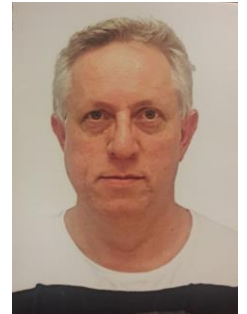


Introdução ao 'Machine Learning' e Análise de Sentimento

Instrutor: Antonio Luis Amadeu
Carga horária: 8 à 10 horas.

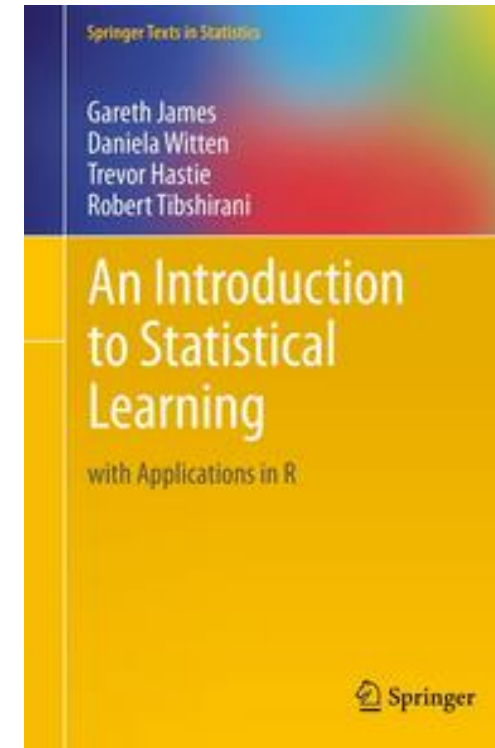
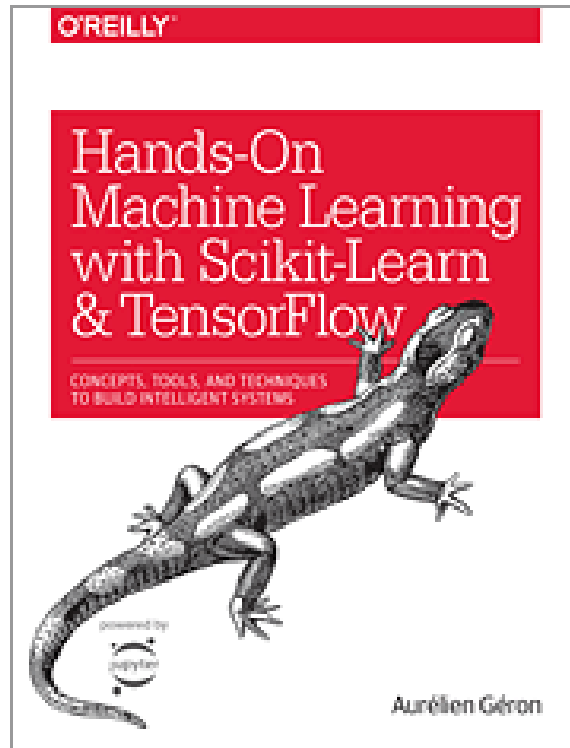


- Bio:
- Formado em Ciências da Computação
 - Atualmente cursa Matemática Aplicada e Computacional
 - Atuou em várias Multinacionais:
 - Unilever
 - TE Connectivity
 - Microsoft
 - Samsung
 - Allianz
-
- Há mais de 5 anos como cientista de dados e especialista em IoT.

Programa do Curso:

- Machine Learning
 - Tipos
 - Supervisionada
 - Não Supervisionada
 - Características
 - Regressão
 - Classificação
 - Agrupamento
 - Preparação dos dados
 - Deep Learning
 - Time Series
 - Análise de Sentimento
-

Referências



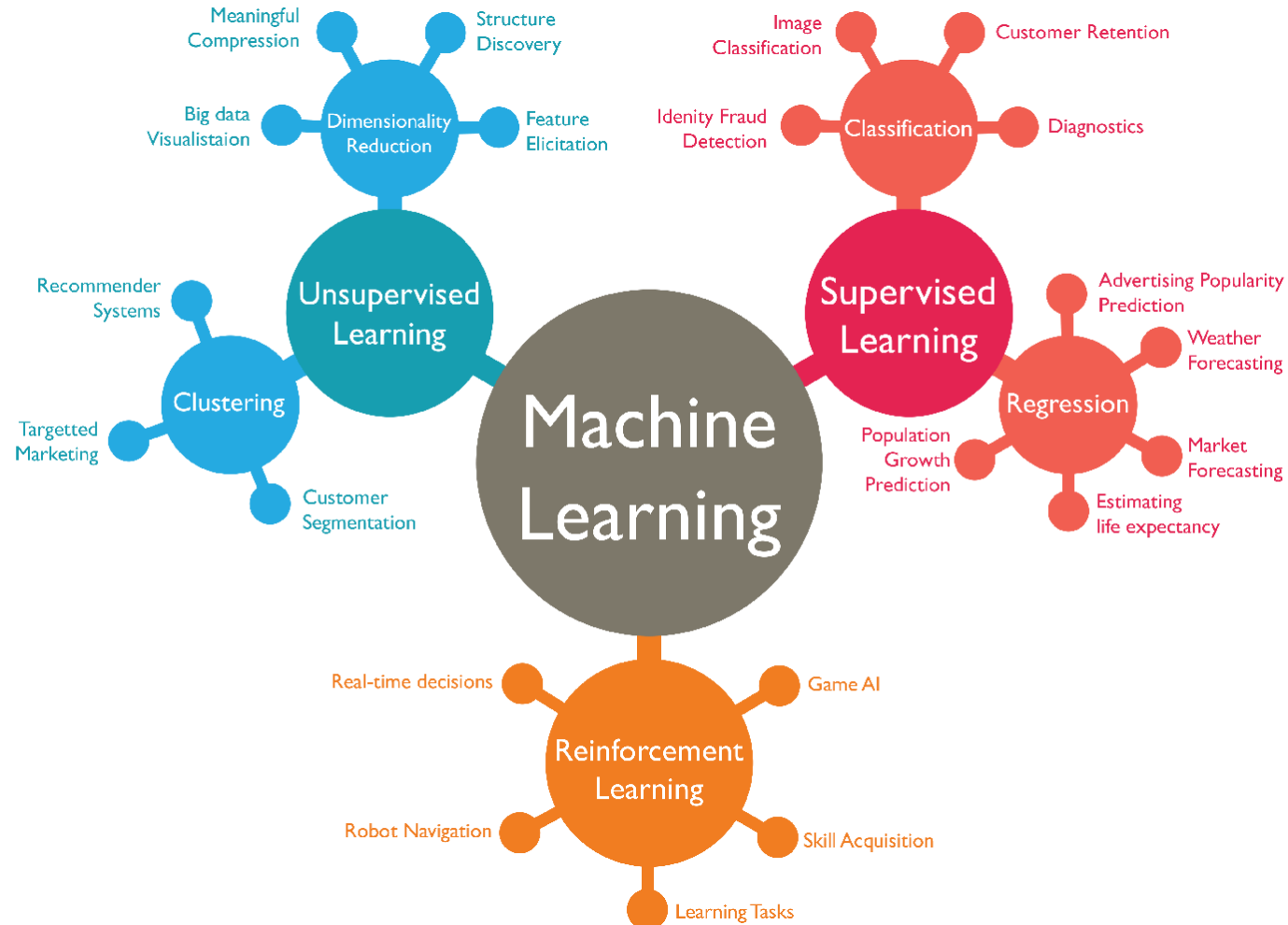
Uma breve introdução ao termo: “Machine Learning”

- O termo foi criado por Arthur Samuel em 1959

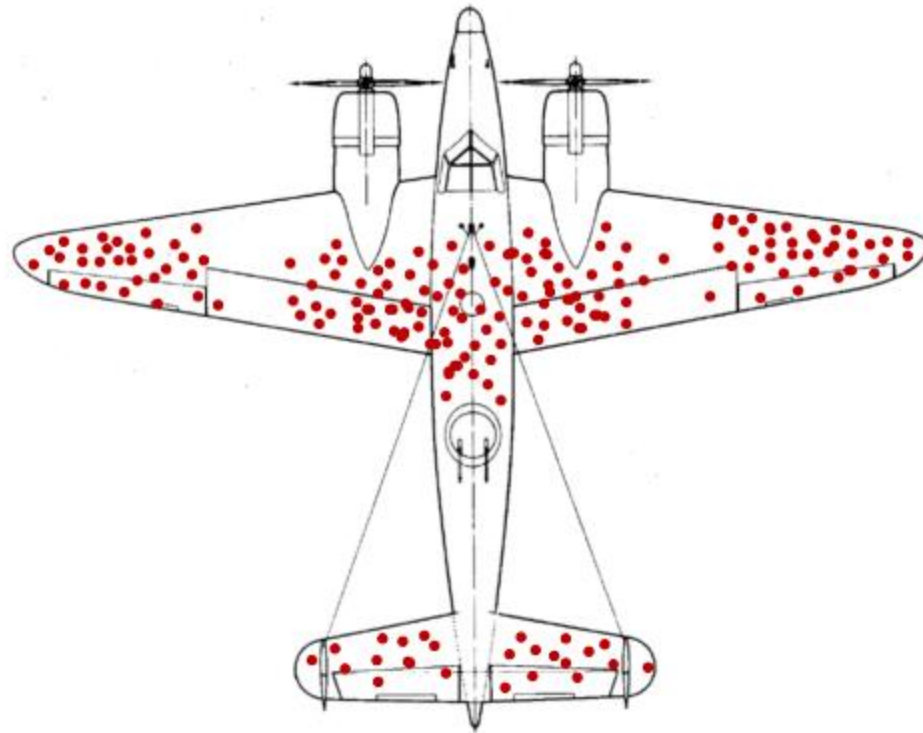


- O nome correto estatisticamente é “Predictive Modelling”
 - O que mudou desde então:
 - Poderio computacional
 - Preço do armazenamento (RAM, HD's, etc.)
 - Disponibilidade dos dados
-

Tipos de “Machine Learning”

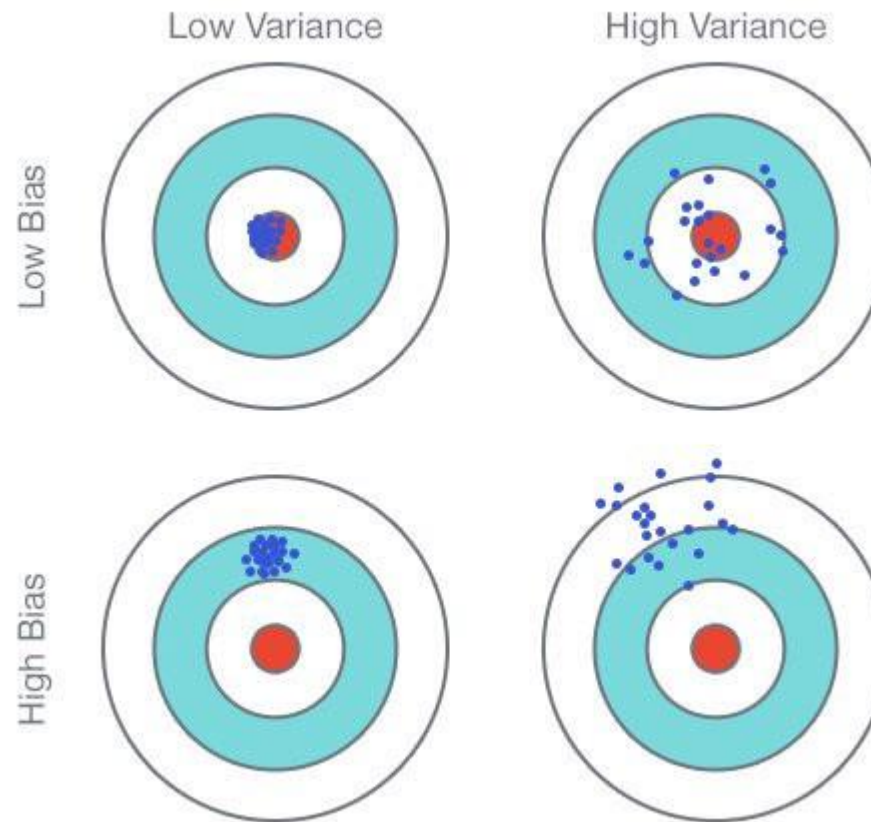


Qual o “truque” para ser um bom cientista de dados?



