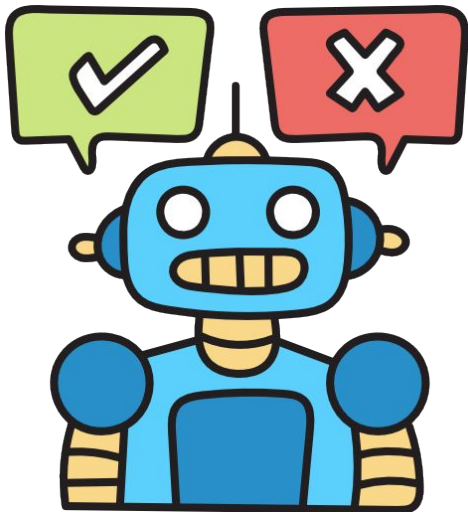


## 1959 - ML: Aprender a partir dos dados - Arthur Samuel

Aprender com a experiência que existe intrínseca aos dados.

Algoritmos de aprendizado de máquina analisam as correlações entre os atributos (variáveis) de um sistema (base de dados) a partir de dados amostrais (base de treinamento)

# Machine Learning - História



## 2012: DS – Entender os Dados

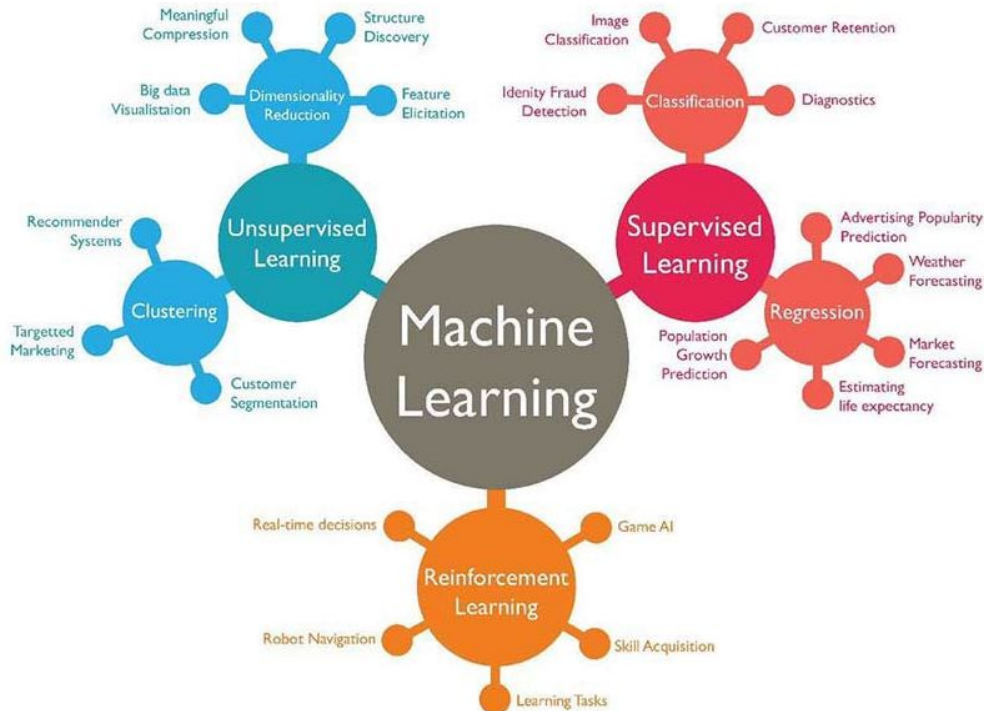
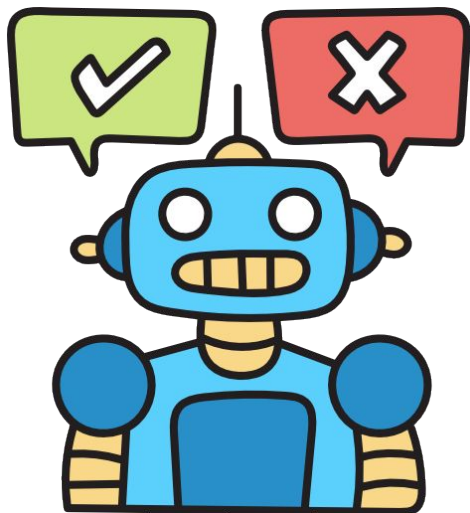
Ciência de dados utilizando probabilidade, estatística álgebra linear e computação.

