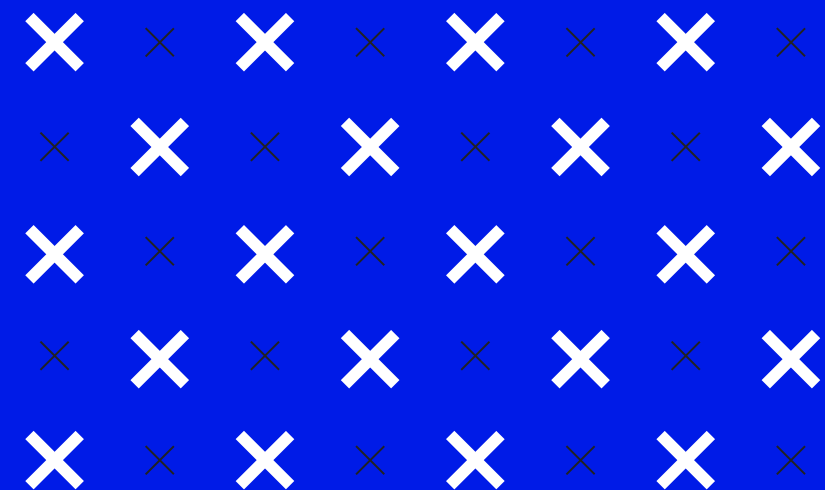




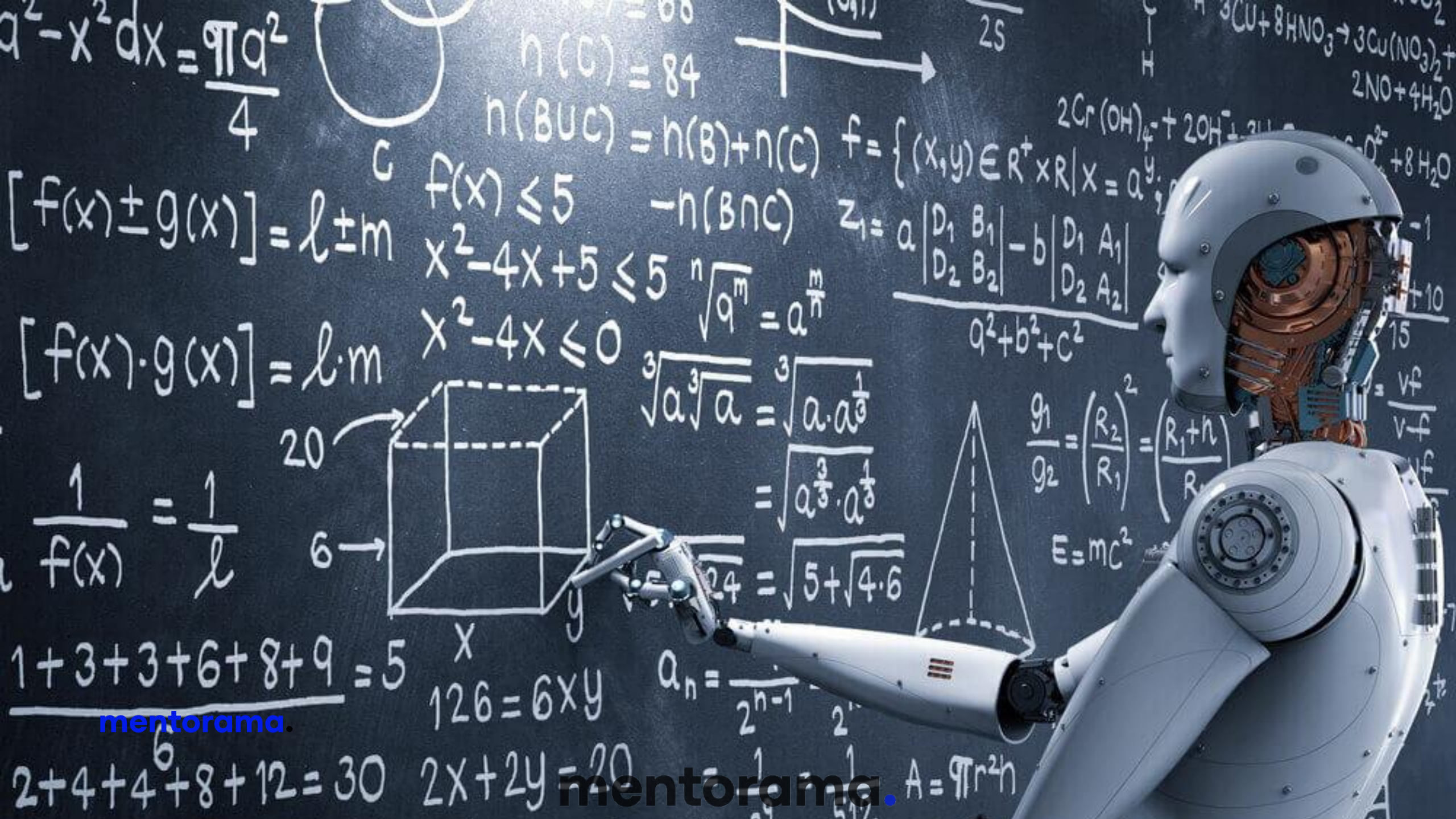
# MACHINE LEARNING



**mentorama.**

@prof.felipeassuncao

**mentorama**



$$\int a^2 - x^2 dx = \frac{\pi a^2}{4}$$

$$[f(x) \pm g(x)] = l \pm m$$

$$[f(x) \cdot g(x)] = l \cdot m$$

$$\frac{1}{f(x)} = \frac{1}{l}$$

$$1 + 3 + 3 + 6 + 8 + 9 = 5$$

$$2 + 4 + 4 + 8 + 12 = 30$$

$$h(C) = 84$$

$$h(BUC) = h(B) + h(C)$$

$$f(x) \leq 5$$

$$x^2 - 4x + 5 \leq 5$$

$$x^2 - 4x \leq 0$$



$$126 = 6 \times y$$

$$2x + 2y = 20$$

$$n(B \cap C)$$

$$f = \{(x, y) \in \mathbb{R}^+ \times \mathbb{R} \mid x = a^y\}$$

$$z_1 = a \frac{\begin{vmatrix} D_1 & B_1 \\ D_2 & B_2 \end{vmatrix} - b \begin{vmatrix} D_1 & A_1 \\ D_2 & A_2 \end{vmatrix}}{a^2 + b^2 + c^2}$$

$$\sqrt[n]{a^m} = a^{\frac{m}{n}}$$

$$\sqrt[3]{a^3 \sqrt{a}} = \sqrt[3]{a \cdot a^{\frac{1}{3}}}$$

$$= \sqrt[3]{a^{\frac{3}{3}} \cdot a^{\frac{1}{3}}}$$

$$= \sqrt[3]{a^{\frac{4}{3}}}$$

$$= \sqrt[3]{5 + \sqrt{4 \cdot 6}}$$

$$a_n = \frac{2^{n-1}}{2^n}$$

$$A = \pi r^2 h$$

$$\frac{25}{2}$$

$$2Cr(OH)_4 + 2OH^- + 3H_2O$$

$$3Cu + 8HNO_3 \rightarrow 3Cu(NO_3)_2 + 2NO + 4H_2O$$

$$E = mc^2$$

$$\frac{g_1}{g_2} = \left(\frac{R_2}{R_1}\right)^2 = \left(\frac{R_1 + h}{R_1}\right)^2$$

$$v = \frac{v_f}{v - f}$$

$$v = \frac{v_f}{v - f}$$

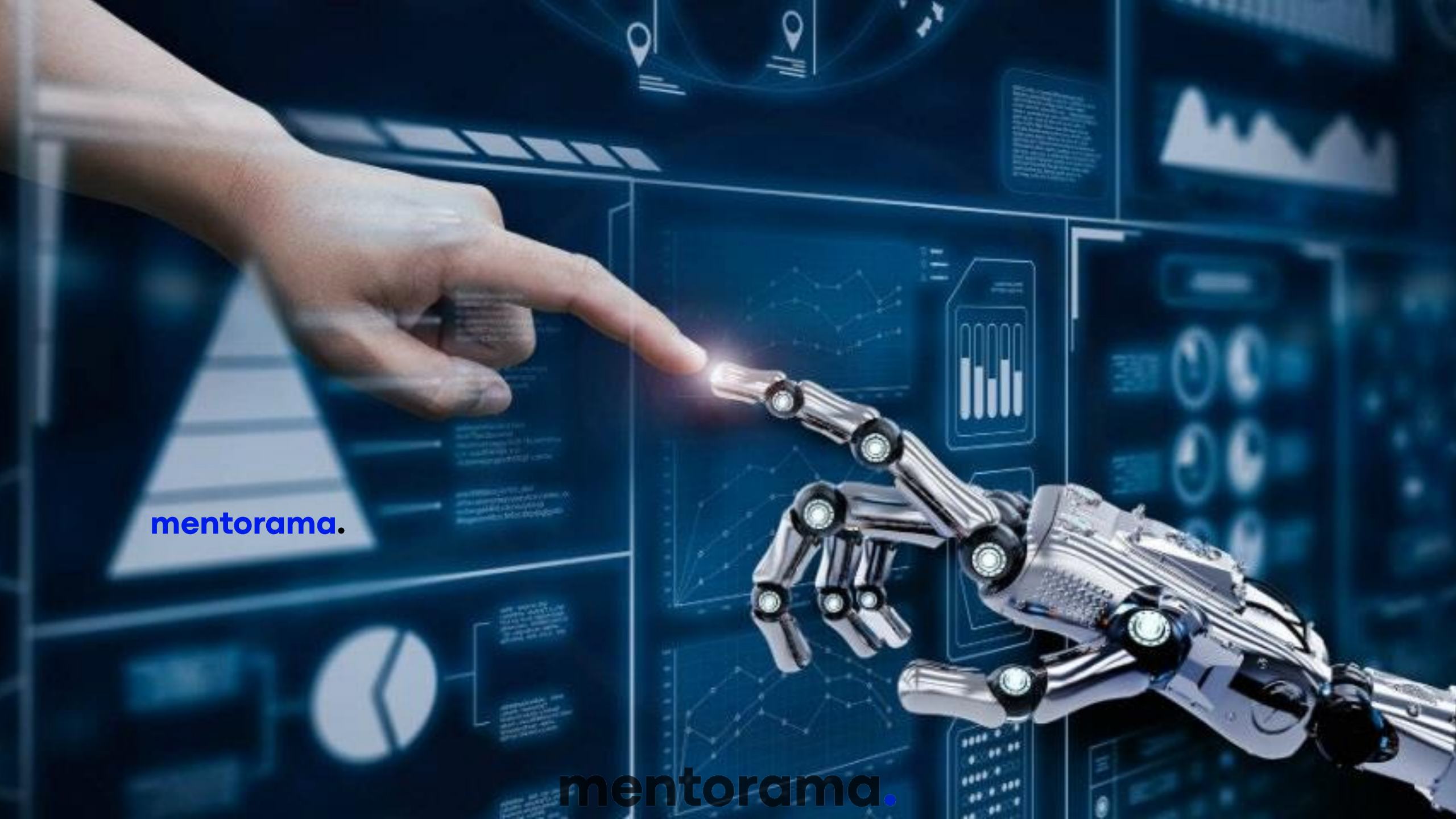
$$v = \frac{v_f}{v - f}$$

$$v = \frac{v_f}{v - f}$$

$$v = \frac{v_f}{v - f}$$

$$v = \frac{v_f}{v - f}$$





mentorama.

mentorama.