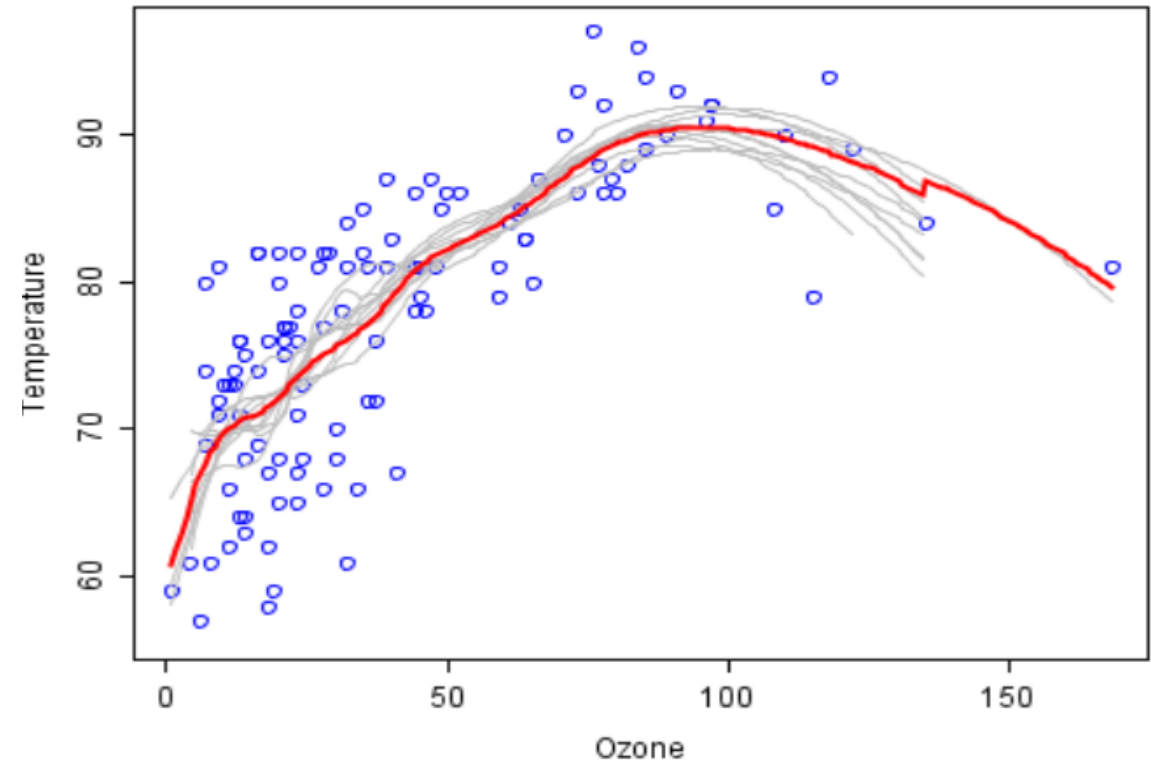
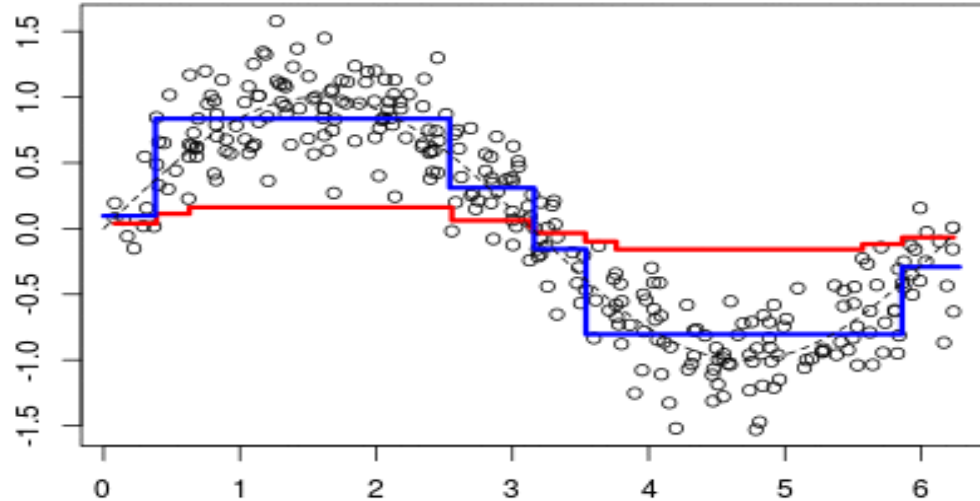
Qual o “truque” para ser um bom cientista de dados?

- Intuição
- Viés e Variância
- Erro de treino e erro de teste
- Validação cruzada

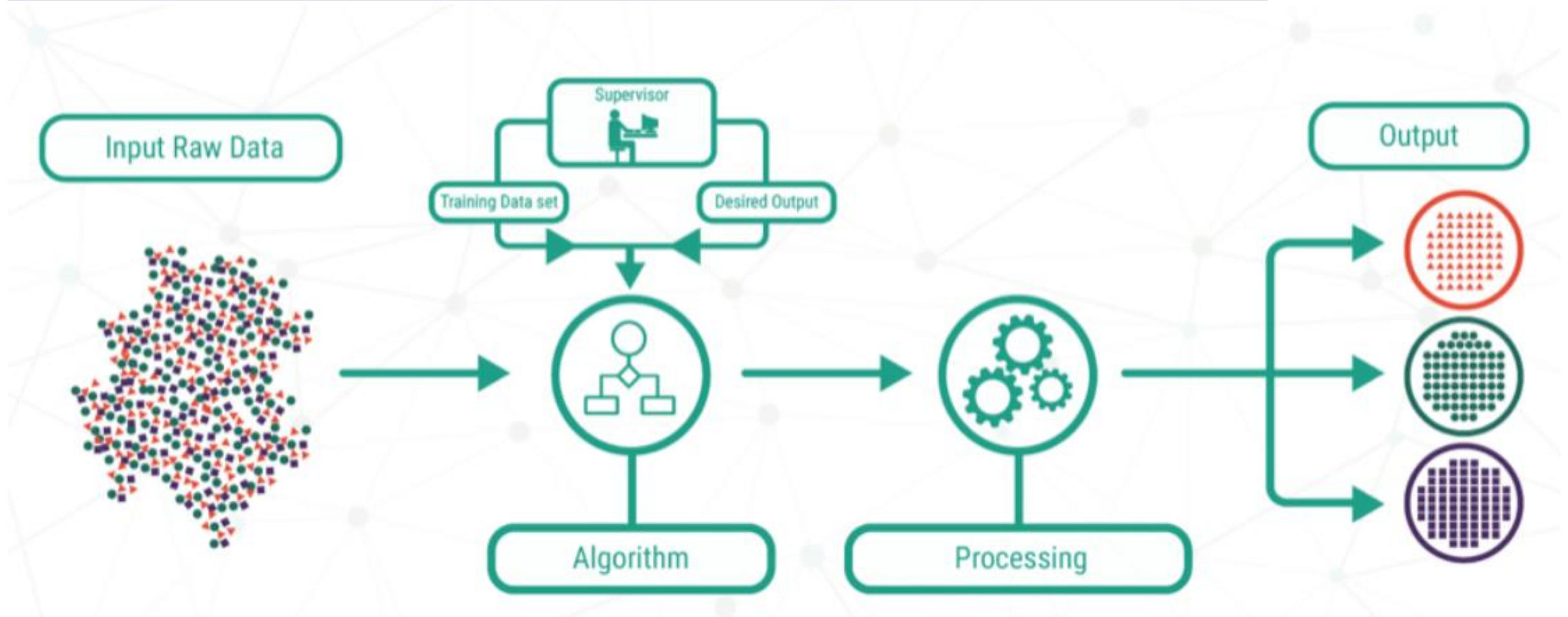


Machine Learning

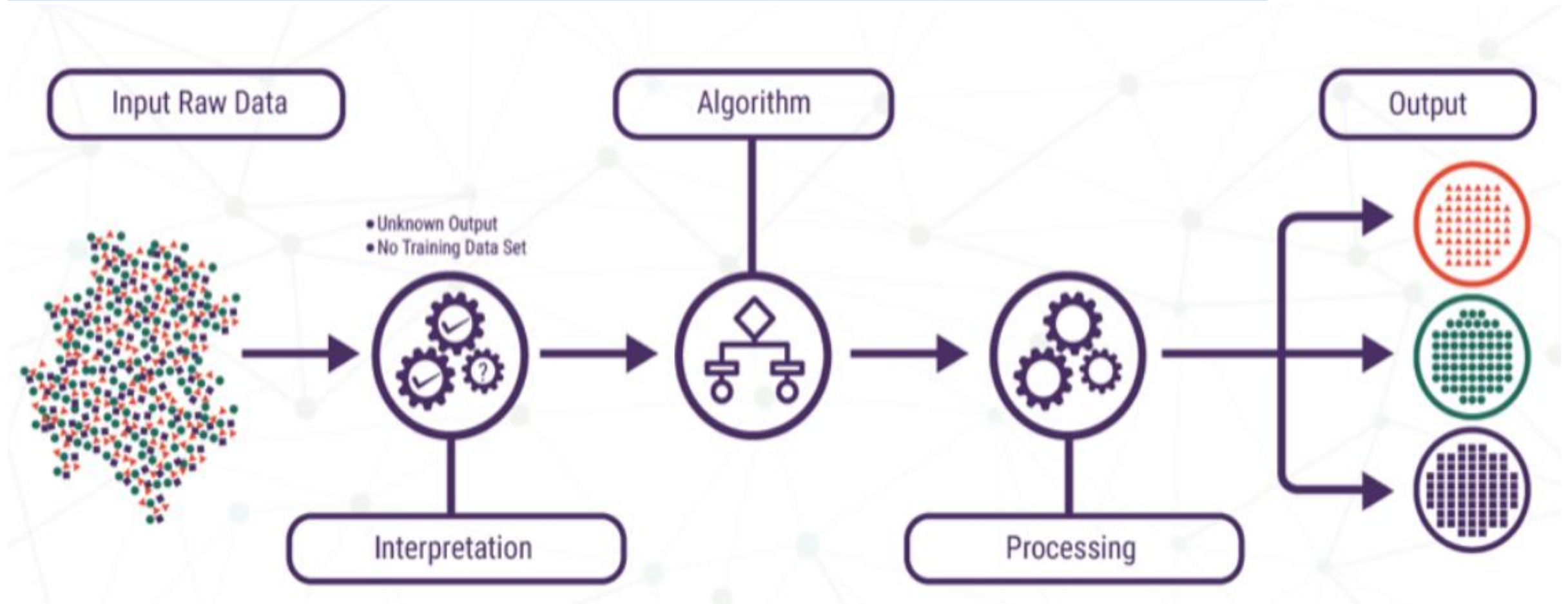
- Intuição
- Viés e Variância
- Erro de treino e erro de teste
- Validação cruzada



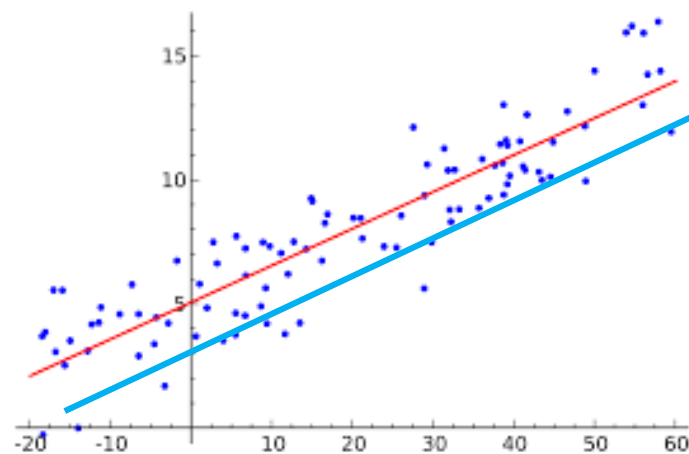
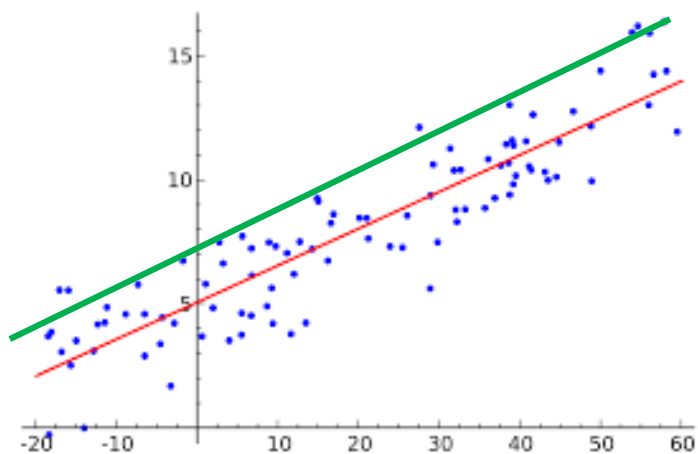
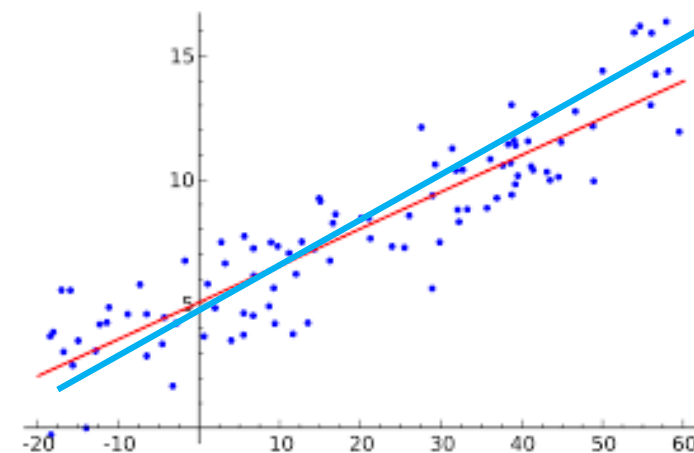
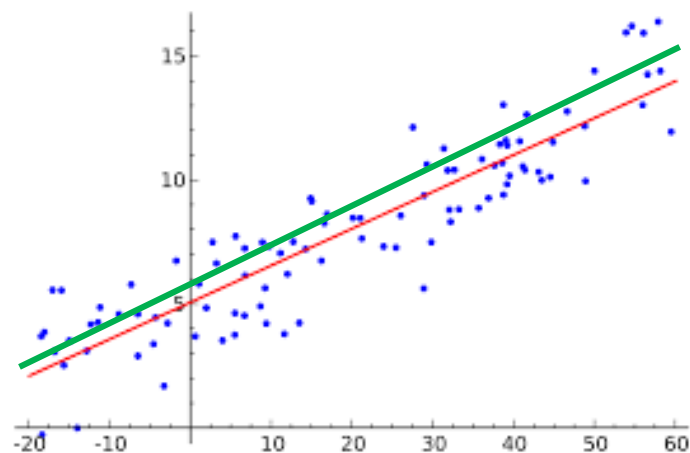
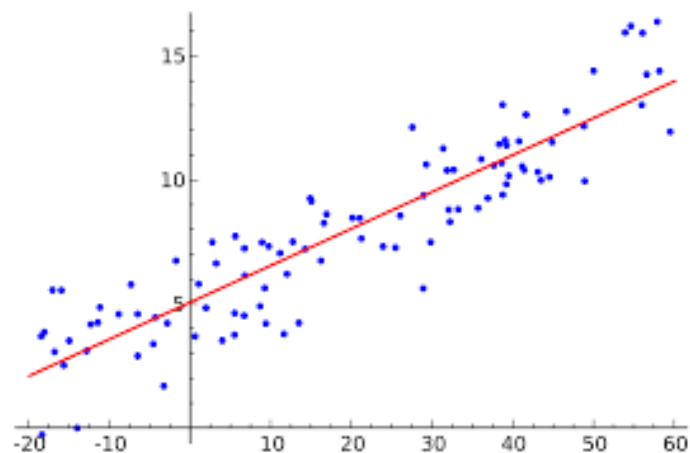
Machine Learning – Supervised Learning