Conhecimentos de IA e ML

*“É a ciência (e arte) de programar computadores de tal forma que eles aprendam a partir de dados”*  
(Aurélien Géron, 2017)

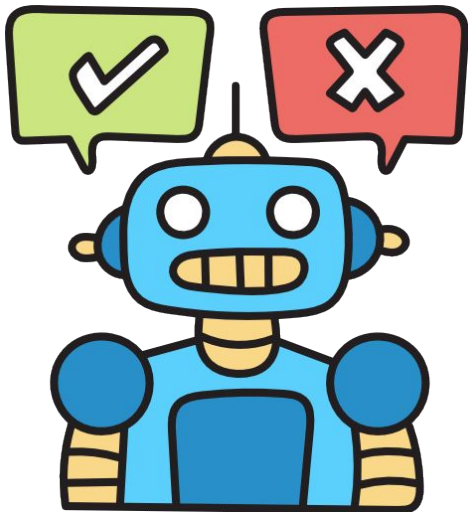


# Machine Learning – Tipos de Aprendizizado





# Machine Learning – Tipo de Aprendizado

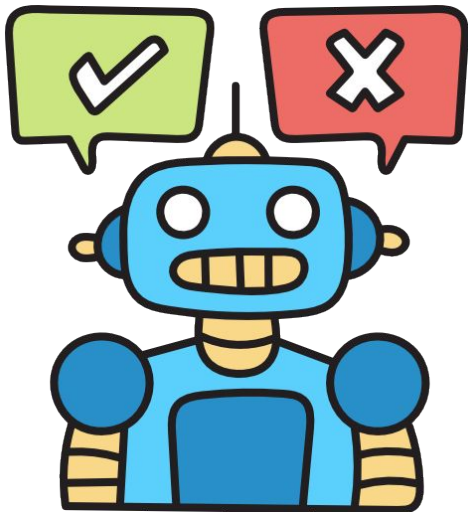


**Supervisionado** → rotulado com saídas esperadas. Modelo gera ao entrar com conjunto de características uma saída rotulada (**Classificação**) ou um valor futuro (**Predição**). Ex: Nosso desafio AgroXP.

**Não Supervisionado** → Não existe rótulo prévio. Analisa a rede de relacionamento entre os dados para agrupá-los por características similares. Ex: Categorização de Clientes

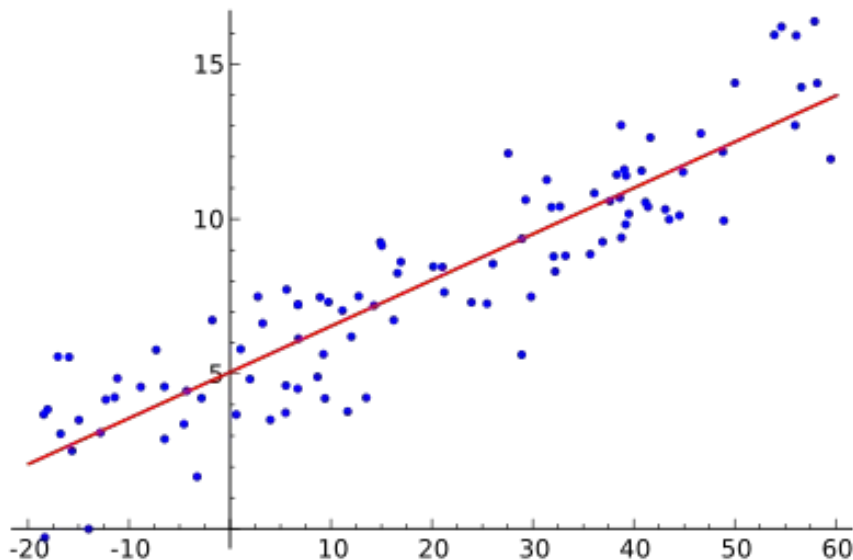
**Reforço** → Maximizar o resultado. Baseado em recompensa / punição. Com isso algoritmo encontrar a “política” que mapeia os dados. Ex: Personagens Jogos