# Neste módulo

Aula 1 - Machine Learning

Aula 2 - Etapas

Aula 3 - Prática

Aula 4 - Projeto

# Recursos e ferramentas

- Sci-kit Learn, Pandas, Numpy, Matplotlib

# 1. MACHINE LEARNING

mentorama.

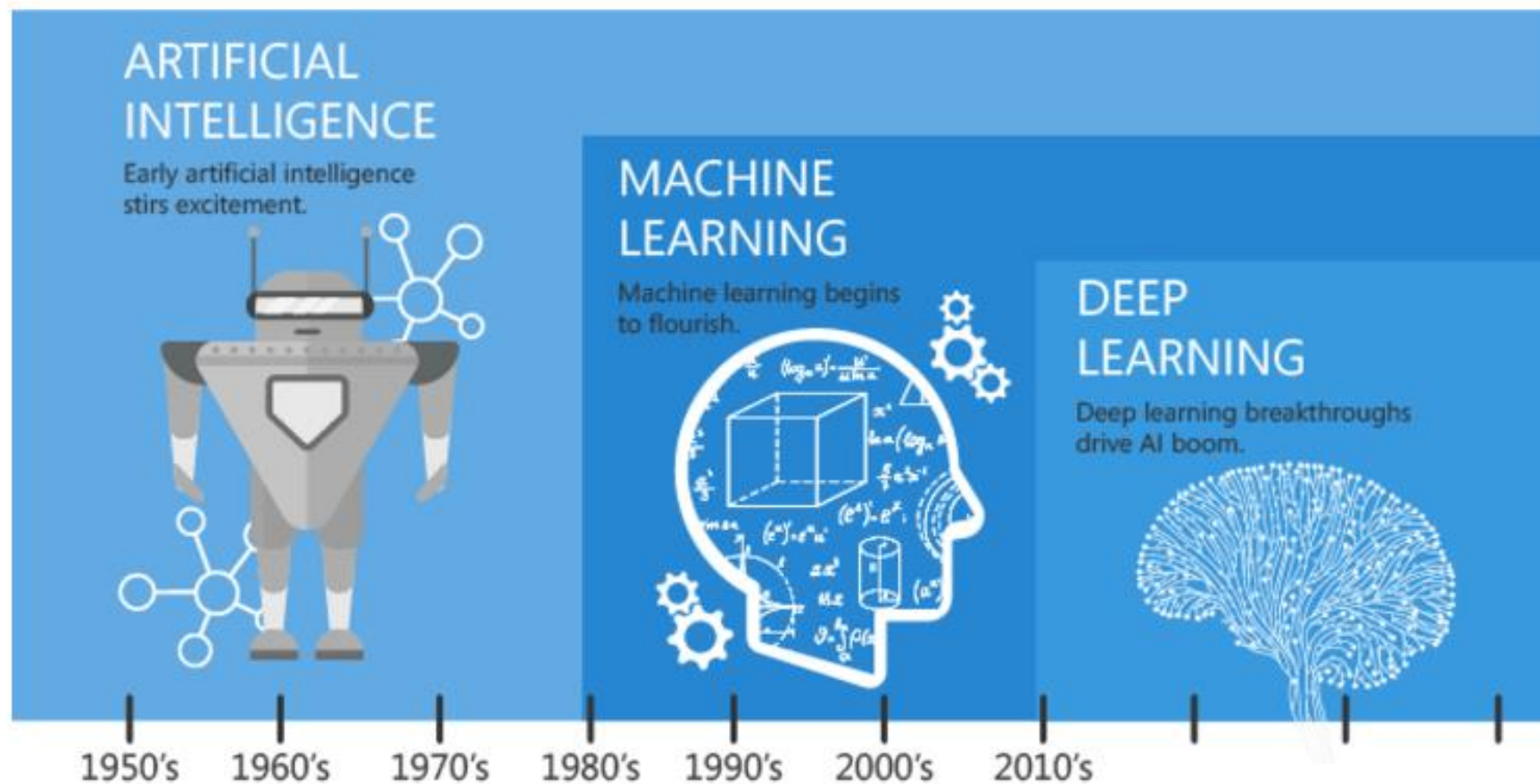
mentorama.



# Machine Learning

- É a ciência (e arte) de programar computadores capazes de aprender com os dados
- É o campo de estudo que dá aos computadores a habilidade de aprender com os dados sem a necessidade de uma programação explícita

# Linha evolutiva



Since an early flush of optimism in the 1950's, smaller subsets of artificial intelligence - first machine learning, then deep learning, a subset of machine learning - have created ever larger disruptions.

**mentorama.**

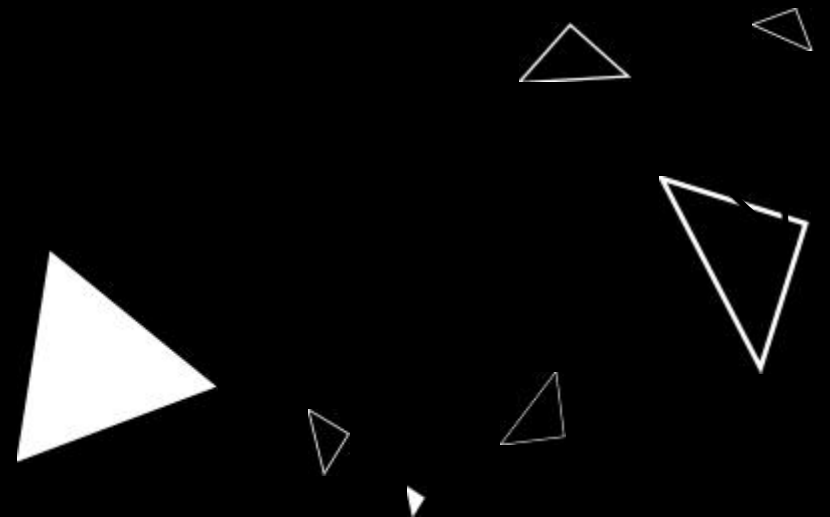
**mentorama.**



# PRINCIPAIS DESAFIOS

mentorama.

mentorama.

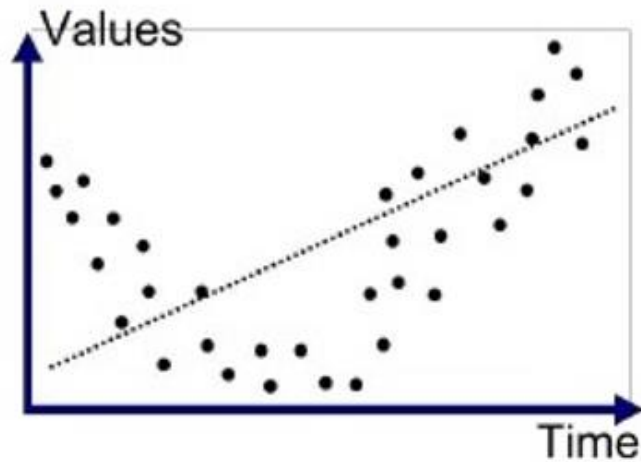


# Machine Learning

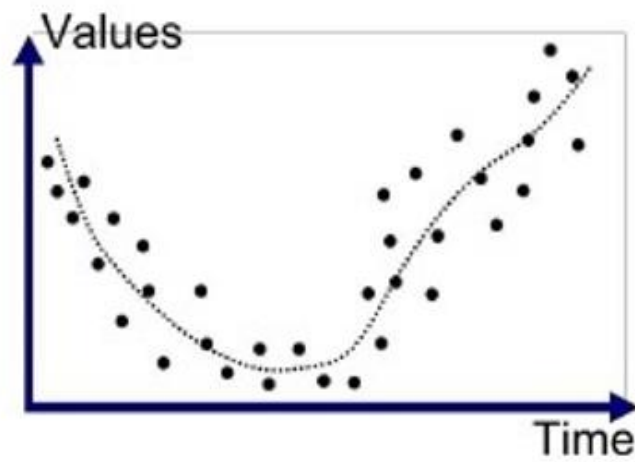
- Insuficiente quantidade de dados de treinamento
- Dados de treinamento não representativos
- Qualidade dos dados ruins
- Características irrelevantes dos dados
- Overfitting e underfitting

# Machine Learning

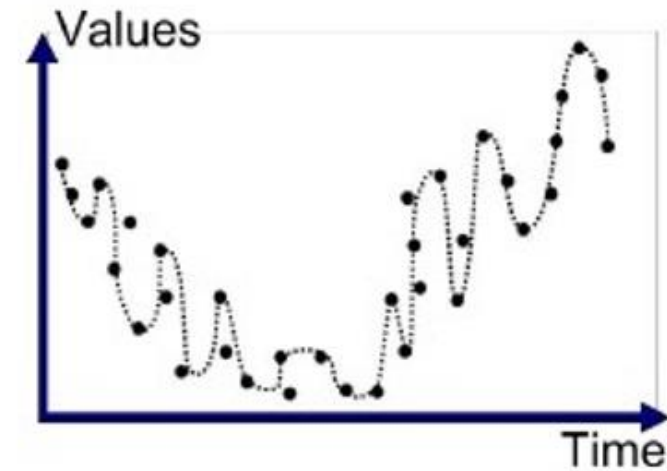
- Overfitting e underfitting



Underfitted



Good Fit/Robust

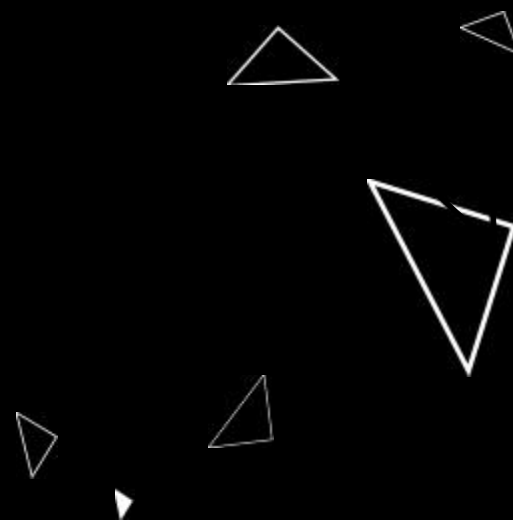


Overfitted

# PRINCIPAIS APLICAÇÕES

mentorama.

mentorama.