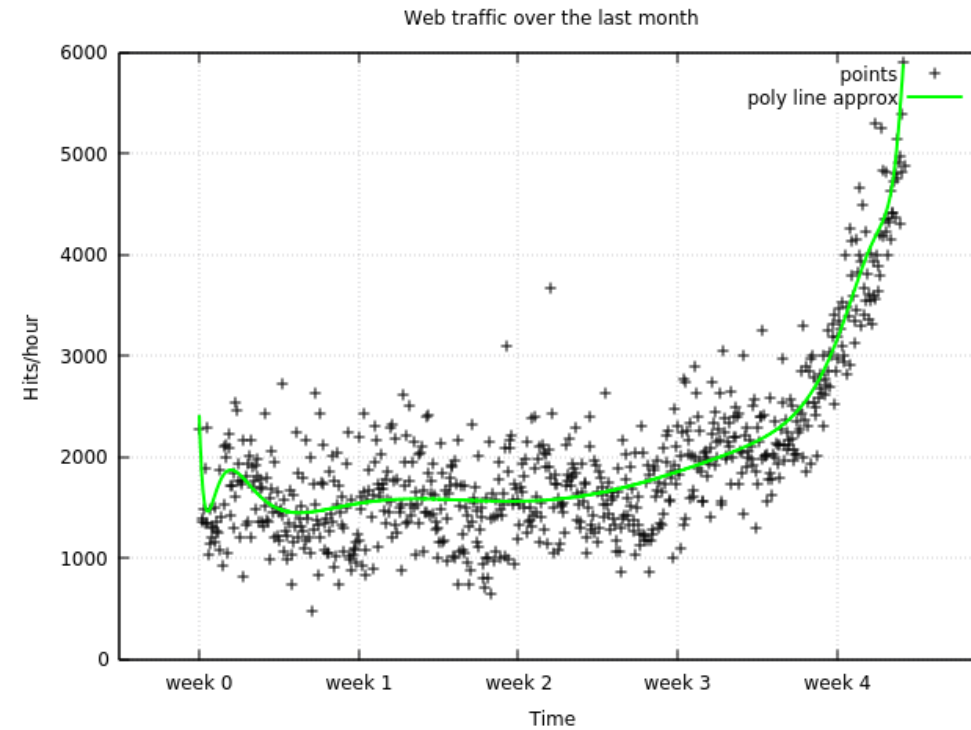
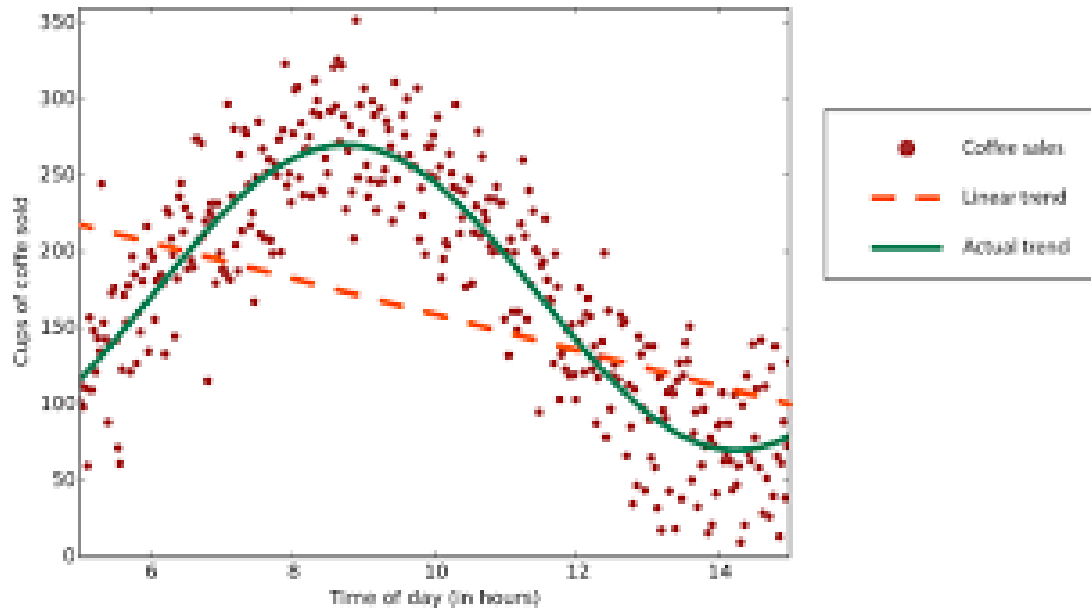
Machine Learning – Unsupervised Learning