# Machine Learning – Exemplos Algoritmos

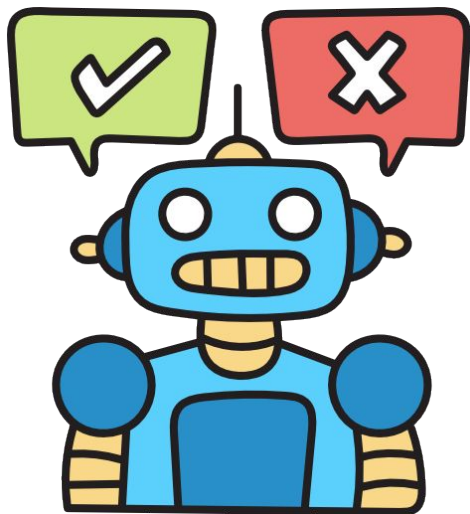


## Regressão Linear (Supervisionado – Predição)

Simples... Busca uma reta para se ajustar aos dados.  
Problemas de relação linear.

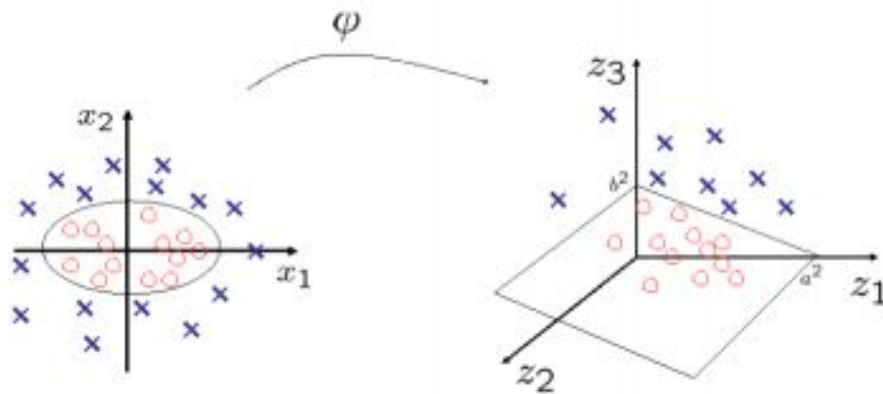


# Machine Learning – Exemplos Algoritmos

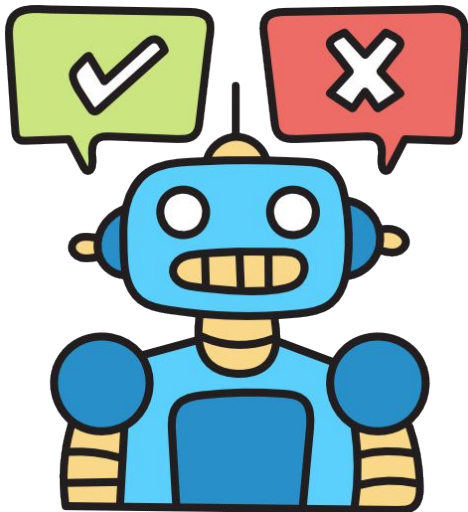


## SVM - Support Vector Machine (**Supervisionado – Classificação**) – Vapnik (1963)

Distância das amostras da linha superfície de separação. Consegue trabalhar com dados não lineares com a premissa de que em alguma dimensão os dados terão linearidade.