**mentorama.**

**Carros autônomos**  
**mentorama.**



mentorama.

Recomendação  
de produtos



**mentorama.**



# Previsão de tendências do mercado de ações

mentorama.





mentorama.

Saúde

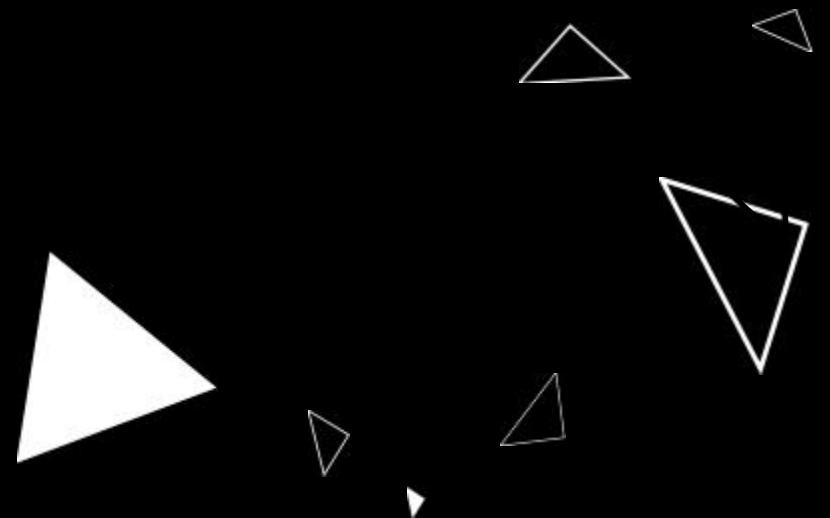
mentorama.



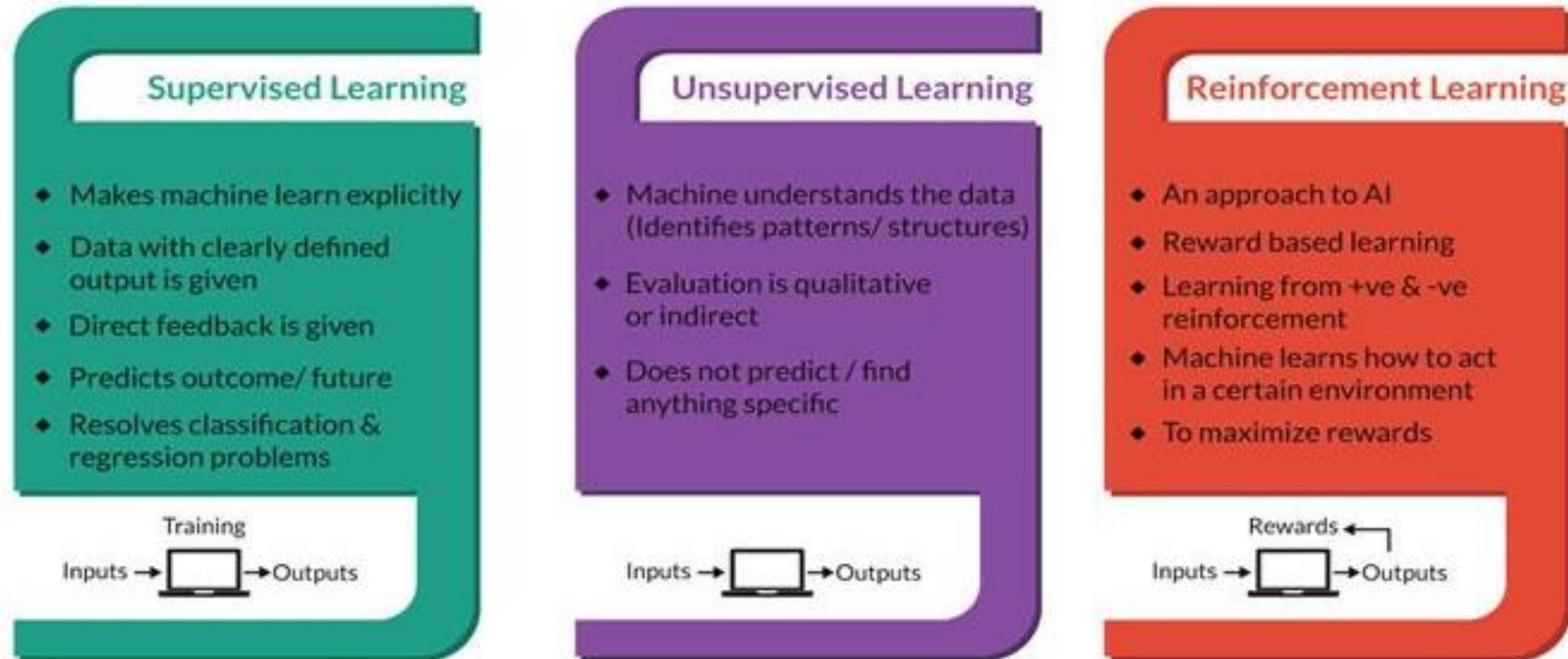
# TIPOS DE APRENDIZADO

mentorama.

mentorama.