Machine Learning – Regressão

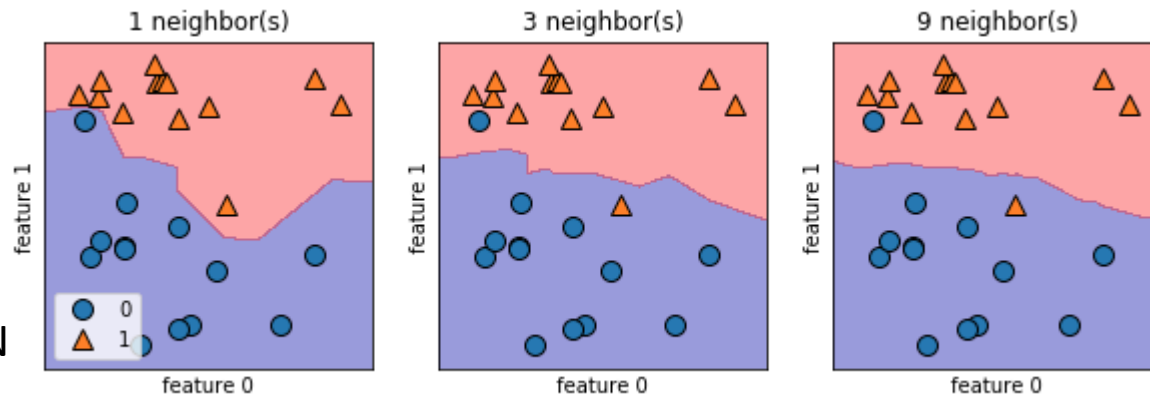


Machine Learning – Regressão

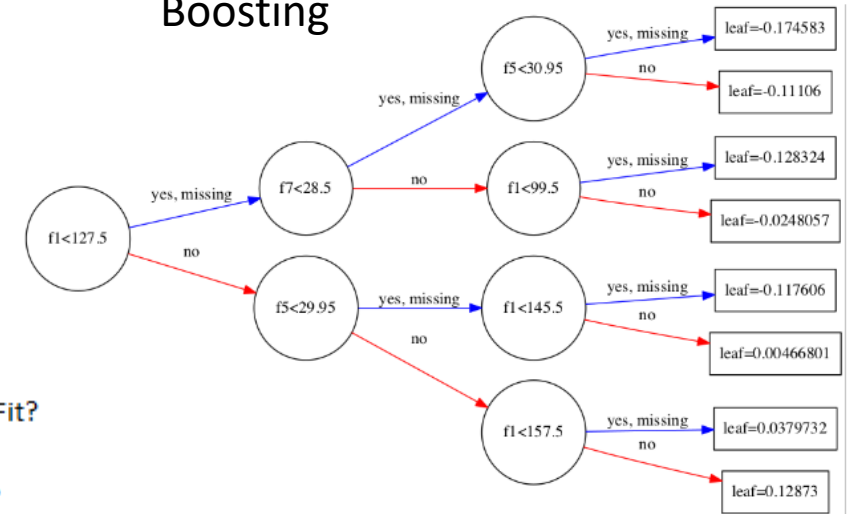


Machine Learning – Classificação

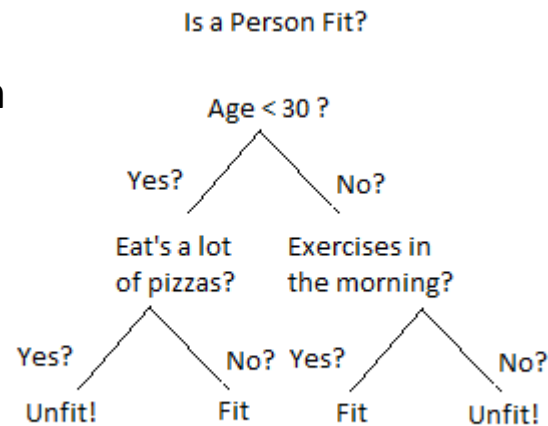
KNN



Gradient Boosting

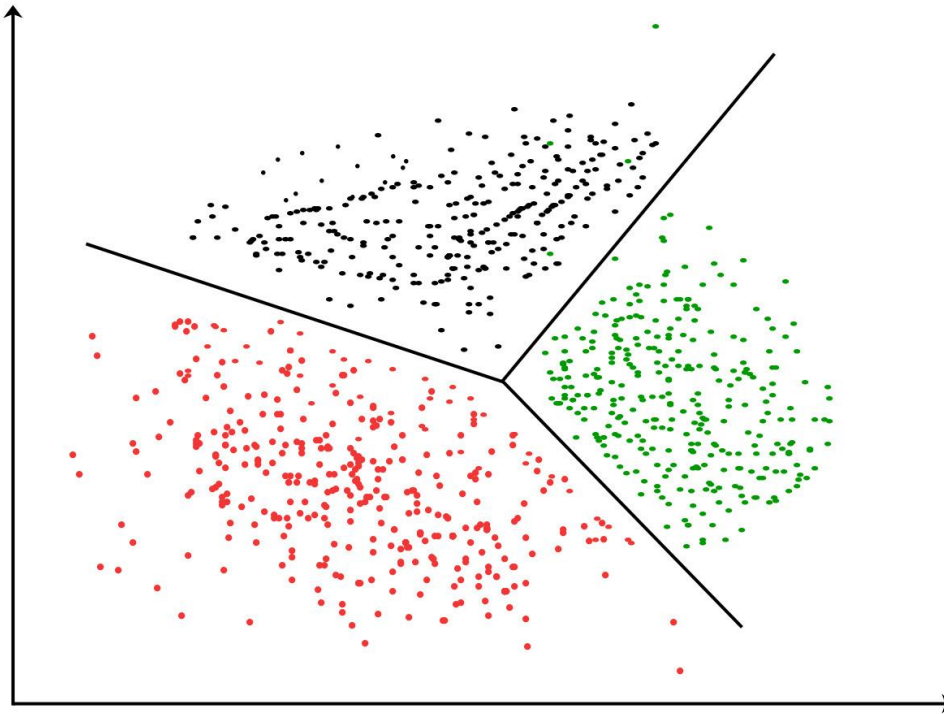


Decision Tree



Machine Learning – Agrupamento

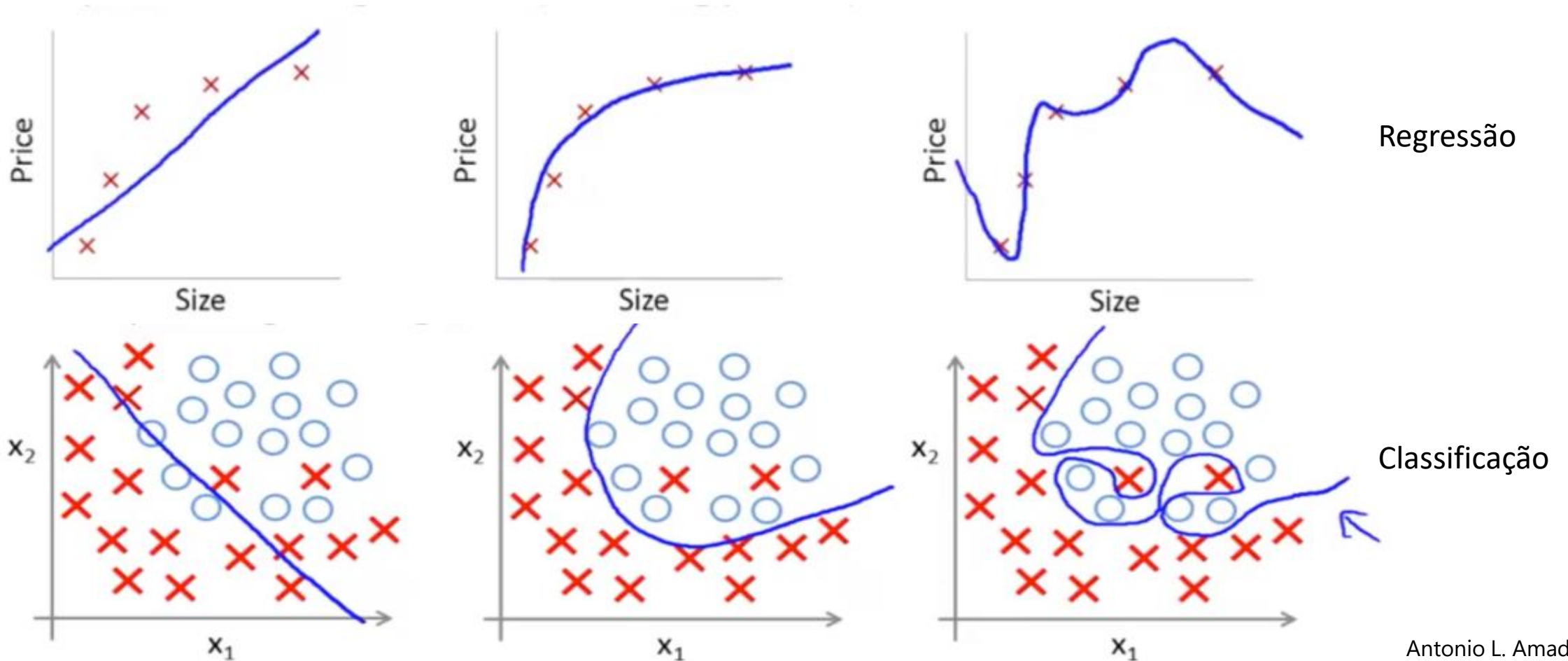
K - Means



Machine Learning – Agrupamento - Prática



Machine Learning – Underfitting e Overfitting



Preparação dos dados

- Missing Data
 - Remover as linhas
 - Acrescentar valores médios
 - Recorrer à regressão linear
- Outliers
 - Remover as linhas
 - Média
- Variáveis categóricas
 - One Hot Encoding *
- Normalização
 - Colocar todos os valores entre determinada faixa de valores, por exemplo, idade e altura*

Preparação dos dados – Variáveis categóricas - One Hot Encoding

Temos as seguintes features:

['color','size','price'] onde temos como 'color': [Azul,Verde,Vermelho']

Transformando a 'feature categórica 'color' para One Hot Encoding:

[Azul, Verde, Vermelho] – Headers

[1 , 0 , 0] – Azul

[0 , 1 , 0] - Verde

[0 , 0 , 1] - Vermelho

Preparação dos dados – Variáveis categóricas - One Hot Encoding

```
1  from numpy import array
2  from numpy import argmax
3  from sklearn.preprocessing import LabelEncoder
4  from sklearn.preprocessing import OneHotEncoder
5  # define example
6  data = ['cold', 'cold', 'warm', 'cold', 'hot', 'hot', 'warm', 'cold', 'warm', 'hot']
7  values = array(data)
8  print(values)
9  # integer encode
10 label_encoder = LabelEncoder()
11 integer_encoded = label_encoder.fit_transform(values)
12 print(integer_encoded)
13 # binary encode
14 onehot_encoder = OneHotEncoder(sparse=False)
15 integer_encoded = integer_encoded.reshape(len(integer_encoded), 1)
16 onehot_encoded = onehot_encoder.fit_transform(integer_encoded)
17 print(onehot_encoded)
18 # invert first example
19 inverted = label_encoder.inverse_transform([argmax(onehot_encoded[0, :])])
20 print(inverted)
```

Preparação dos dados – Normalização

A técnica estatística mais comum e mais usada é:

$$z = \frac{x - \mu}{\sigma}$$

Onde:

z = Valor normalizado

x = Valor não normalizado

μ = Média dos valores

σ = Desvio Padrão

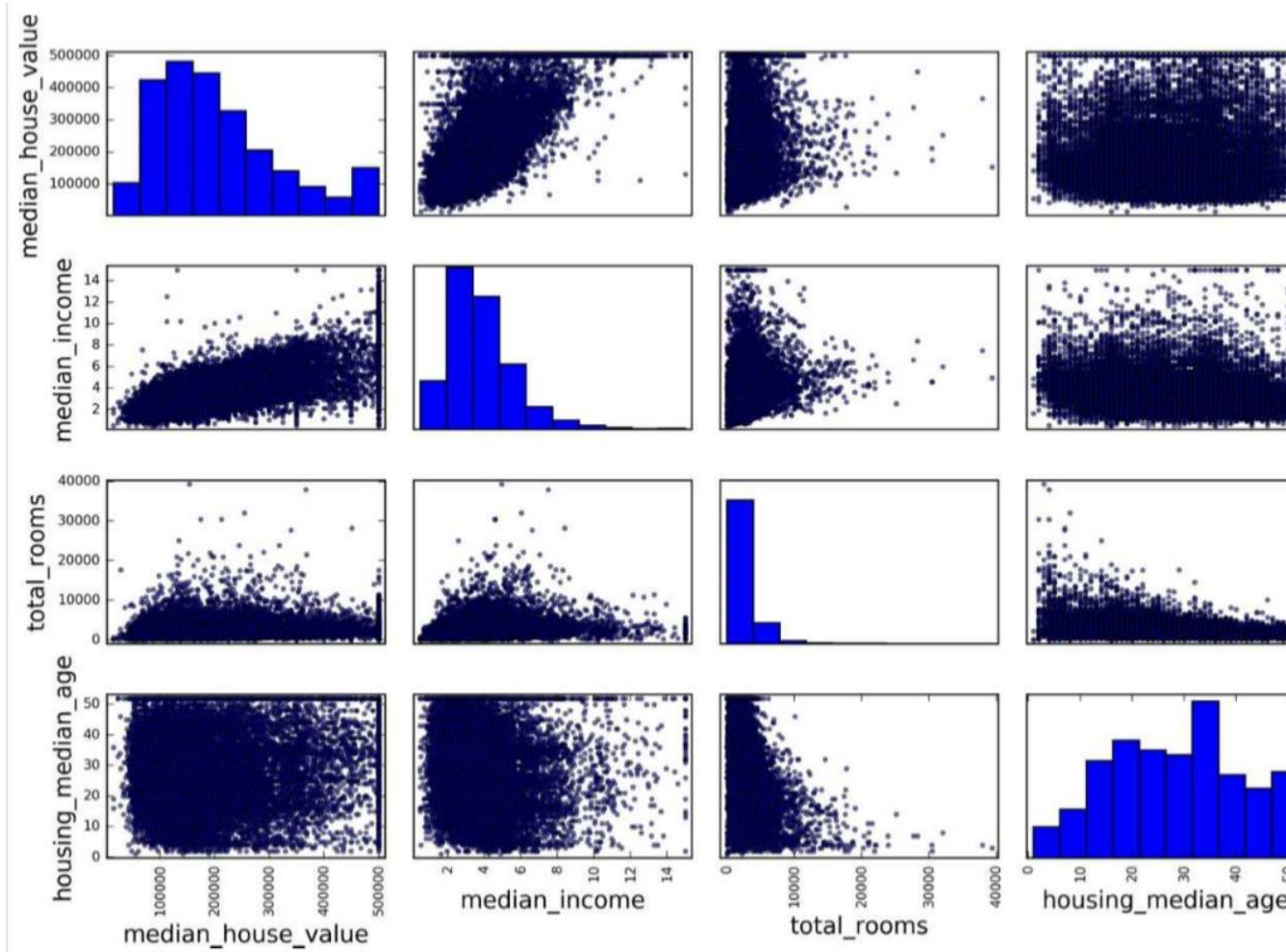
Desvio Padrão:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

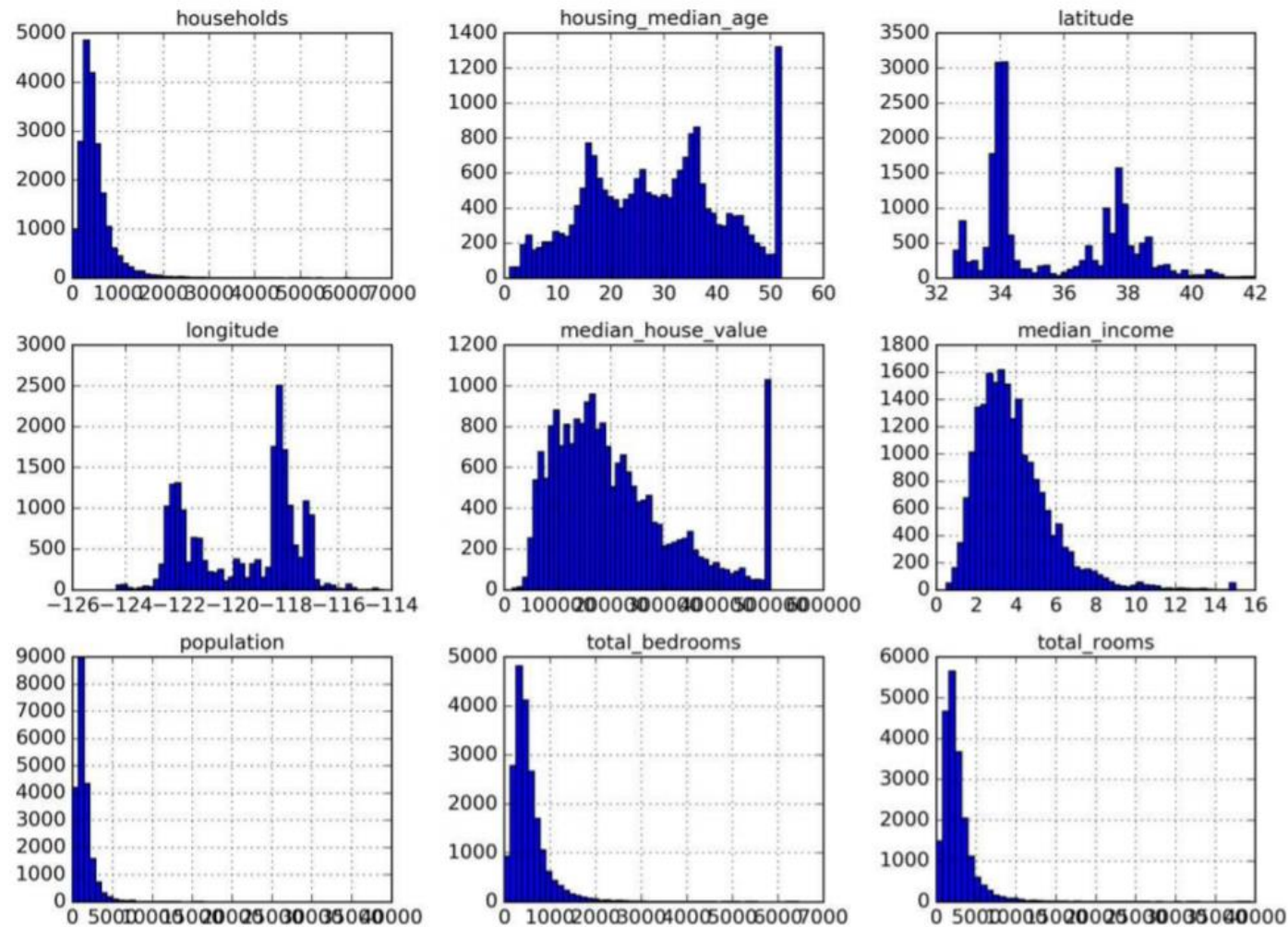
→ Variância

```
3  # Normalize the data attributes for the Iris dataset.
4  from sklearn.datasets import load_iris
5  from sklearn import preprocessing
6  # load the iris dataset
7  iris = load_iris()
8  print(iris.data.shape)
9  # separate the data from the target attributes
10 X = iris.data
11 y = iris.target
12 # normalize the data attributes
13 normalized_X = preprocessing.normalize(X)
```