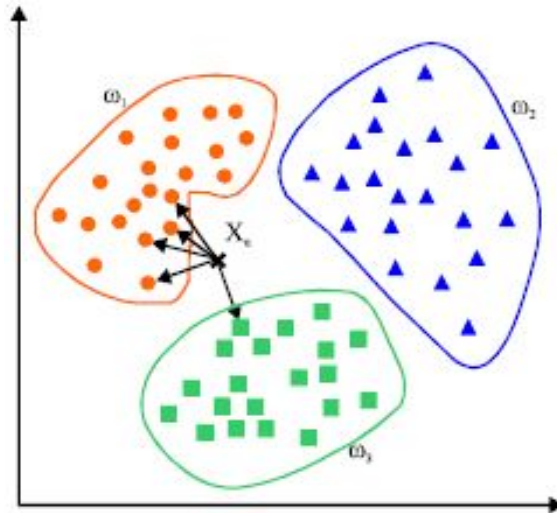


# Machine Learning – Exemplos Algoritmos

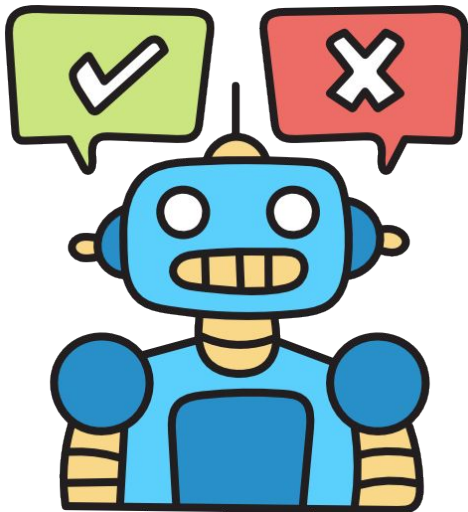


## KNN – K-Nearest Neighbors (Supervisionado – Classificação)

Baseado em encontrar o valor de K que consiga através de funções básicas de distância Euclidiana encontrar a melhor superfície de separação

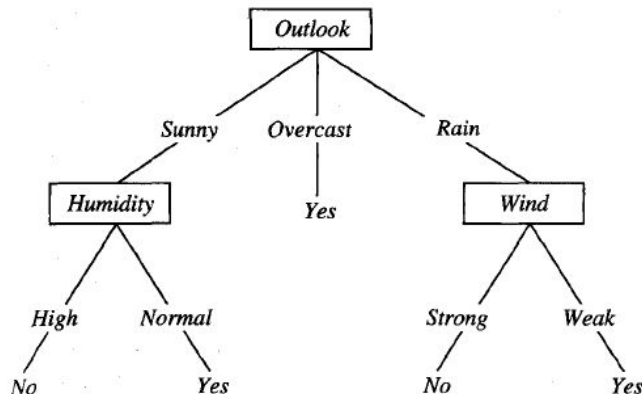


# Machine Learning – Exemplos Algoritmos

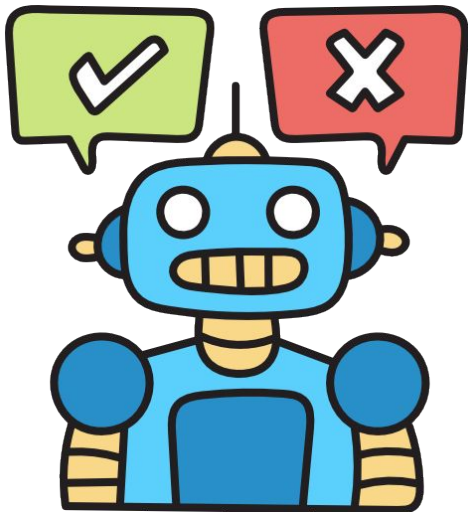


## Árvore de Decisão (Supervisionado – Classificação)

De fácil explicação do modelo obtido, este algoritmo utiliza a categorização utilizando técnicas referente a Ganho de Informação dos atributos (o quanto a variável sozinha classifica os exemplos de treinamento). Pode ser utilizado para dados numérico ou simbólicos.



# Machine Learning – Exemplos Algoritmos



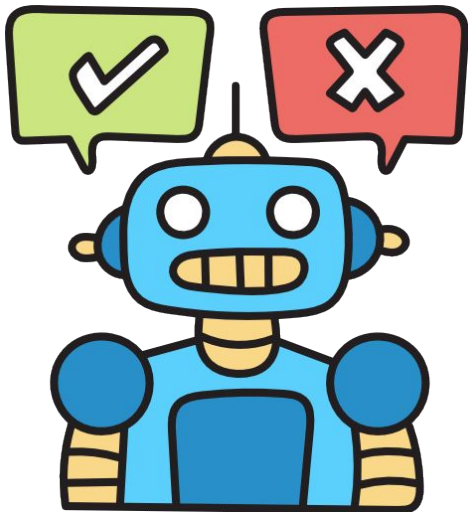
## K-Means – (Não Supervisionado)

Forma clusters que contêm pontos homogêneos aos dados.