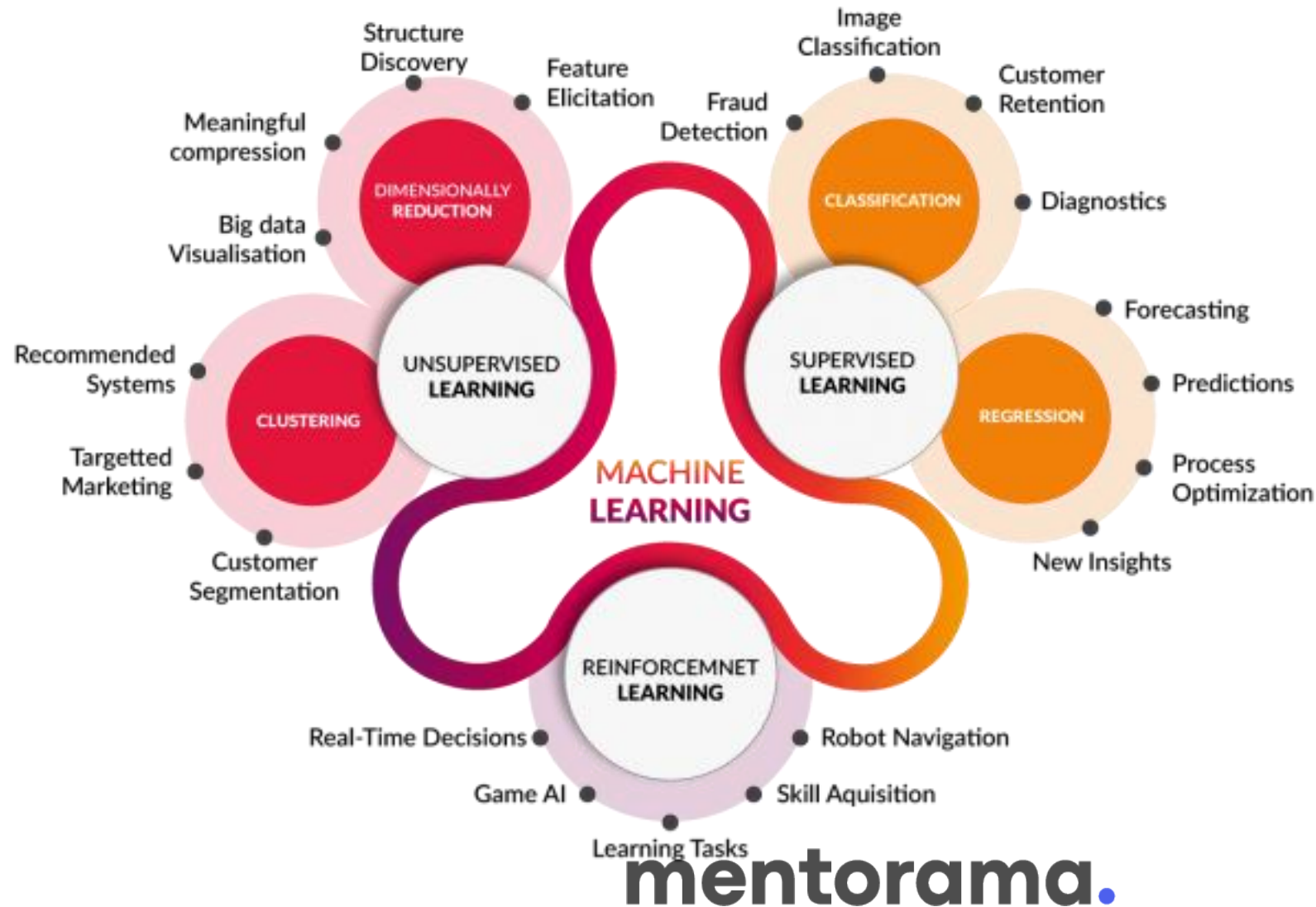


# Machine Learning em ação





# Machine Learning em ação



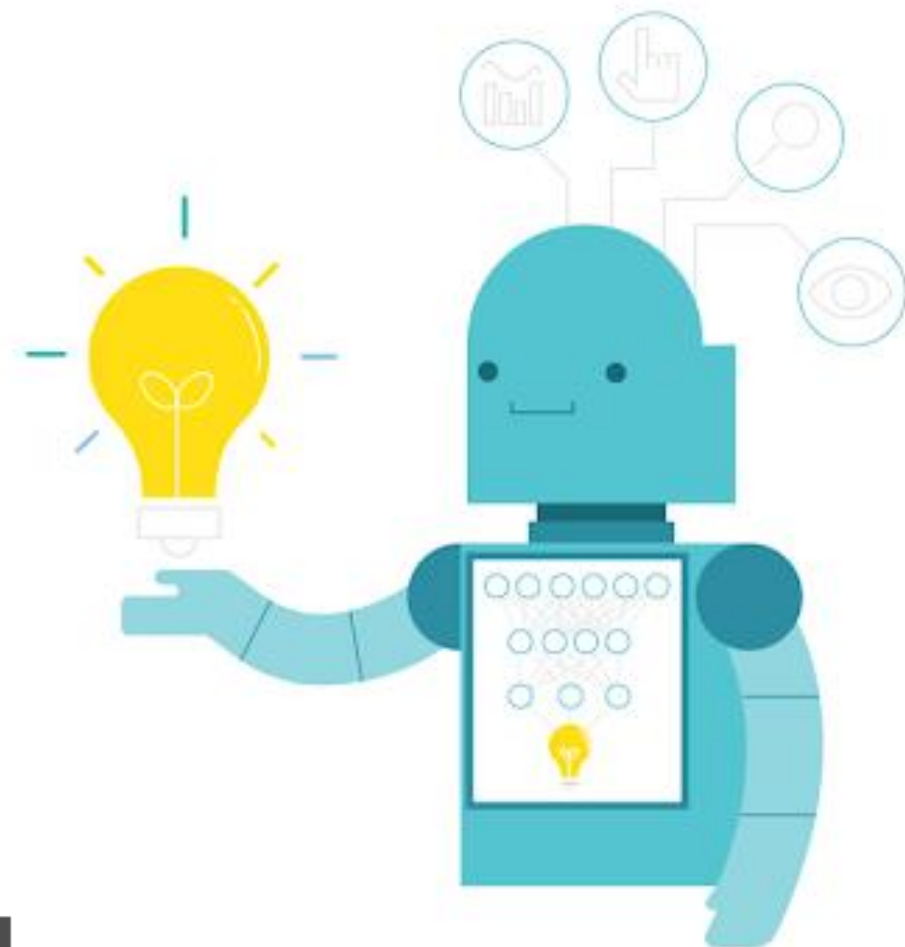


# Principais algoritmos

- Regressão linear
- SVM (Support Vector Machine)
- KNN (K-vizinhos mais próximos)
- Regressão Logística
- Árvore de decisão
- K-Means
- Random Forest
- Naive Bayes

**mentorama.**

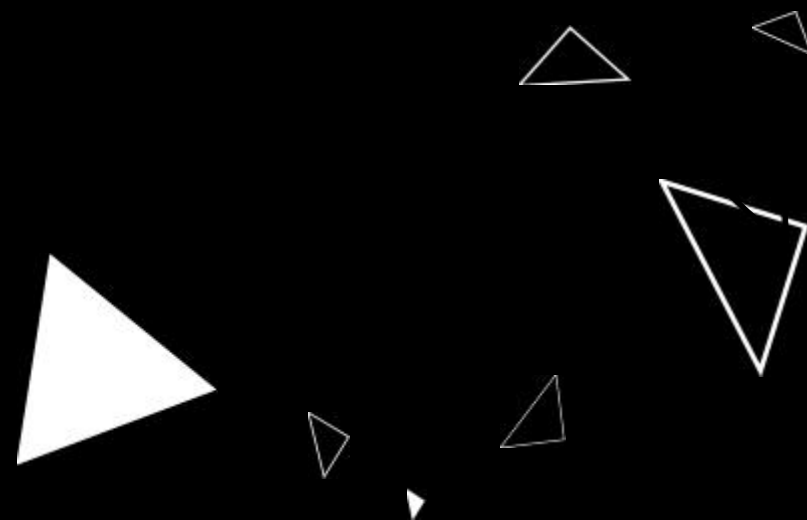
**mentorama.**



# BIBLIOTECAS

**mentorama.**

mentorama.



# Bibliotecas



**mentorama.**






**mentorama.**

# Bibliotecas

- Numpy
- Pandas
- Matplotlib
- Plotnine
- Seaborn
- ScikitLearn

mentorama.

conda-forge / packages / plotnine 0.8.0




A grammar of graphics for python


Conda


Files


Labels


Badges


 License: [GPL 2.0](#)

 Home: <https://github.com/has2k1/plotnine>

 Development: <https://github.com/has2k1/plotnine>

 Documentation: <https://plotnine.readthedocs.io>


 98505 total downloads

 Last upload: 1 day and 4 hours ago


### Installers

Info: This package contains files in non-standard labels.



conda install ?

 linux-64


 v0.2.1

 win-32


 v0.2.1

  noarch

 v0.8.0

 win-64

 v0.2.1

 osx-64

 v0.2.1

To install this package with conda run one of the following:

```
conda install -c conda-forge plotnine
conda install -c conda-forge/label/gcc7 plotnine
conda install -c conda-forge/label/cf201901 plotnine
conda install -c conda-forge/label/cf202003 plotnine
```

mentorama.



# Como instalar as bibliotecas

Anaconda Prompt (anaconda3) - conda install seaborn - conda install -c conda-forge plotnine

```
(base) C:\Users\felip>conda activate deeplearning  
  
(deeplearning) C:\Users\felip>conda install seaborn  
Collecting package metadata (current_repodata.json): done  
Solving environment: done  
  
## Package Plan ##  
  
environment location: C:\Users\felip\anaconda3\envs\deeplearning  
  
added / updated specs:  
- seaborn
```

The following packages will be downloaded:

package	build	
ca-certificates-2021.1.19	haa95532_1	119 KB
openssl-1.1.1k	h2bbff1b_0	4.8 MB
Total:		4.9 MB

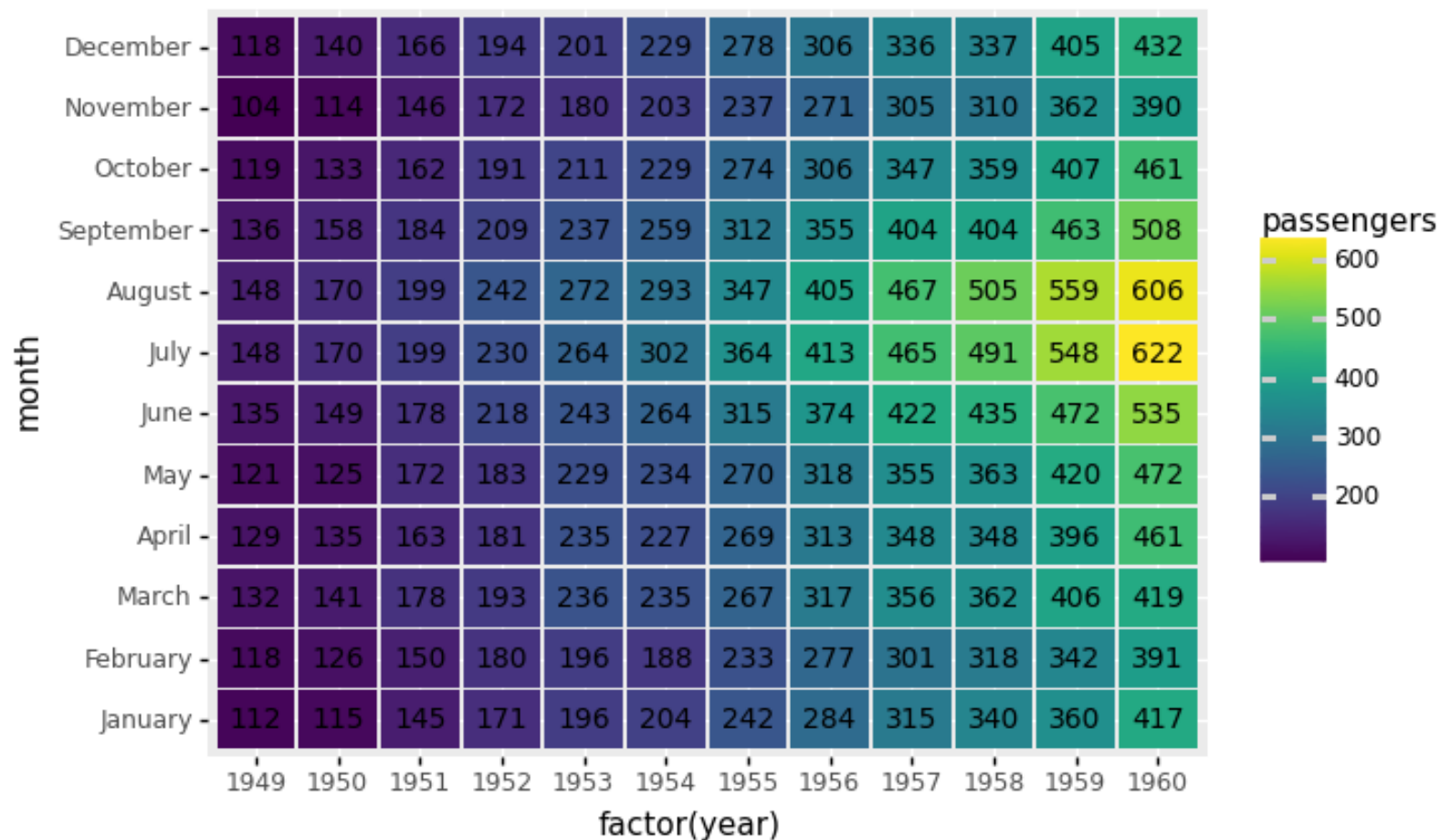
The following packages will be UPDATED:

ca-certificates	2021.1.19-haa95532_0 --> 2021.1.19-haa95532_1
openssl	1.1.1j-h2bbff1b_0 --> 1.1.1k-h2bbff1b_0

mentorama

mentorama.