Preparação dos dados – Definir features importantes - Correlação



Preparação dos dados – Definir features importantes - Histograma



Preparação dos dados – Definir features importantes

- PCA - <https://www.dezyre.com/data-science-in-python-tutorial/principal-component-analysis-tutorial>

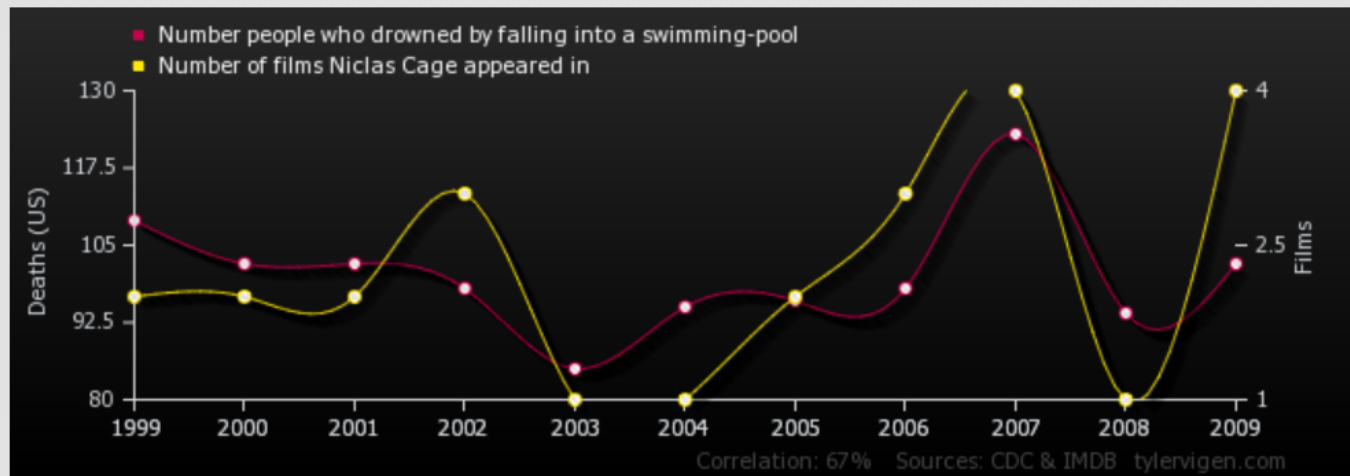


Correlações absurdas:

<http://tylervigen.com/old-version.html>

Preparação dos dados – Definir features importantes

Number people who drowned by falling into a swimming-pool
correlates with
Number of films Nicolas Cage appeared in

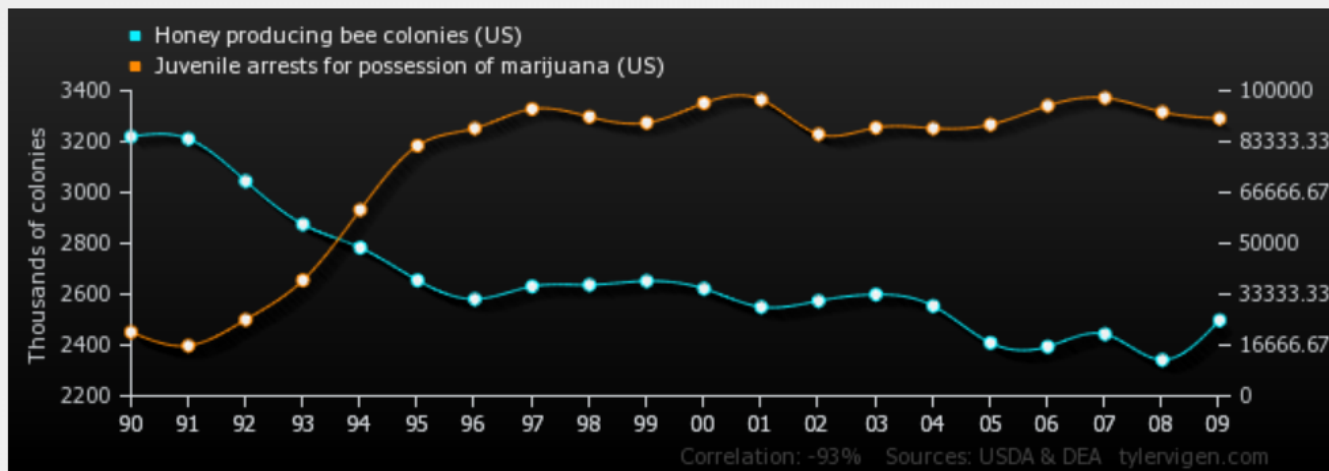


	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Number people who drowned by falling into a swimming-pool Deaths (US) (CDC)	109	102	102	98	85	95	96	98	123	94	102
Number of films Nicolas Cage appeared in Films (IMDB)	2	2	2	3	1	1	2	3	4	1	4

Correlation: 0.666004

Preparação dos dados – Definir features importantes

Honey producing bee colonies (US) inversely correlates with Juvenile arrests for possession of marijuana (US)



Honey producing bee colonies (US)
Thousands of colonies (USDA)

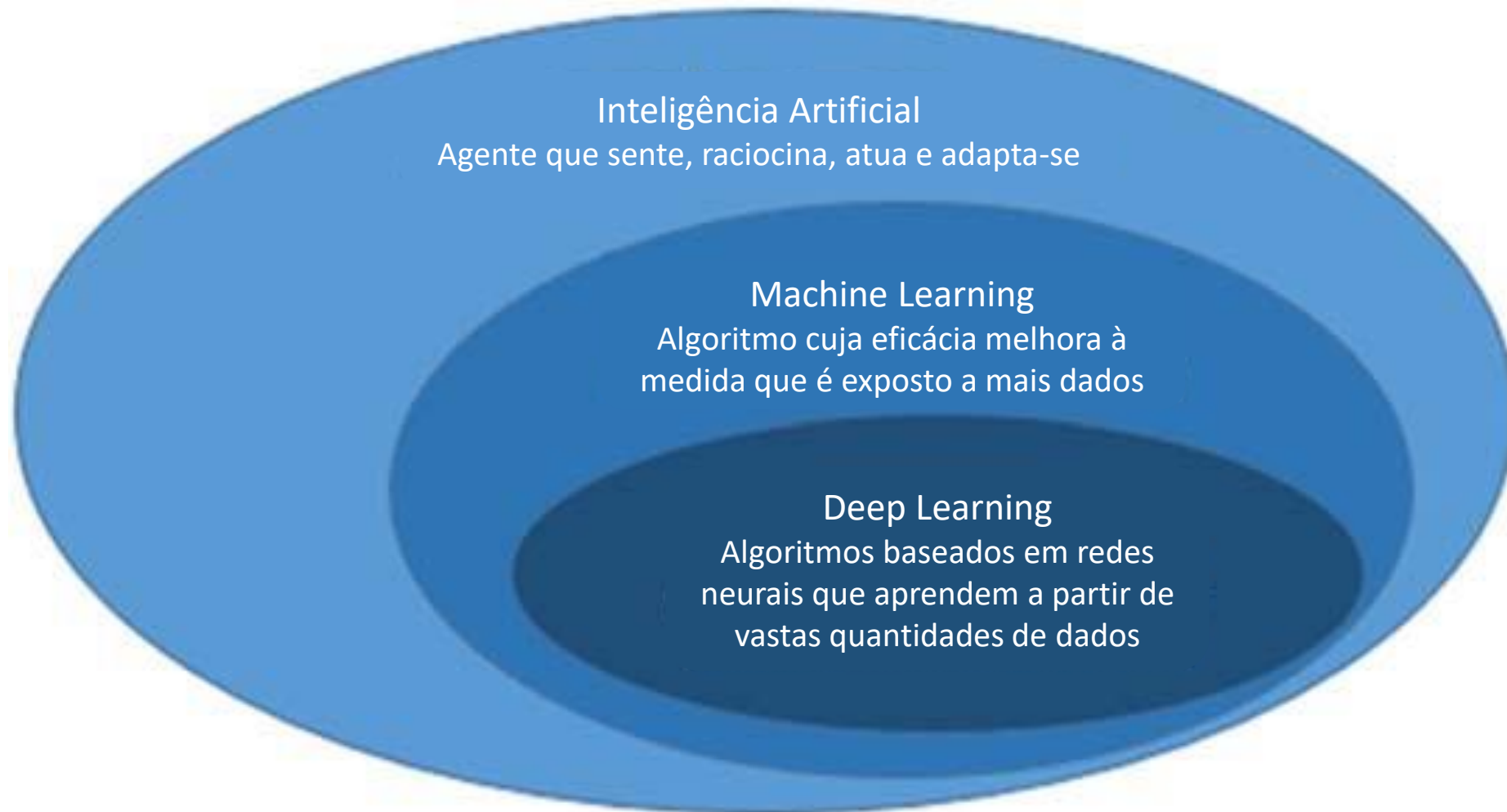
'90: 3,220; '91: 3,211; '92: 3,045; '93: 2,875; '94: 2,783; '95: 2,655; '96: 2,581; '97: 2,631; '98: 2,637; '99: 2,652; '00: 2,622; '01: 2,550; '02: 2,574; '03: 2,599; '04: 2,554; '05: 2,409; '06: 2,394; '07: 2,443; '08: 2,342; '09: 2,498

Juvenile arrests for possession of marijuana (US)
Arrests (DEA)

'90: 20,940; '91: 16,490; '92: 25,004; '93: 37,915; '94: 61,003; '95: 82,015; '96: 87,712; '97: 94,046; '98: 91,467; '99: 89,523; '00: 95,962; '01: 97,088; '02: 85,769; '03: 87,909; '04: 87,717; '05: 88,909; '06: 95,120; '07: 97,671; '08: 93,042; '09: 90,927