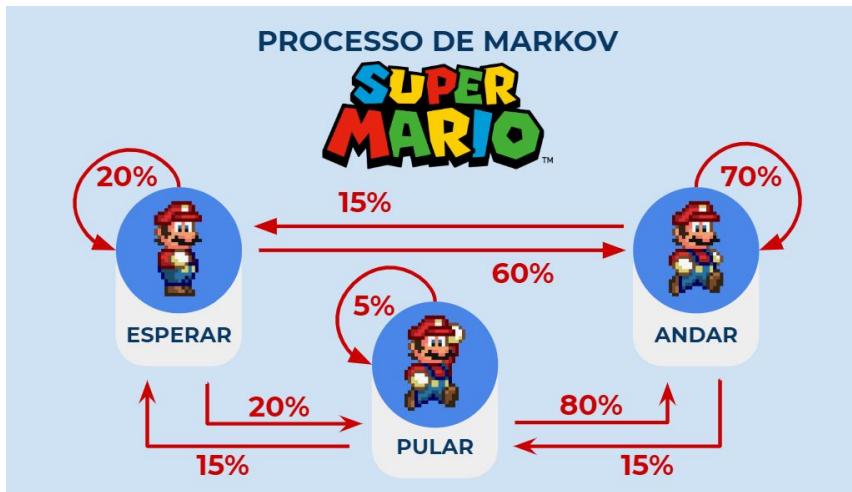


# Machine Learning – Exemplos Algoritmos

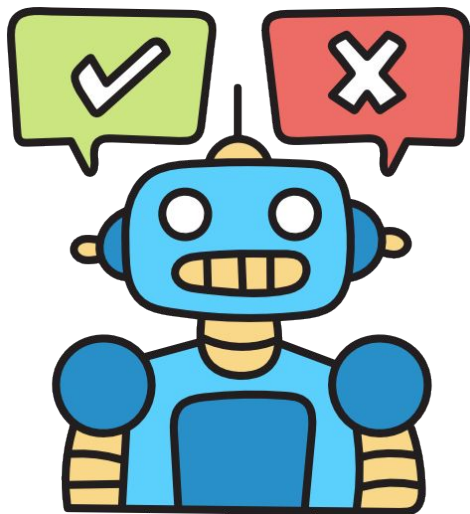


## Cadeia de Markov (Reforço)

Processo estocástico (futuro  $\rightarrow$  estado atual). Com base na cadeia e suas probabilidades o algoritmo toma uma decisão e, se houver recompensa, reforça a decisão tomada. Se houver uma punição rechaça.



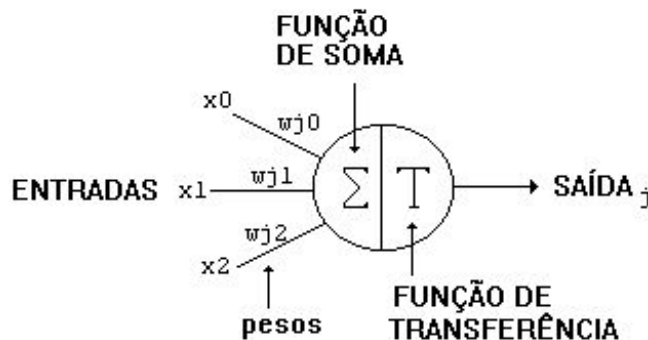
# Machine Learning – Exemplos Algoritmos