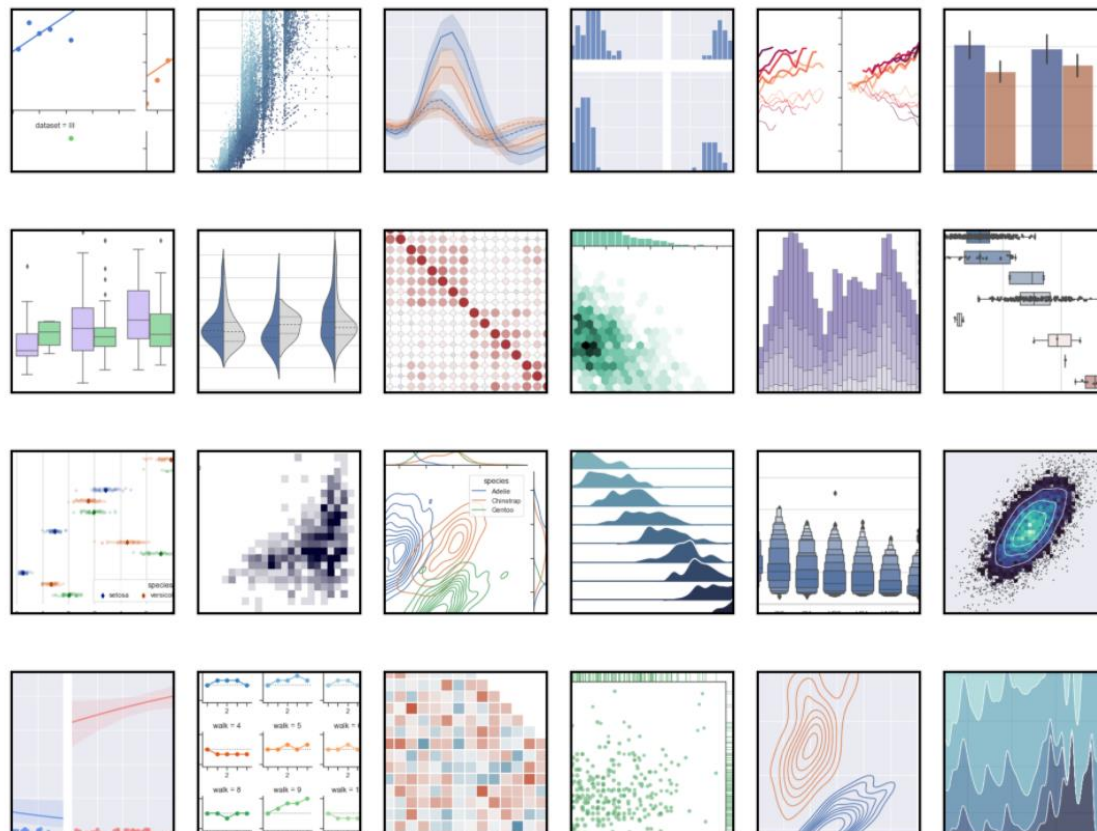
# Plotnine



mentorama.

mentorama.

# Seaborn



**mentorama.**

**mentorama.**

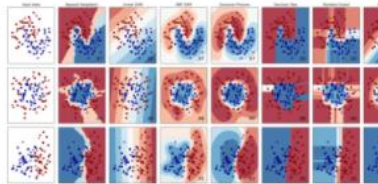
# ScikitLearn

## Classification

Identifying which category an object belongs to.

**Applications:** Spam detection, image recognition.

**Algorithms:** SVM, nearest neighbors, random forest, and more...



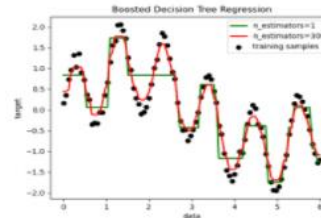
Examples

## Regression

Predicting a continuous-valued attribute associated with an object.

**Applications:** Drug response, Stock prices.

**Algorithms:** SVR, nearest neighbors, random forest, and more...



Examples

## Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation, Grouping experiment outcomes

**Algorithms:** k-Means, spectral clustering, mean-shift, and more...



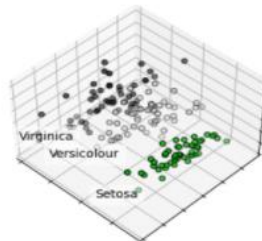
Examples

## Dimensionality reduction

Reducing the number of random variables to consider.

**Applications:** Visualization, Increased efficiency

**Algorithms:** k-Means, feature selection, non-negative matrix factorization, and more...



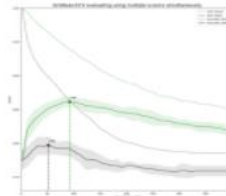
Examples

## Model selection

Comparing, validating and choosing parameters and models.

**Applications:** Improved accuracy via parameter tuning

**Algorithms:** grid search, cross validation, metrics, and more...



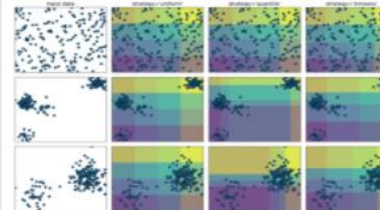
Examples

## Preprocessing

Feature extraction and normalization.

**Applications:** Transforming input data such as text for use with machine learning algorithms.

**Algorithms:** preprocessing, feature extraction, and more...



Examples

mentorama.

mentorama.

# Etapas básicas

- Formulação do problema
- Importar os dados
- **Análise exploratória / Limpeza / Pré processamento**
- Dividir o conjunto em treino e teste
- Selecionar e treinar o modelo
- Fazer a predição
- Avaliar e aprimorar o modelo



# Resumo

- Principais aplicações
- Linha evolutiva
- Tipos de aprendizado
- Bibliotecas
- Algoritmos
- Etapas



# 2. GOOGLE COLAB

mentorama.

mentorama.



# O que é o Google Colab?

The logo for Google Colab, featuring the word "colab" in a bold, sans-serif font. The "co" is yellow with a subtle gradient, and the "lab" is orange. The letters are closely spaced.

mentorama.

mentorama.

# Principais características

- Não necessita configurações
- Já conta com bibliotecas pré instaladas
- Facilita o compartilhamento de código
- Utilidade em ML, IA, Data Science
- Conta com exemplos na plataforma
- Uso de GPU gratuitamente (Tensor Flow)
- O código é salvo no Google Drive
- Integração com GitHub / Gist

**mentorama.**

**mentorama.**



# Vamos praticar?

- Como iniciar um Notebook?
- Como habilitar uma GPU?
- Como importar dados externos?
- Como integrar o Colaboratory com o GitHub?
- Como instalar bibliotecas externas?
- Como lidar com erros?



**mentorama.**

**mentorama.**

# Resumo

- O que é o Google Colab
- Principais características
- Habilitação da GPU
- Hello World
- Importar base de dados
- Integrar com o GitHub
- Instalando bibliotecas
- Lidando com erros



# 2.ETAPAS

mentorama.

mentorama.



# Etapas

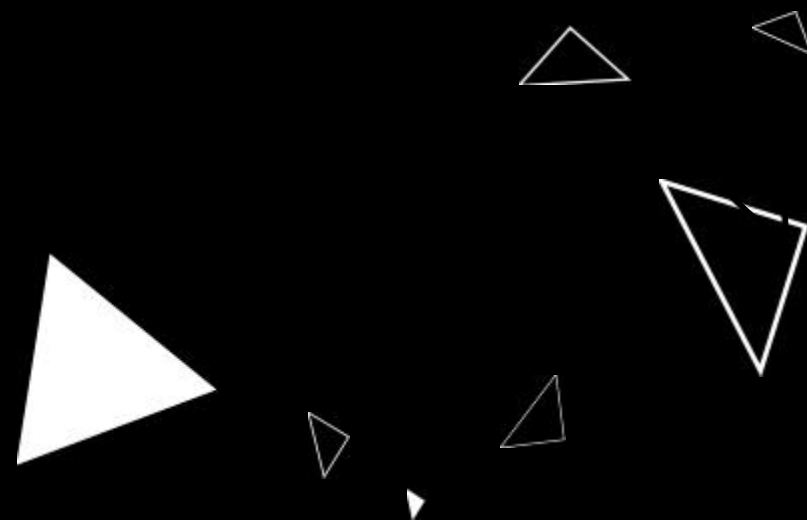
- Formulação do problema
- Importar os dados
- **Análise exploratória / Limpeza / Pré processamento**
- Dividir o conjunto em treino e teste
- Selecionar e treinar o modelo
- Fazer a predição
- Avaliar e aprimorar o modelo



# FORMULAÇÃO DO PROBLEMA

mentorama.

mentorama.

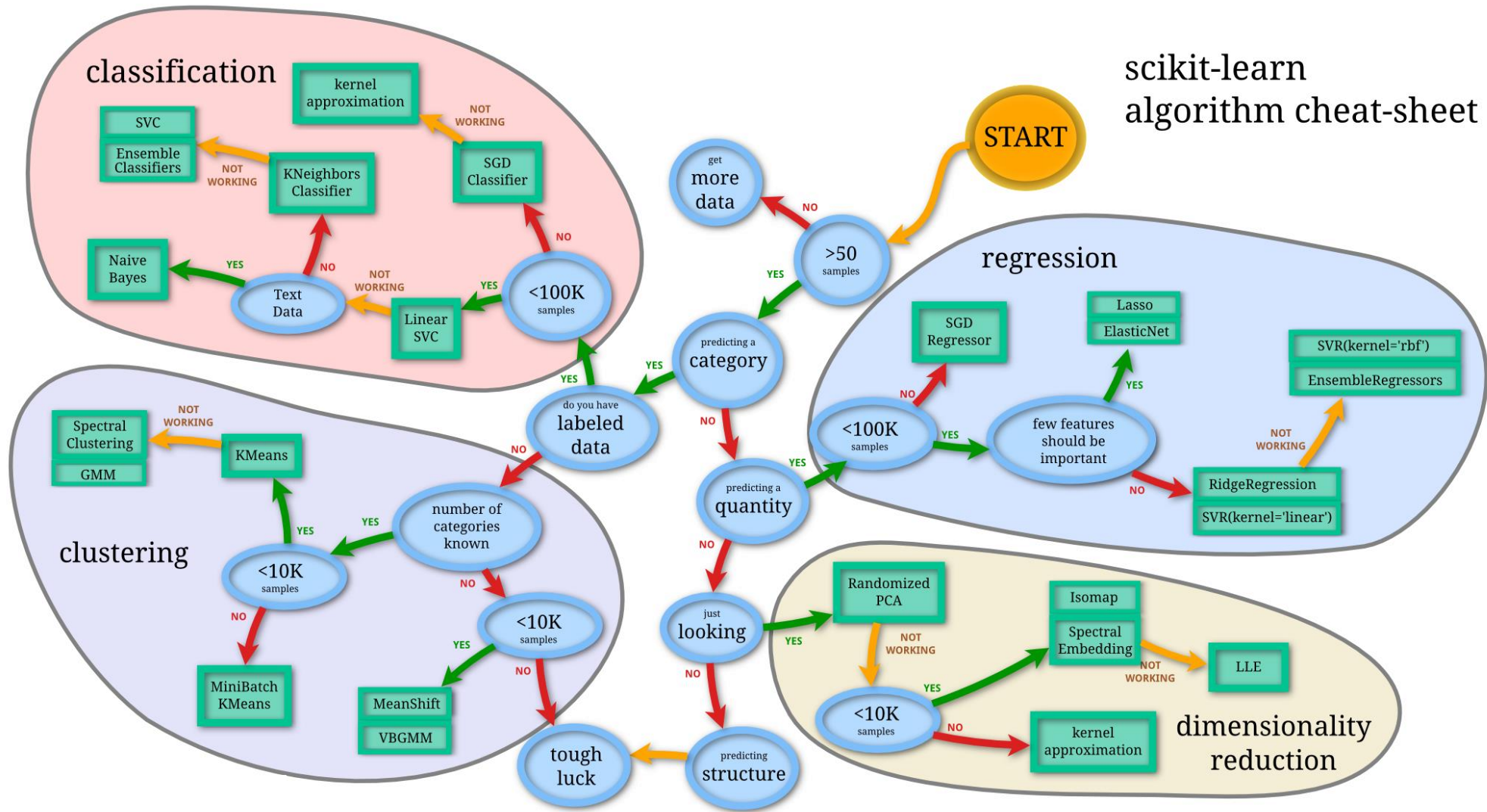


# Algumas perguntas para classificação

- Como é o dataset?
- Quais são as classes?
- Como escolher o algoritmo de classificação?
- Como saber se o modelo performou bem?
- Como escolher as métricas de avaliação?