Correlation: -0.933389

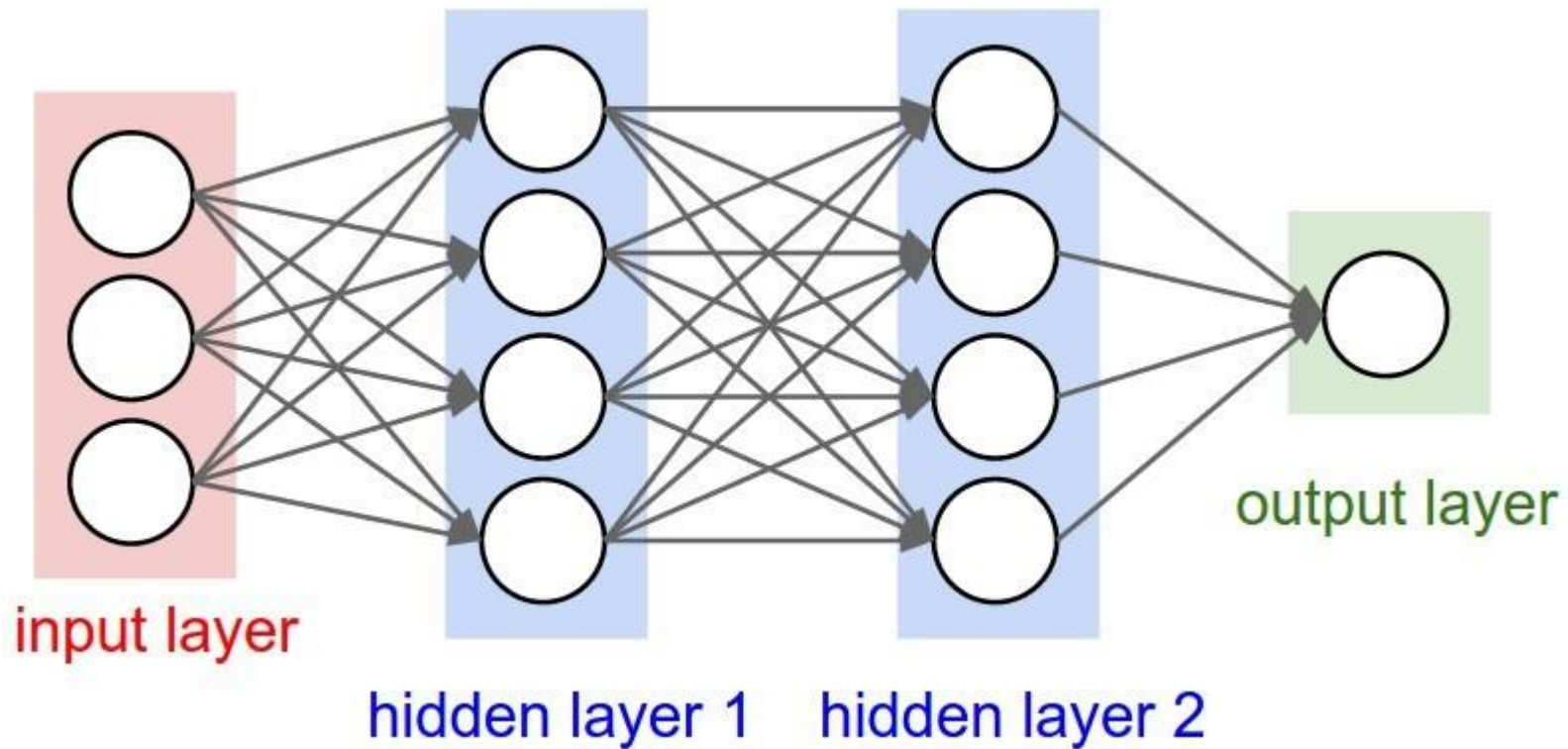
Deep Learning



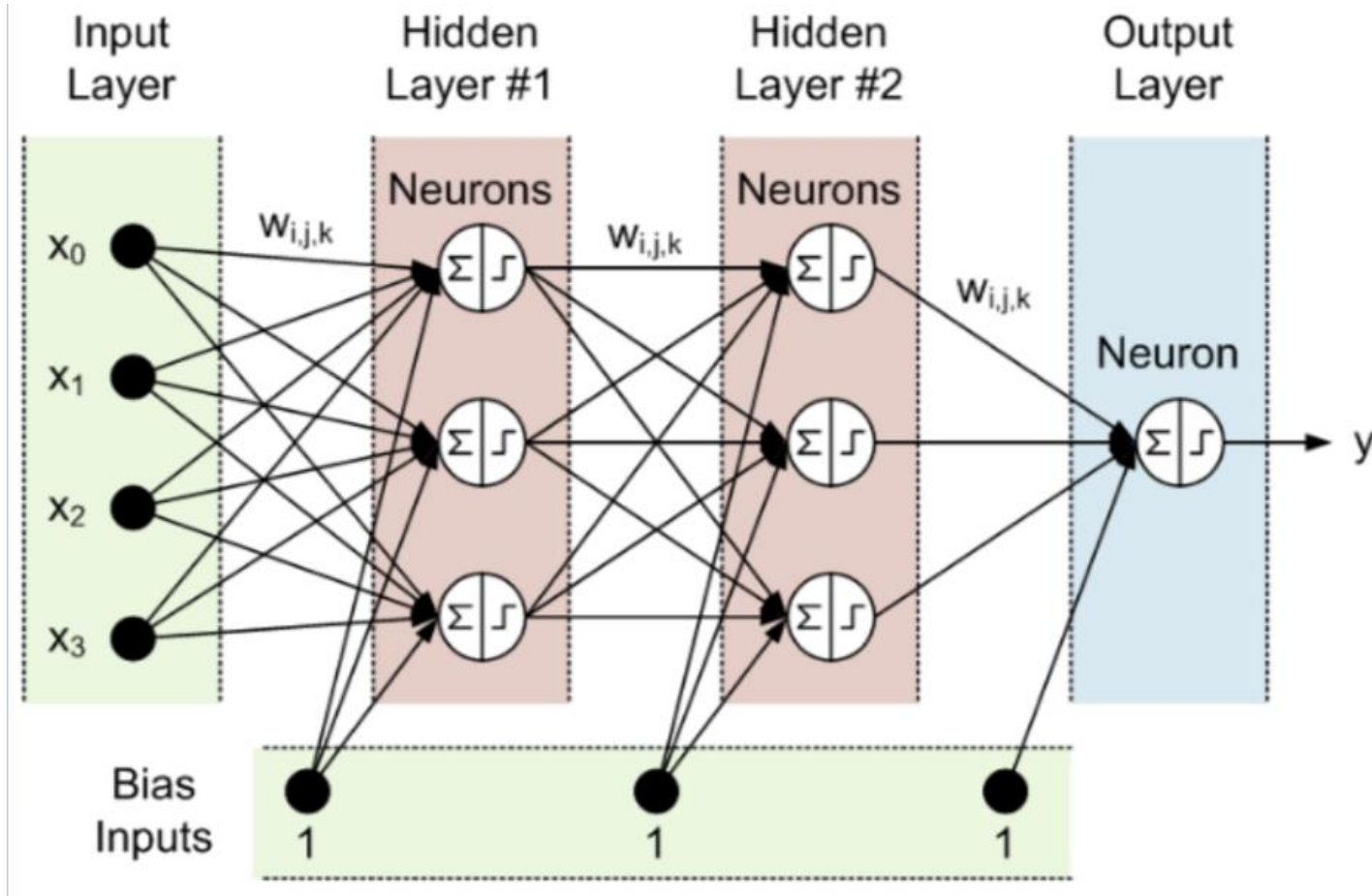
Deep Learning

Bibliotecas Python para Deep Learning:

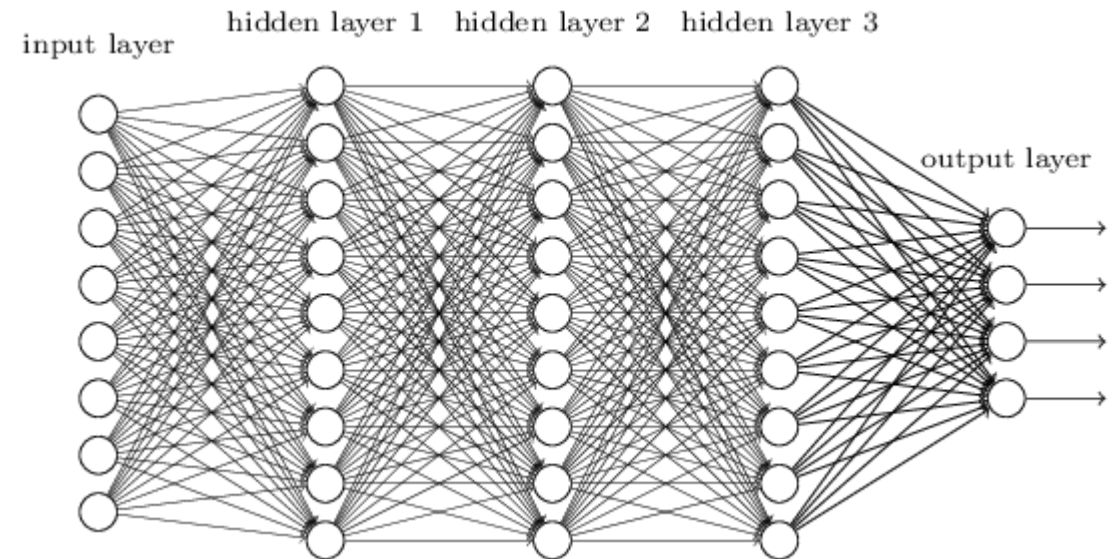
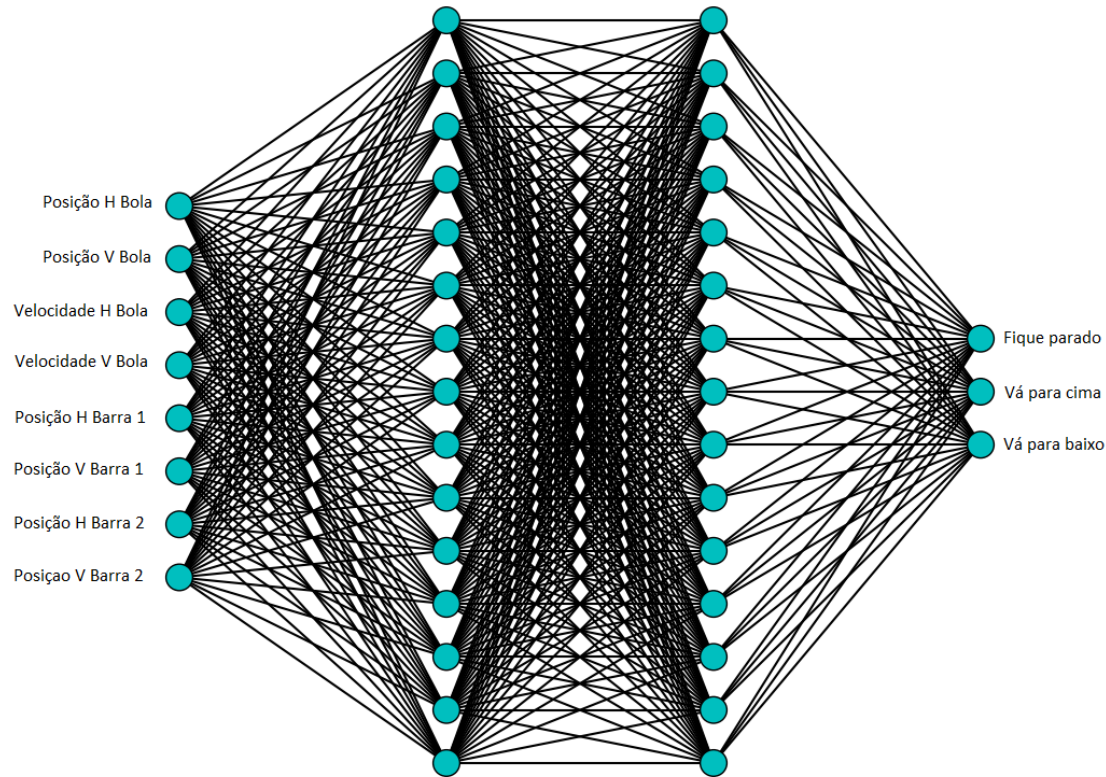
- Keras
- Tensorflow