## Redes Neurais (Supervisionado – Classificação)

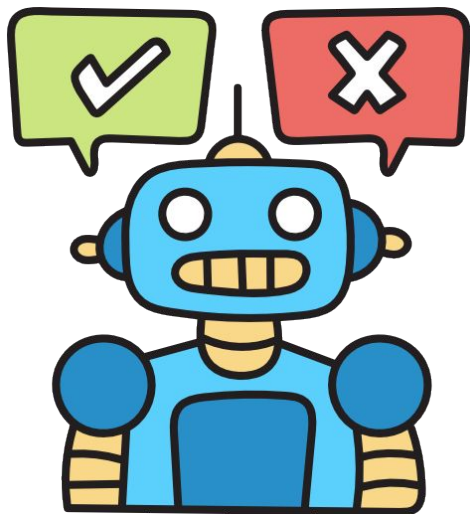
Baseado no conceito matemático e computacional (1943) que visa descrever o modelo artificial para um neurônio biológico.

Responde “ligando/desligando” os vários neurônios interligada e com isso classifica as características de entrada no rótulo predito pelo modelo.



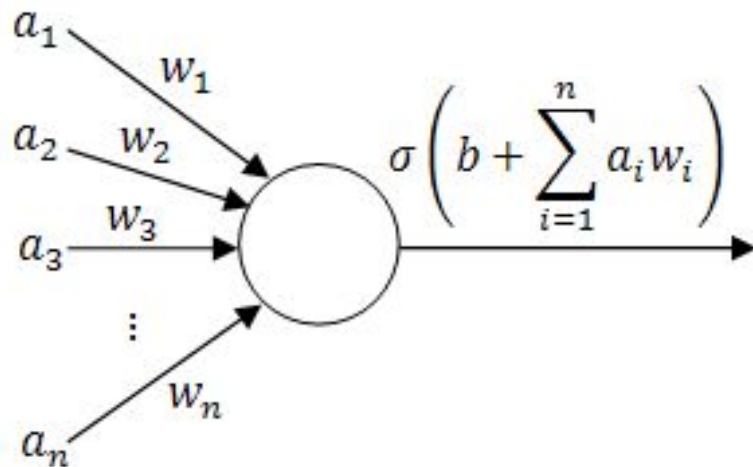


# Machine Learning – Exemplos Algoritmos

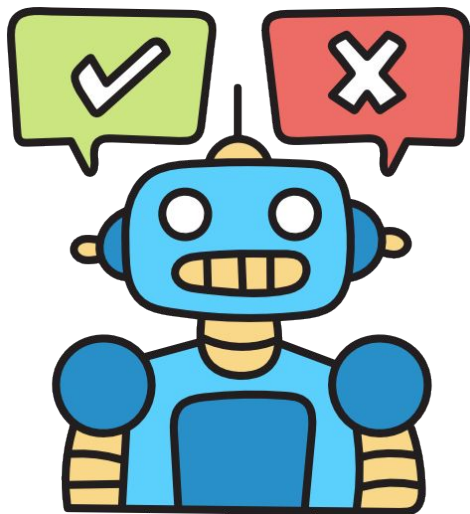


## Redes Neurais (Supervisionado – Classificação)

Perceptron → Tipo básico de rede neural. Demonstrou em 1957 a possibilidade de simulação de um neurônio biológico.