# Escolhendo o estimador certo

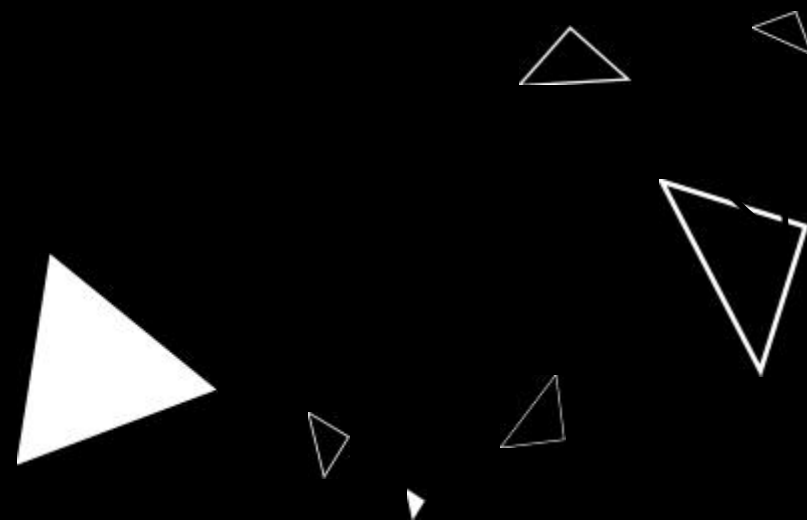
mentorama.



# OBTENDO OS DADOS

mentorama.

mentorama.





# Obtenção dos dados

- Kaggle
- Google Dataset Search
- Microsoft Azure Public Datasets
- Sci-kit Learn datasets
- UCI Machine Learning Repository
- Public Datasets on Github

kaggle



**mentorama.**

**mentorama.**

# Sci-kit Learn datasets

<code>load_boston(* [, return_X_y])</code>	Carregue e retorne o conjunto de dados de preços de casas em Boston (regressão).
<code>load_iris(* [, return_X_y, as_frame])</code>	Carregue e retorne o conjunto de dados da íris (classificação).
<code>load_diabetes(* [, return_X_y, as_frame])</code>	Carregue e retorne o conjunto de dados de diabetes (regressão).
<code>load_digits(* [, n_class, return_X_y, as_frame])</code>	Carregue e retorne o conjunto de dados de dígitos (classificação).
<code>load_linnerud(* [, return_X_y, as_frame])</code>	Carregue e retorne o conjunto de dados linnerud do exercício físico.
<code>load_wine(* [, return_X_y, as_frame])</code>	Carregue e retorne o conjunto de dados do vinho (classificação).
<code>load_breast_cancer(* [, return_X_y, as_frame])</code>	Carregue e devolva o conjunto de dados de Wisconsin (classificação) do câncer de mama.

**mentorama.**

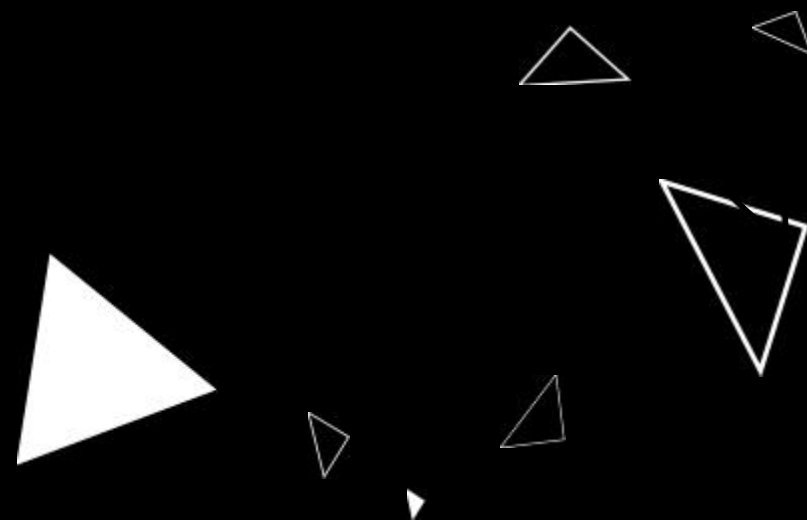
`digits = dataset.load_digits()`

**mentorama.**

# PRÉ PROCESSAMENTO

mentorama.

mentorama.



# Pré-processamento

- Limpeza de dados
- Manipulando texto e atributos categóricos
- Transformações customizadas (combinar atributos por ex.)
- Dimensionamento das features (normalização / padronização)
- Pipelines de transformação (padronização por ex.)
- Operações morfológicas (dilatação e erosão por ex.)
- Retirar brilho, aumentar contraste
- Aumentar os dados (data augmentation)
- Converter imagens em tons de cinza ou outros espaços de cores
- Diminuir / Aumentar tamanho da imagem

**mentorama.**

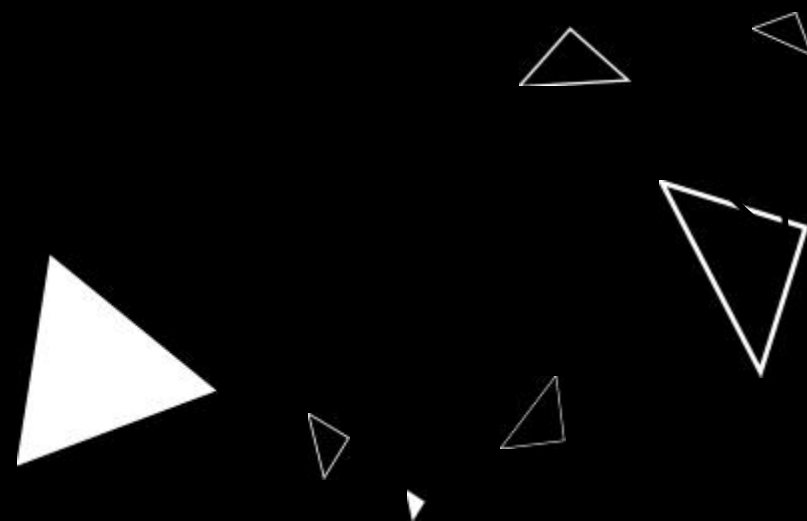
**mentorama.**



# SELECIONANDO E TREINANDO UM MODELO

mentorama.

mentorama.

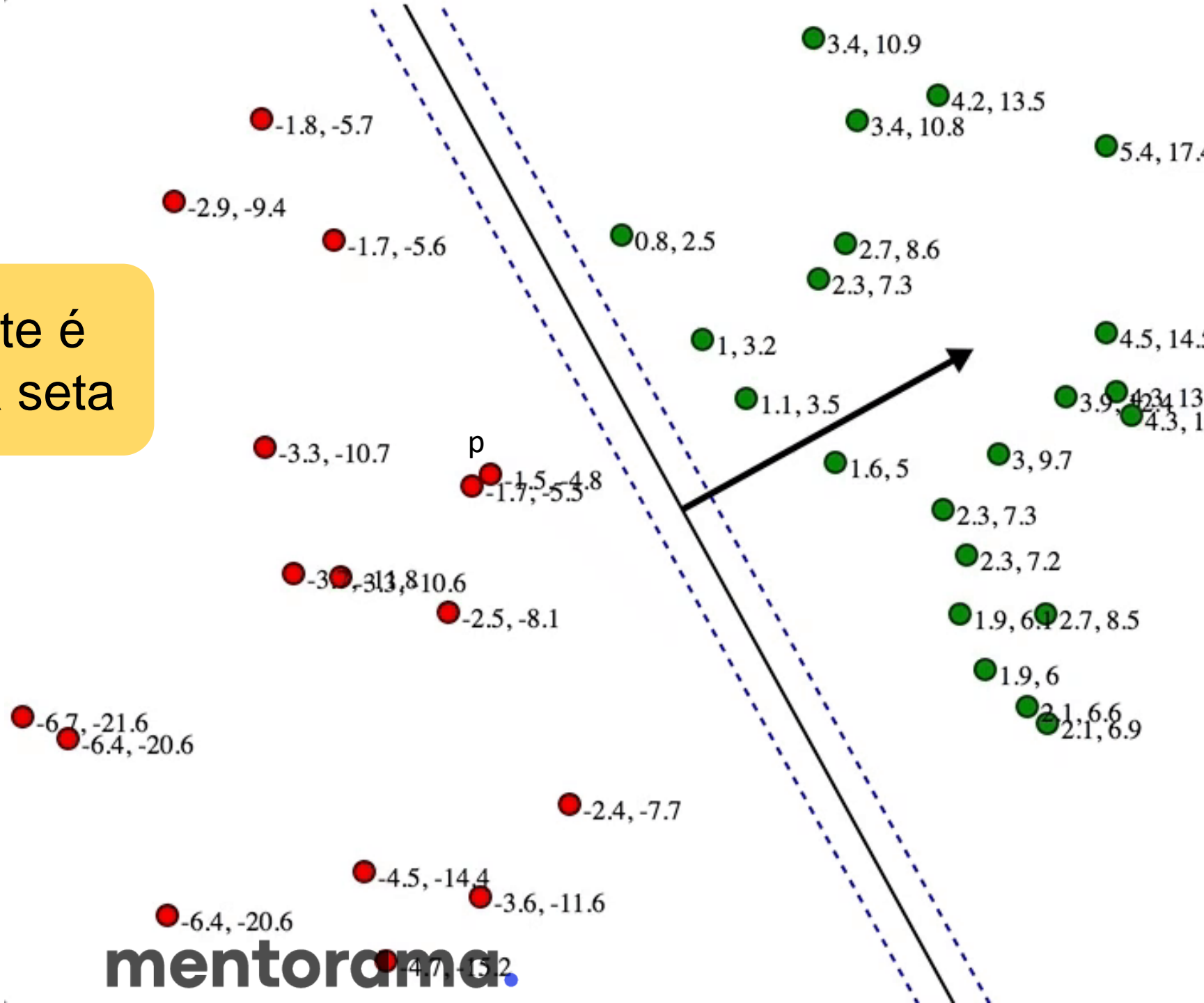


# Modelo de classificação

- Um classificador é basicamente um algoritmo que usa “conhecimento” obtido dos dados de treinamento para mapear os dados de entrada para uma categoria ou classe específica
- Classificadores podem ser binários ou multiclases