Deep Learning



Deep Learning

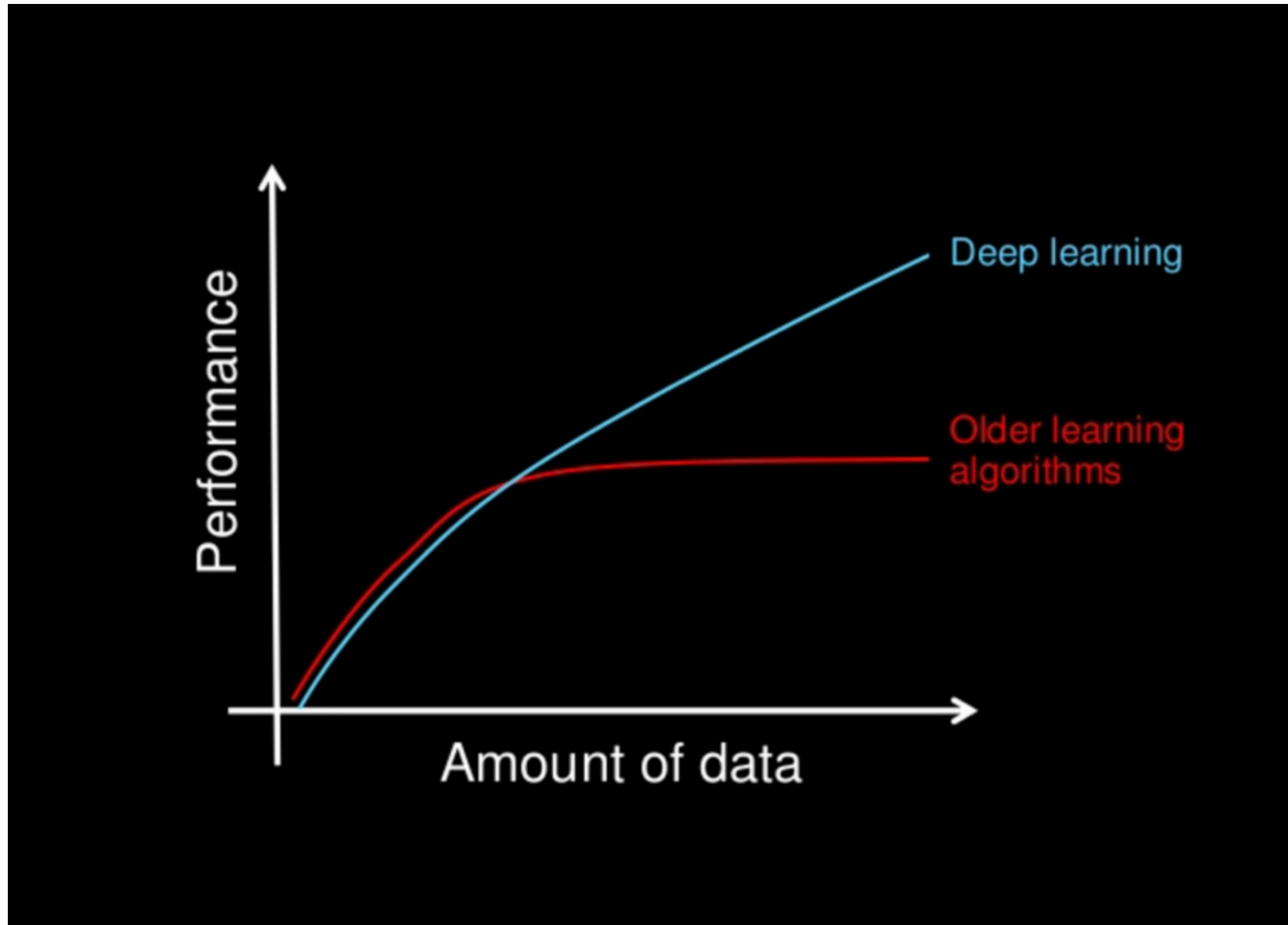


<https://pong-2.com/>

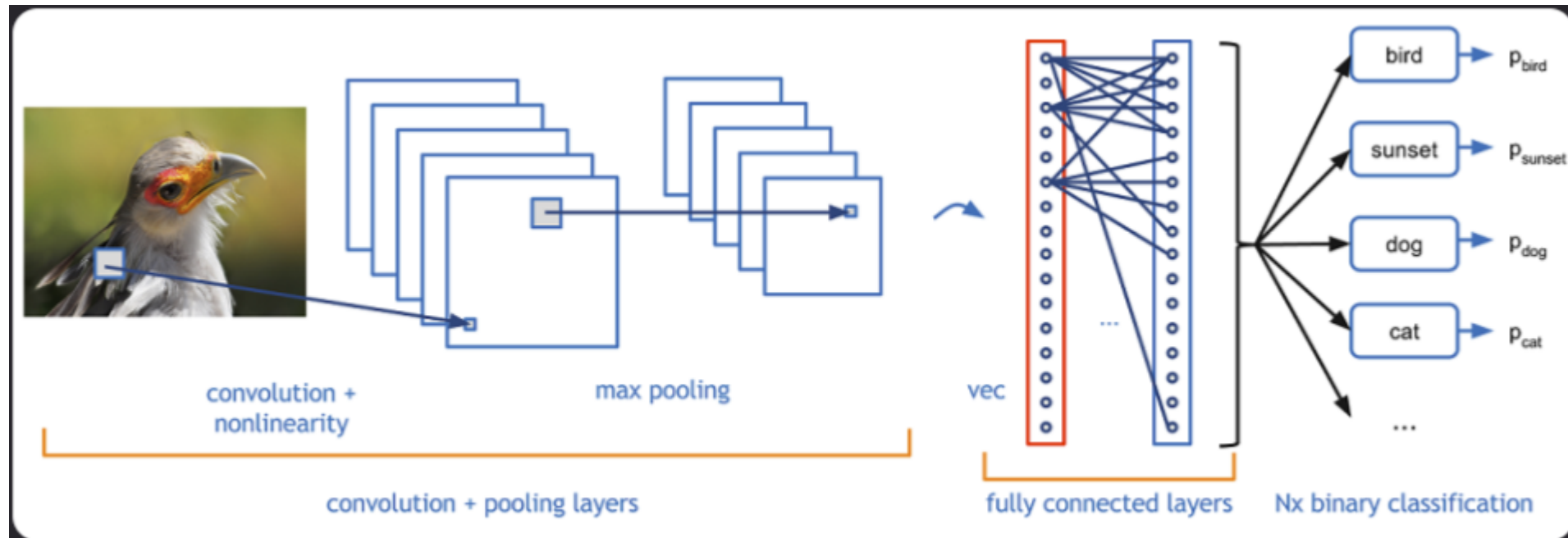
<https://medium.com/pilotorobo/pythonjogapong-parte-1-capturando-a-tela-e-extraindo-informa%C3%A7%C3%B5es-334135880144>

https://github.com/pilotorobo/pongplay/blob/master/screen_features.py

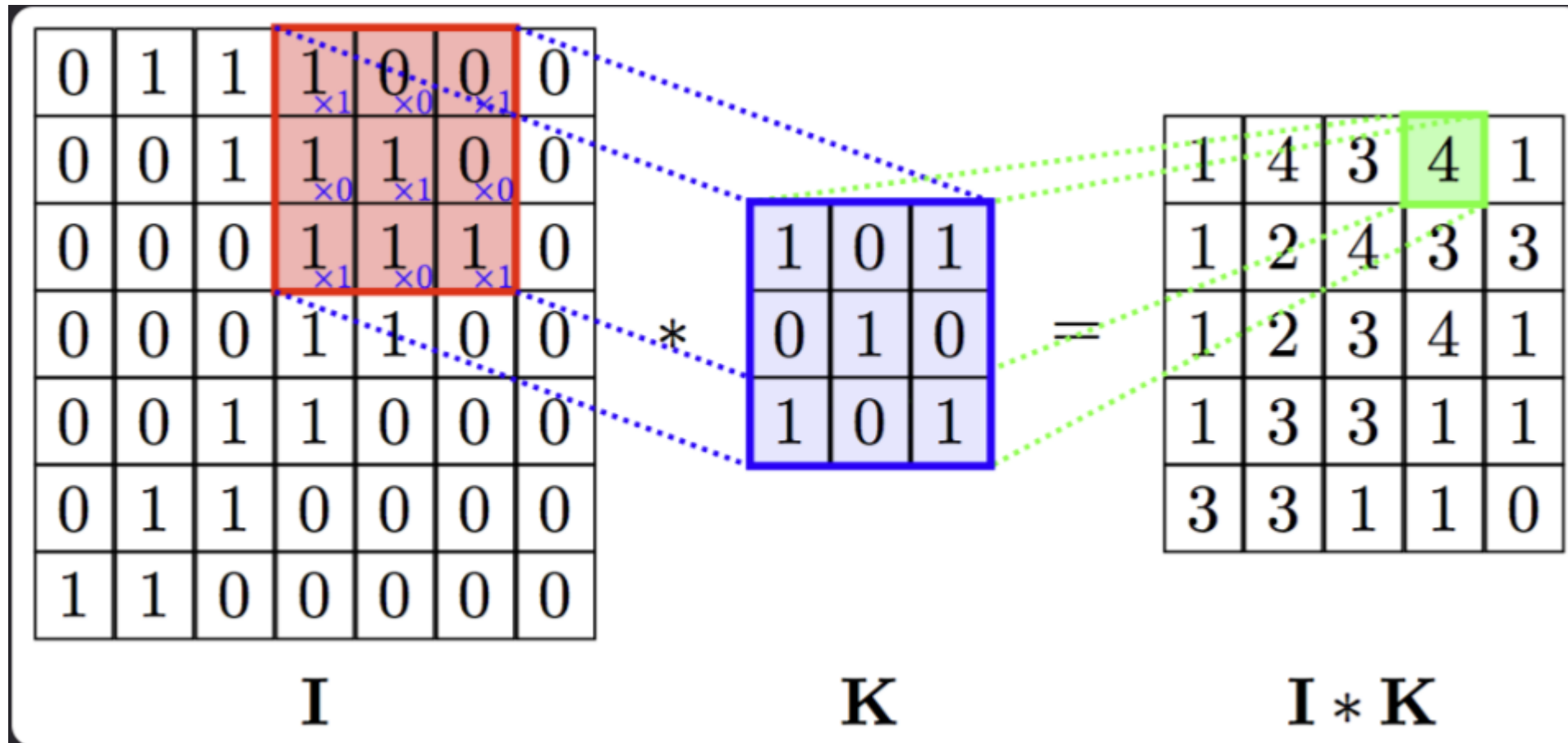
Machine Learning x Deep Learning



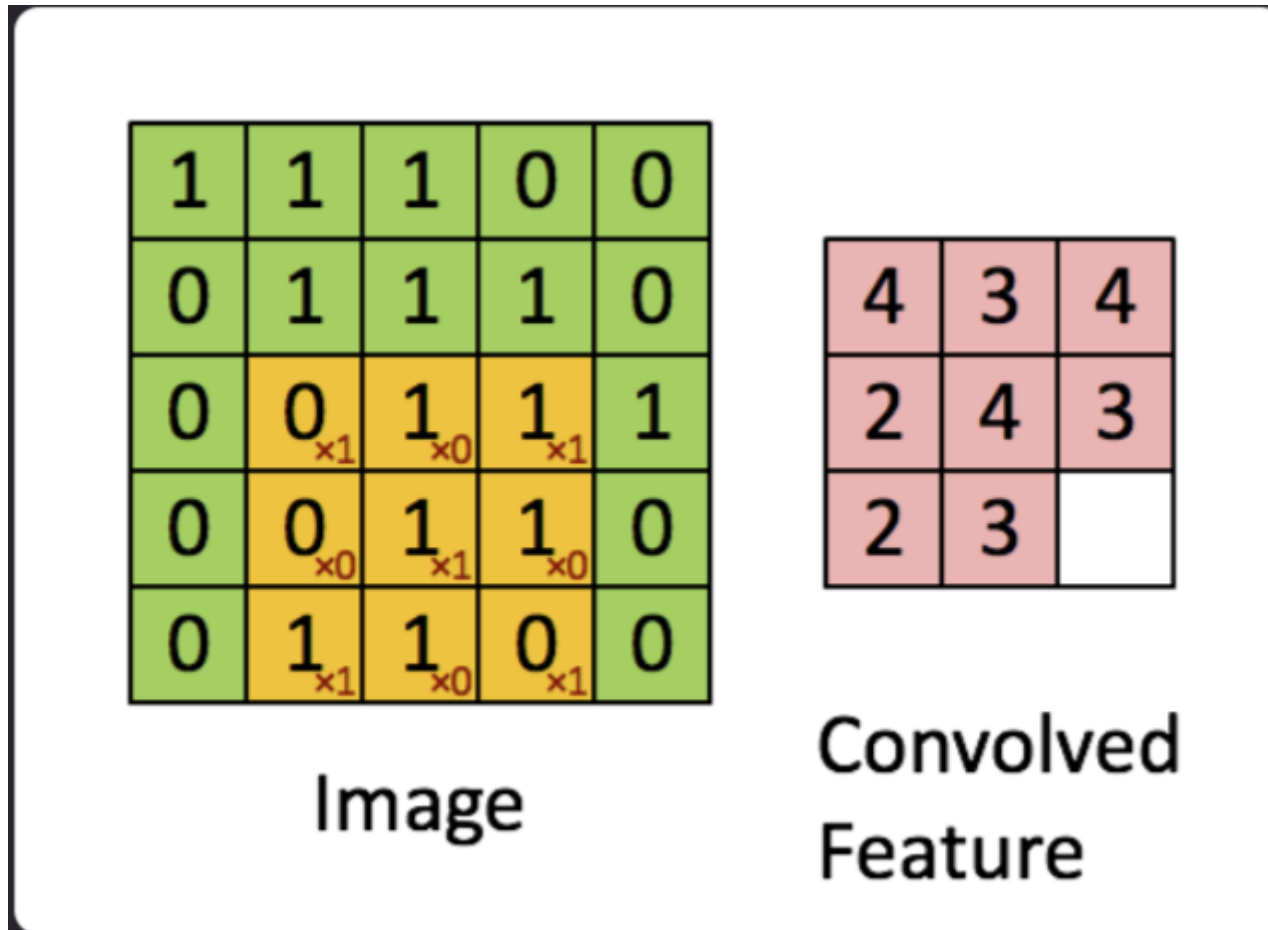
Deep Learning – Convolutional Neural Network