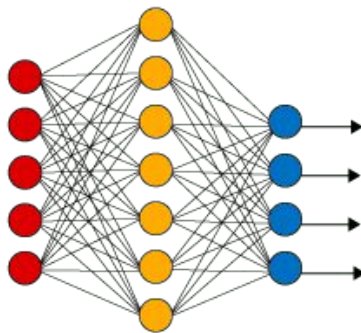


# Machine Learning – Exemplos Algoritmos

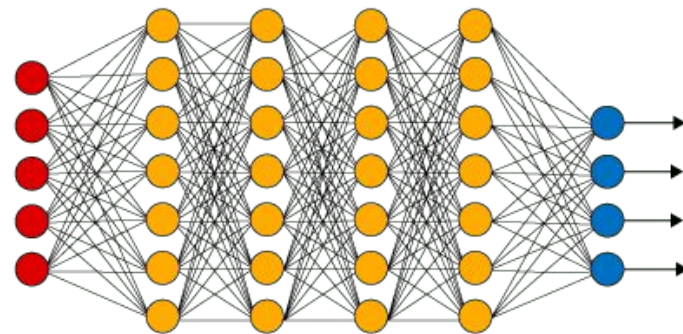


## Redes Neurais (Supervisionado – Classificação)

Simple Neural Network



Deep Learning Neural Network

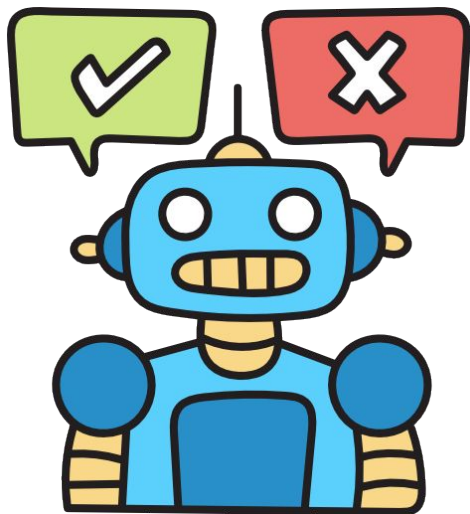


● Input Layer

● Hidden Layer

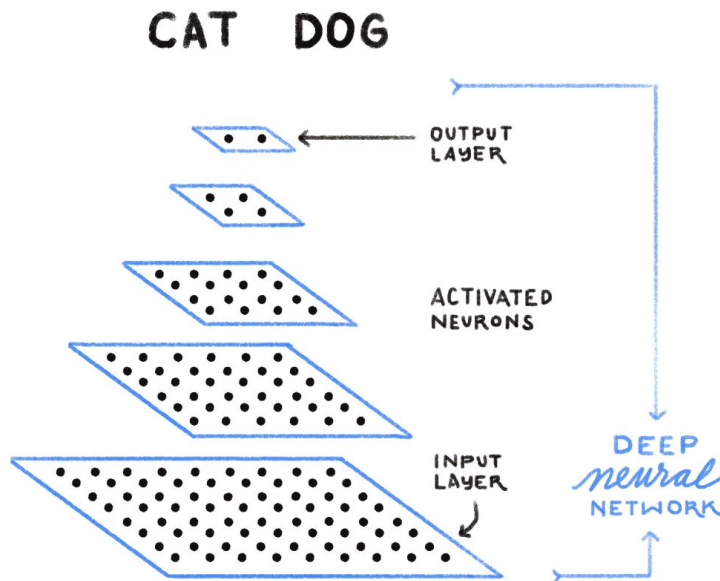
● Output Layer

# Machine Learning – Exemplos Algoritmos

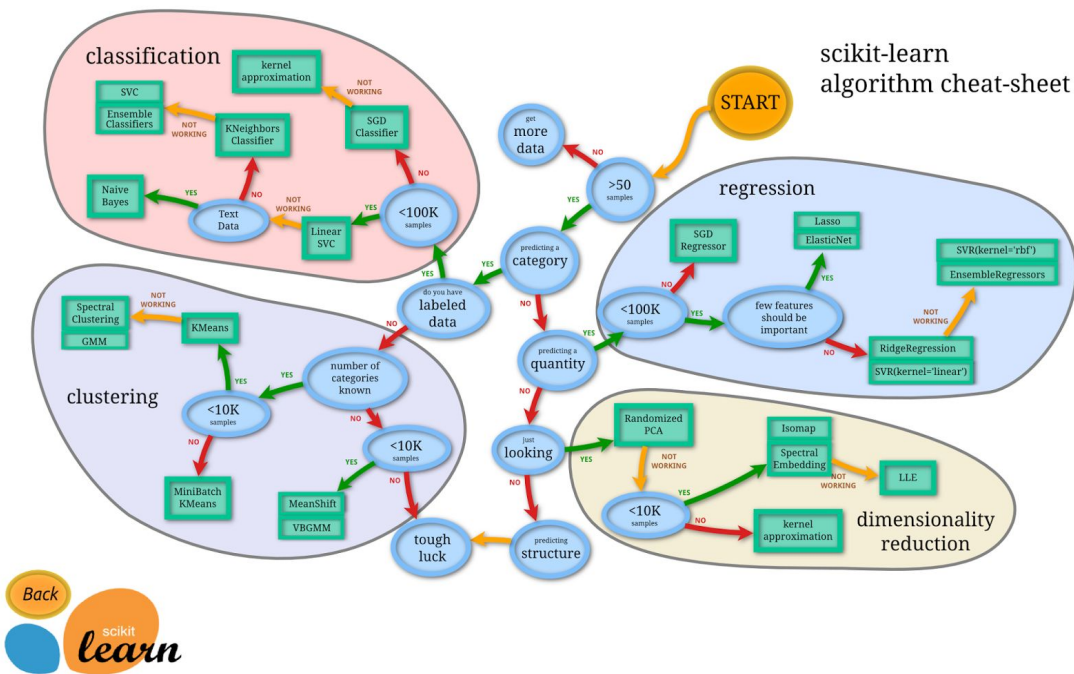
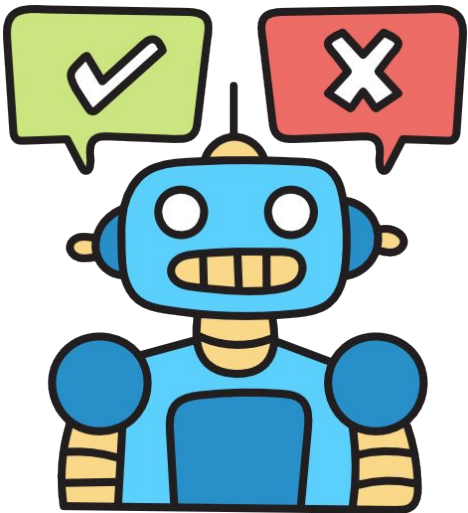


## Redes Neurais (Supervisionado – Classificação)

IS THIS A  
CAT or DOG?



## Algoritmo x Características Datos



**Fonte:**[https://scikit-learn.org/stable/tutorial/machine\\_learning\\_map/](https://scikit-learn.org/stable/tutorial/machine_learning_map/)



# Usando Machine Learning

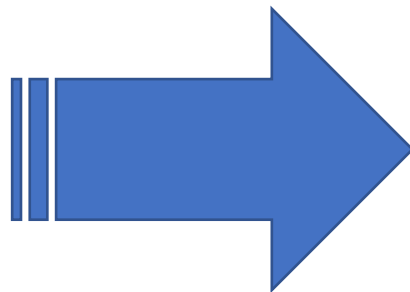
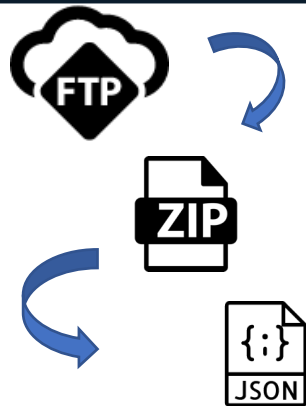


# O Trabalho do Cientista de Dados > Desafio Pessoal

1. Definição do problema e levantamento de perguntas a serem respondidas
2. Planejamento do processo de Data Science
3. Coleta de dados
4. Processamento e limpeza dos dados
5. Armazenamento dos dados
6. Análise de dados
7. Construção e validação de algoritmos e modelos
8. Data Visualization
9. Disseminação da informação
10. Colocar modelo em produção



# ETL na Prática – Homework



JSON → Com remuneração variável por funcionário

ftp server: <ftp.drivehq.com>

User: datascienceandbigdata@gmail.com

Password: ds2019FTP

Diretório: GroupWrite

Arquivo: remunera.zip

Planilha Excel com:

Matrícula

Nome Funcionário

Cargo

Valor Hora

Último Dia e Hora Marcação Ponto

Total Remuneração Variável



# Obrigado!

📄 Charles Adriano dos Santos

✉️ [charles.a.santos@caelis.it](mailto:charles.a.santos@caelis.it)

🌐 chadri

📞 41 99144 6663

📄 Rafael Roberto Dias

✉️ [rafael.dias@madeiramadeira.com.br](mailto:rafael.dias@madeiramadeira.com.br)

🌐 [rafael-roberto-dias-00b39123](#)

📞 41 99672 7170