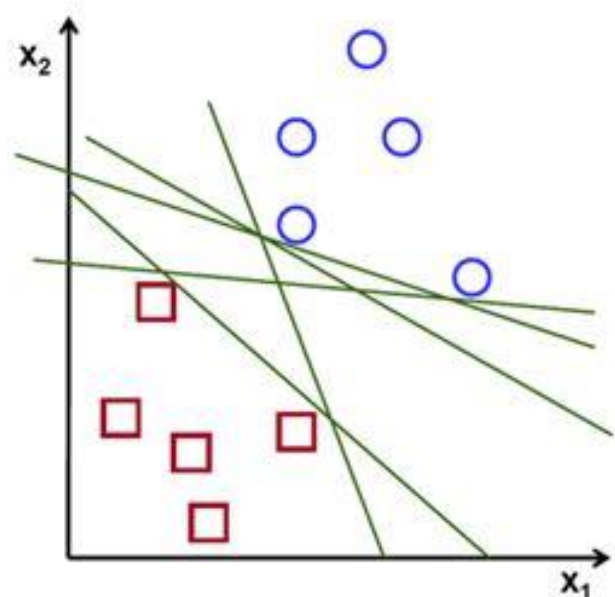
# SVM

O vetor de suporte é representado pela seta

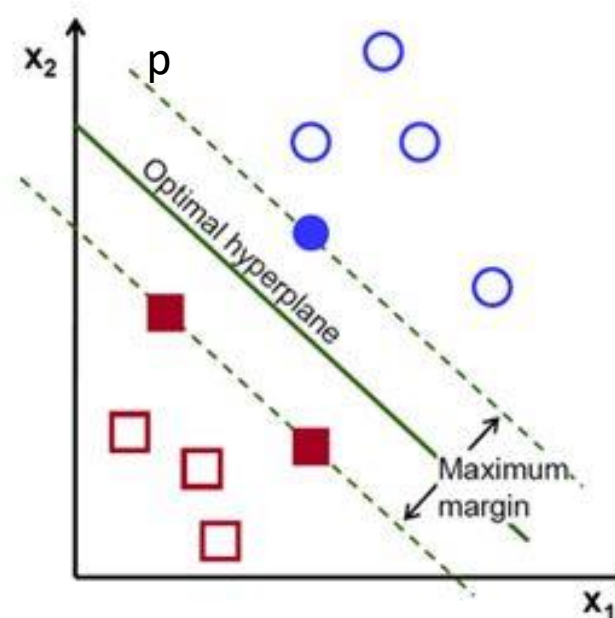


# SVM

- A partir de duas ou mais classes rotuladas de dados, o algoritmo busca encontrar um hiperplano ideal que separe todas as classes



Possible hyperplanes



mentorama.

mentorama.

# Cross validation

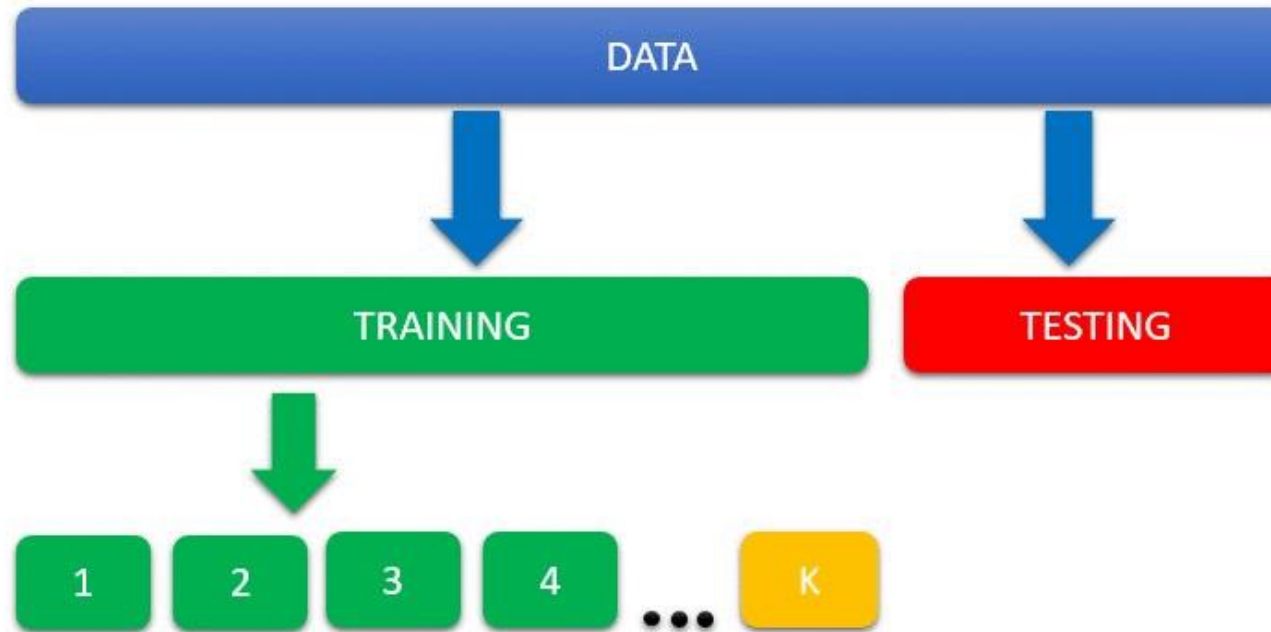


mentorama.

mentorama.



# Cross validation



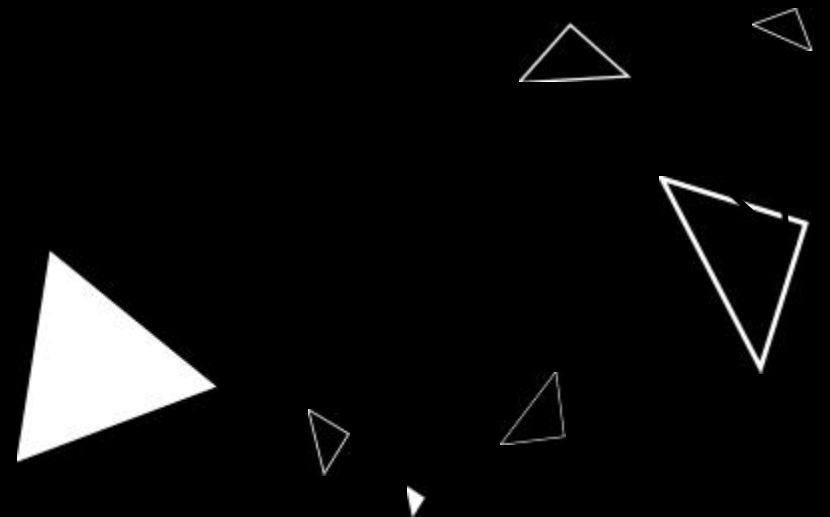
mentorama.

mentorama.

# AVALIAÇÃO DOS RESULTADOS

mentorama.

mentorama.



# Matriz de confusão

- Uma matriz de confusão é uma tabela que indica os erros e acertos do seu modelo, comparando com o resultado esperado (ou rótulos / labels).

		Detectada	
		Sim	Não
Real	Sim	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

mentorama.

mentorama.

# Matriz de confusão

	Fraude (sim)	Legítima (não)
Fraude (sim)	200 (VP)	50 (FP)
Legítima (não)	50 (FN)	700 (VN)

- True Positive (VP): classificação correta da classe Positivo.
- False Negative (FN): erro em que o modelo previu a classe Negativo quando o valor real era classe Positivo.
- False Positive (FP): erro em que o modelo previu a classe Positivo quando o valor real era classe Negativo.
- True Negative (VN): classificação correta da classe Negativo.

# Matriz de confusão

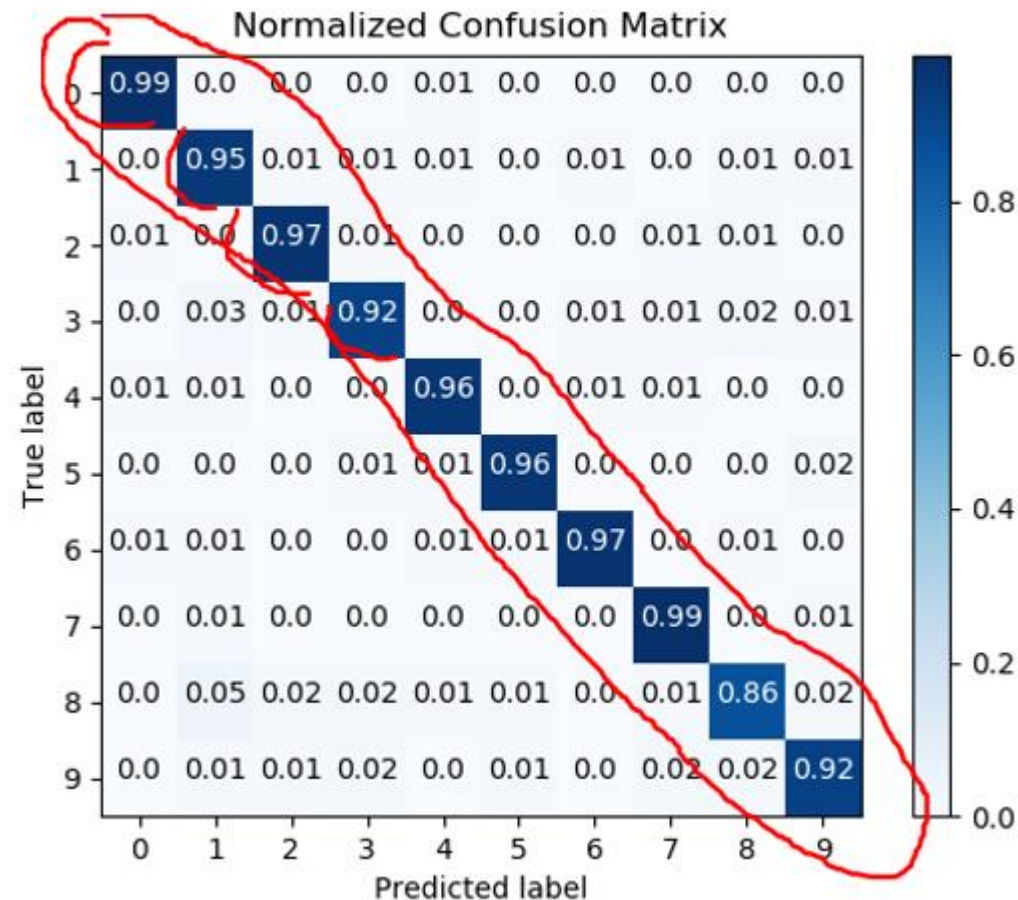
	Fraude (sim)	Legítima (não)
Fraude (sim)	200 (VP)	50 (FP)
Legítima (não)	50 (FN)	700 (VN)

- Total de registros: 1000.
- Total de transações com fraude: 250 - 25%.
- Total de transações legítimas: 750 - 75%.
- Taxa de acerto (acurácia): 90%.



# Matriz de confusão

- A nomenclatura apresentada auxilia na compreensão para aplicação em diversas métricas como acurácia, recall, precision, F1 score dentre outros.



# Acurácia

- A acurácia é uma boa indicação geral de como o modelo performou, e pode ser definida como:

$$(TP + TN) / (TP + TN + FP + FN)$$

ou

**Total de acertos / Total de dados do conjunto**

ou

**1 – taxa de erro**

# Acurácia

- Nível de acurácia entre 0% e 30%
- Nível de acurácia entre 30% e 70%
- Nível de acurácia entre 70% e 100%



# Acurácia

- Algumas perguntas:
  - Quanto maior a taxa de acurácia, melhor?
  - Uma taxa de acurácia alta permite saber se o modelo é bom ou ruim?

	Fraude (sim)	Legítima (não)
Fraude (sim)	200 (VP)	50 (FP)
Legítima (não)	50 (FN)	700 (VN)

# Principais métricas

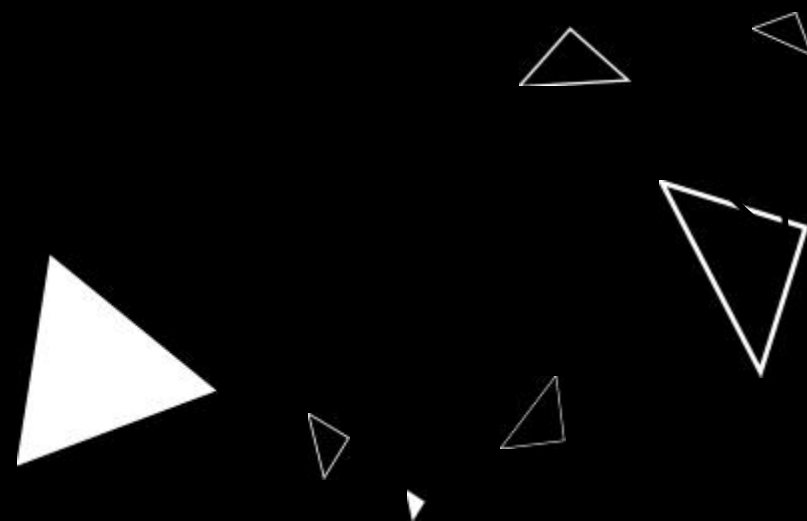
Métrica	Fórmula	Interpretação
Acurácia	$\frac{TP + TN}{TP + TN + FP + FN}$	Desempenho geral do modelo
Precisão	$\frac{TP}{TP + FP}$	Quão precisas são as predições positivas
Revocação Sensibilidade	$\frac{TP}{TP + FN}$	Cobertura da amostra positiva real
Specificity	$\frac{TN}{TN + FP}$	Cobertura da amostra negativa real
F1 score	$\frac{2TP}{2TP + FP + FN}$	Métrica híbrida útil para classes desequilibradas



# APRIMORAR O MODELO

mentorama.

mentorama.

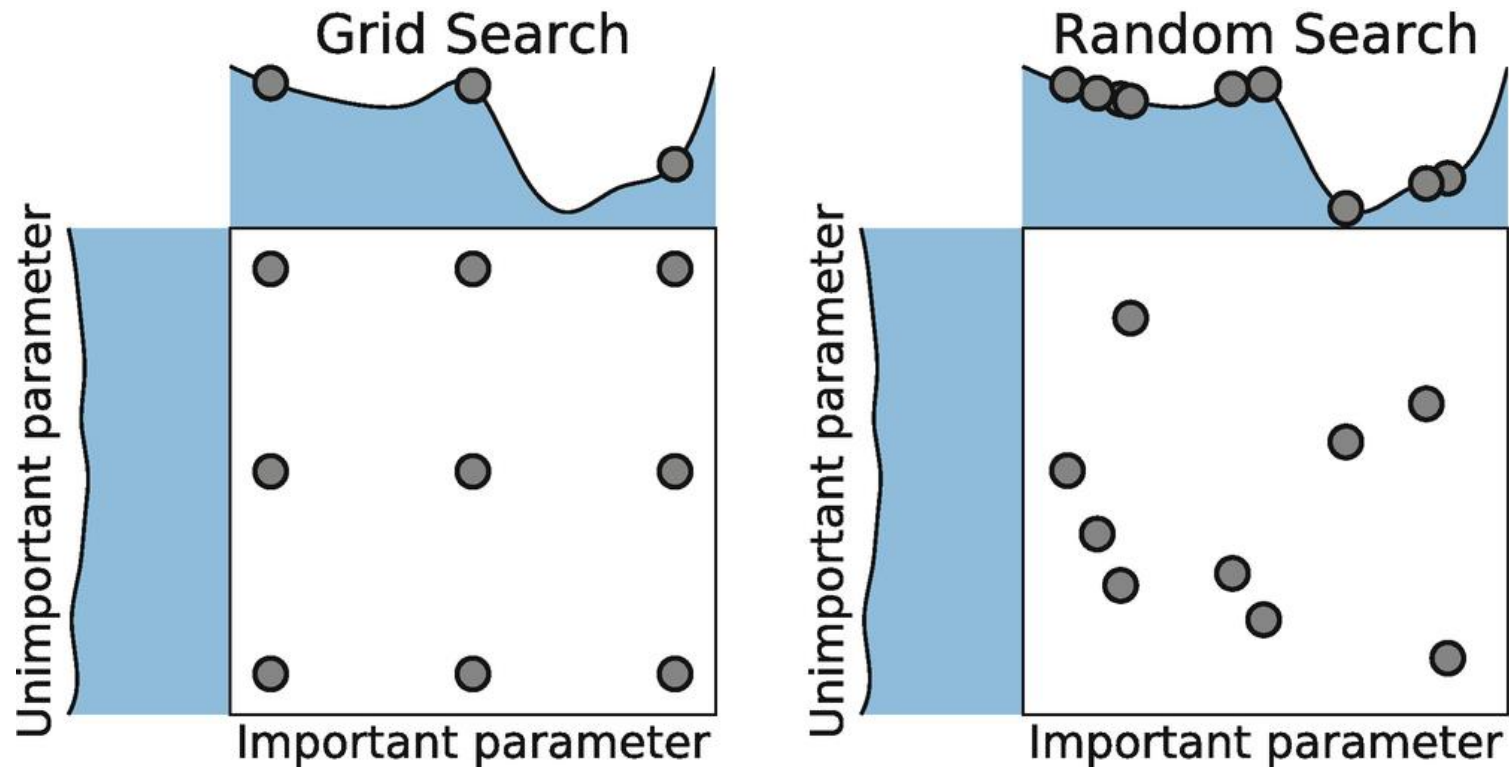


# Aprimorar o modelo

- Verificar a base de dados
- Verificar o algoritmo
- Ajuste de parâmetros
- Treinar, Validar e Testar
- Avaliar os resultados
- Comparar com outros resultados



# Aprimorando o modelo



mentorama.

mentorama.

# Grid search

```
from sklearn.model_selection import GridSearchCV

param_grid = [
    {'n_estimators': [3, 10, 30], 'max_features': [2, 4, 6, 8]},
    {'bootstrap': [False], 'n_estimators': [3, 10], 'max_features': [2, 3, 4]},
]

forest_reg = RandomForestRegressor()

grid_search = GridSearchCV(forest_reg, param_grid, cv=5,
                           scoring='neg_mean_squared_error',
                           return_train_score=True)

grid_search.fit(housing_prepared, housing_labels)

>>> grid_search.best_params_
{'max_features': 8, 'n_estimators': 30}
```

**mentorama.**

**mentorama.**

# Random search

- Vejamos o exemplo:
  - Se você permitir que a pesquisa aleatória seja executada por, digamos, 1.000 iterações, essa abordagem explorará 1.000 valores diferentes para cada hiperparâmetro (em vez de apenas alguns valores por hiperparâmetro com a abordagem de pesquisa em grade).
  - Simplesmente definindo o número de iterações, você tem mais controle sobre o quanto de recurso computacional você deseja alocar para a pesquisa de hiperparâmetros.

# Resumo

- Principais etapas
- Obtenção dos dados
- Pré processamento
- SVM
- Treinamento, validação e teste
- Avaliação do modelo
- Grid Search e Random Search





# 3. PRÁTICA

mentorama.

mentorama.



# Vamos praticar?

- Nesta prática iremos explorar a utilização de algoritmos de Machine Learning para a tarefa de classificação de imagens para reconhecimentos de dígitos manuscritos.



# Resumo

- Classificação de imagens com Mnist dataset
- Obtenção dos dados
- Construção do modelo
- Avaliação do modelo



# PROJETO

mentorama.

mentorama.