Deep Learning – Convolutional Neural Network



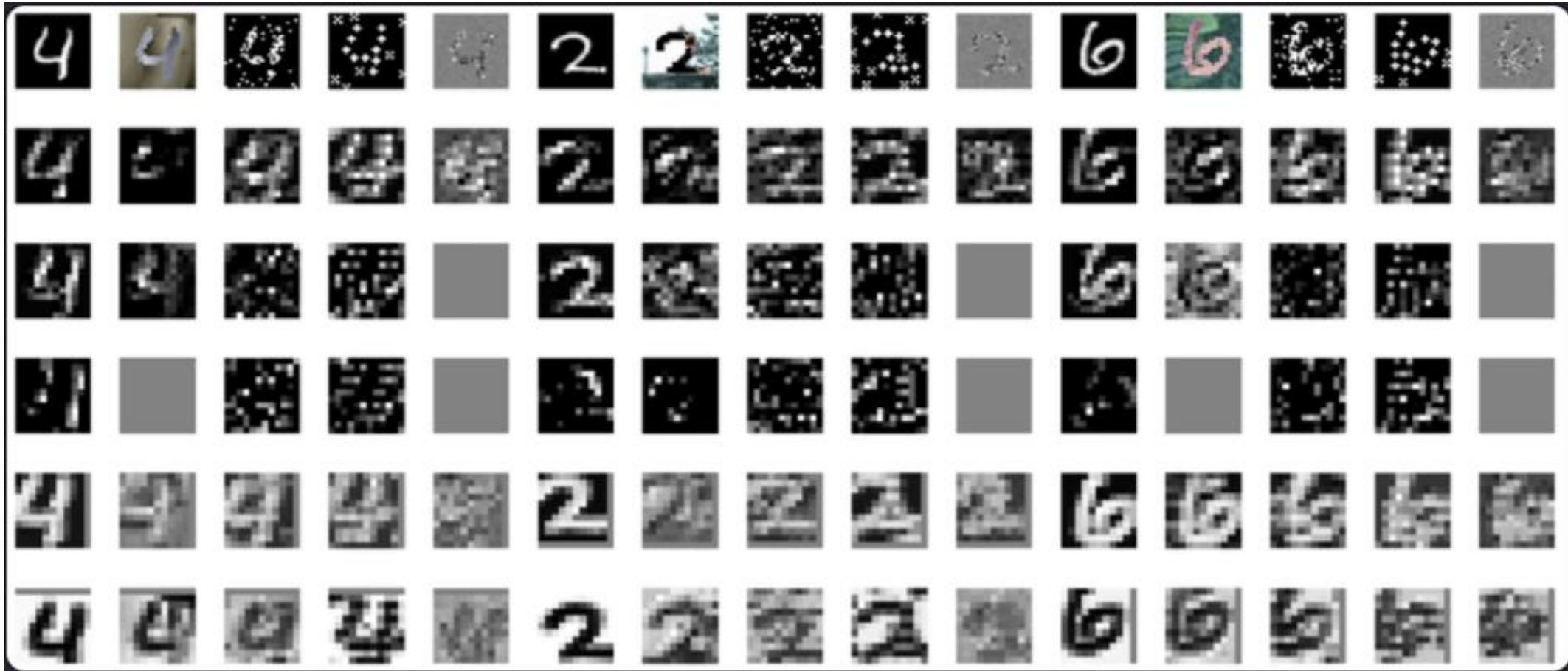
Deep Learning – Convolutional Neural Network



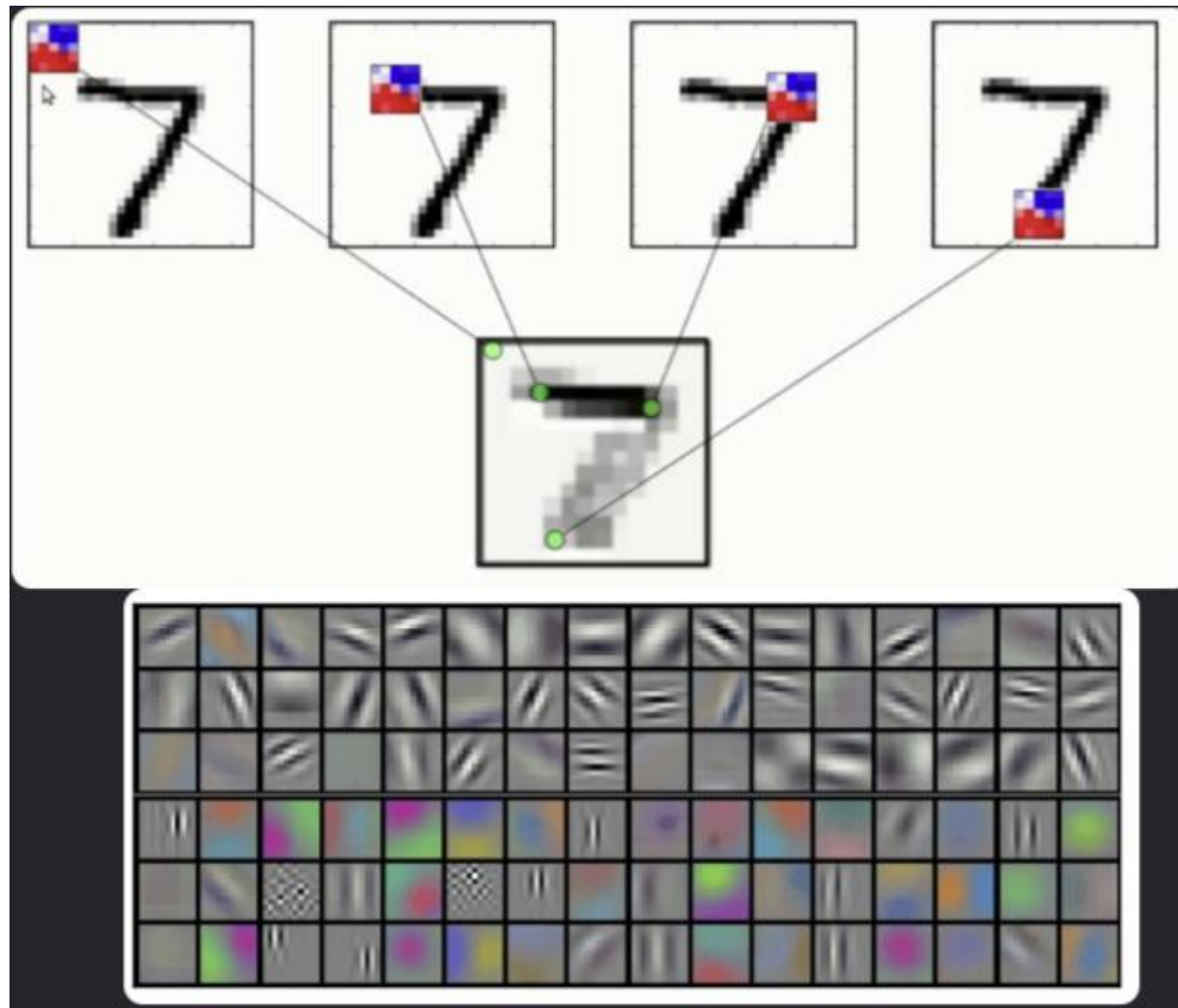
Deep Learning – Convolutional Neural Network

Operation	Kernel ω	Image result $g(x,y)$			
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$		Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$		Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$		Gaussian blur 3 x 3 (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$		Gaussian blur 5 x 5 (approximation)	$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	
			Unsharp masking 5 x 5 Based on Gaussian blur with amount as 1 and threshold as 0 (with no image mask)	$\frac{-1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & -476 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	

Deep Learning – Convolutional Neural Network



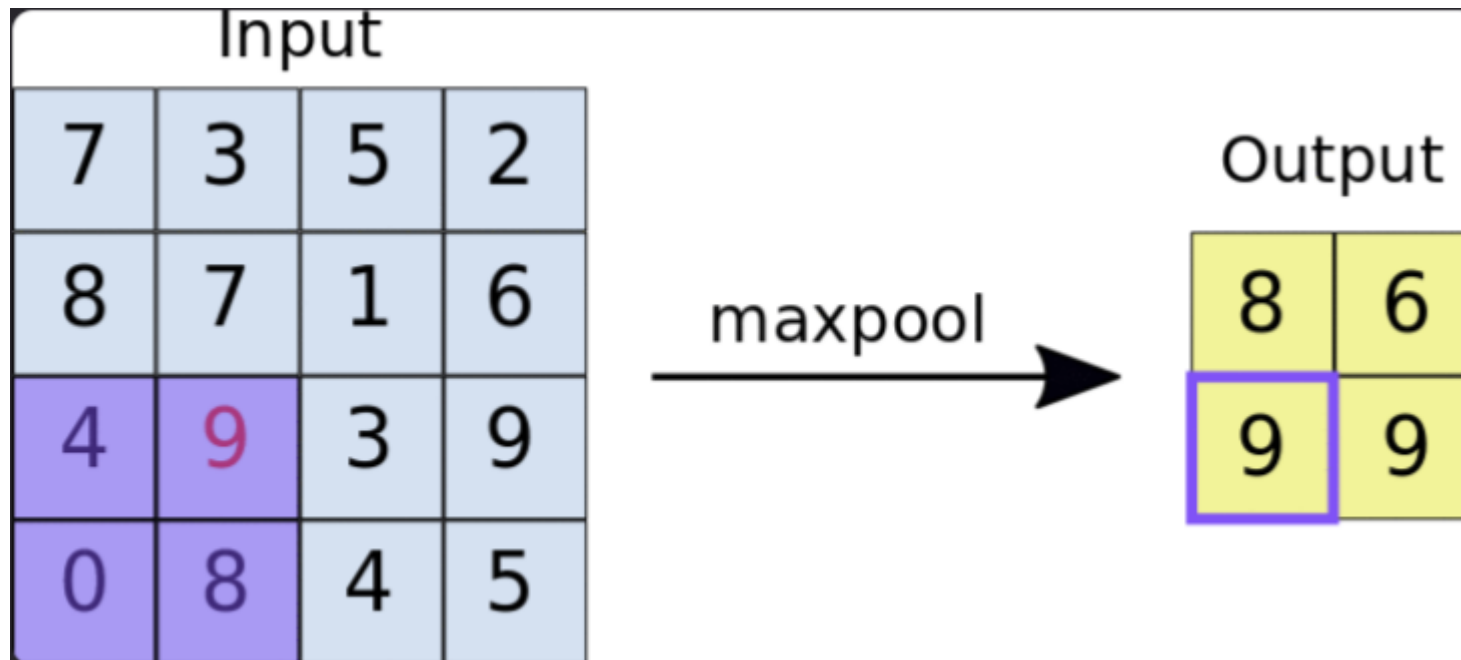
Deep Learning – Convolutional Neural Network



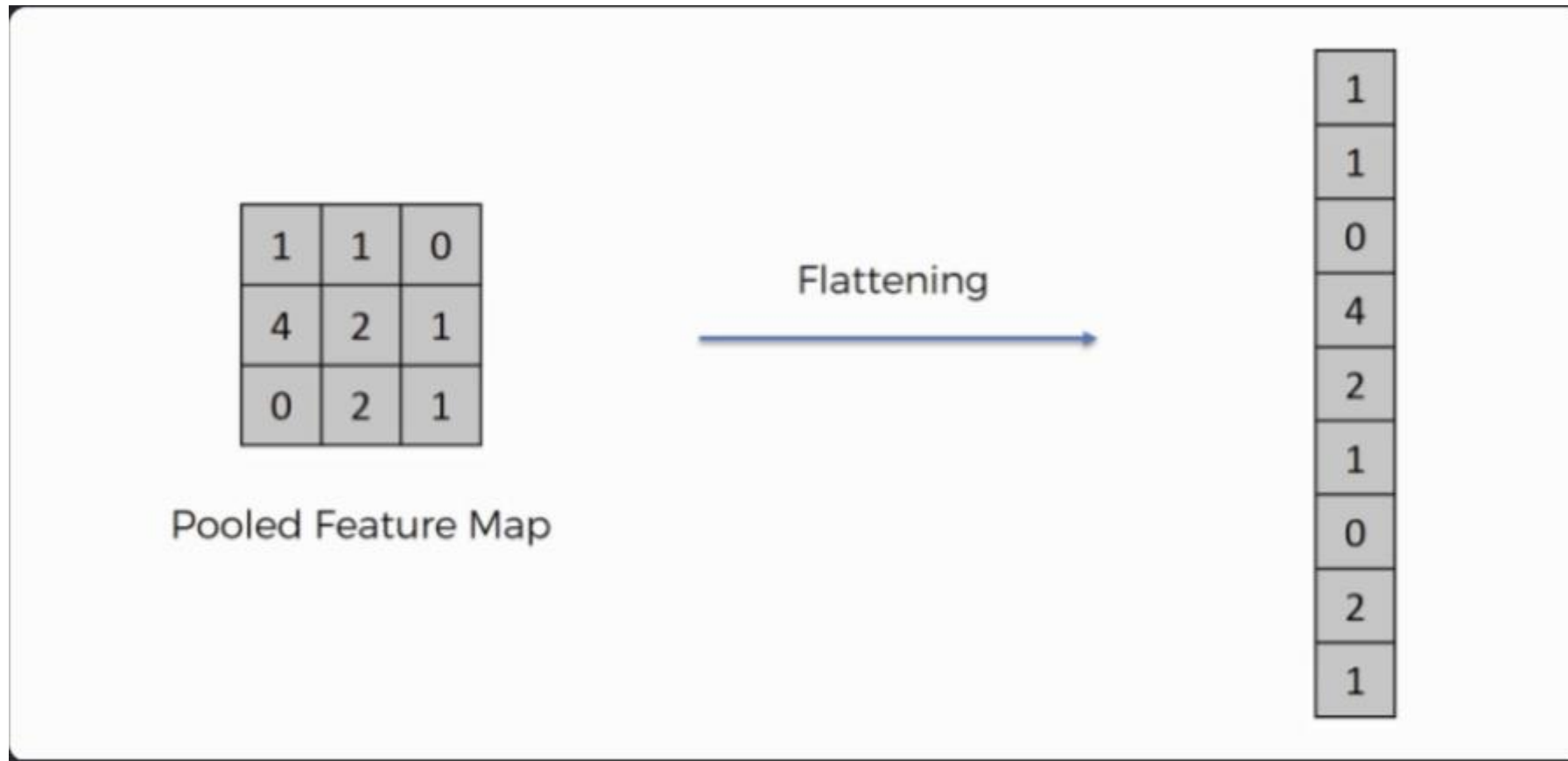
Deep Learning – Convolutional Neural Network



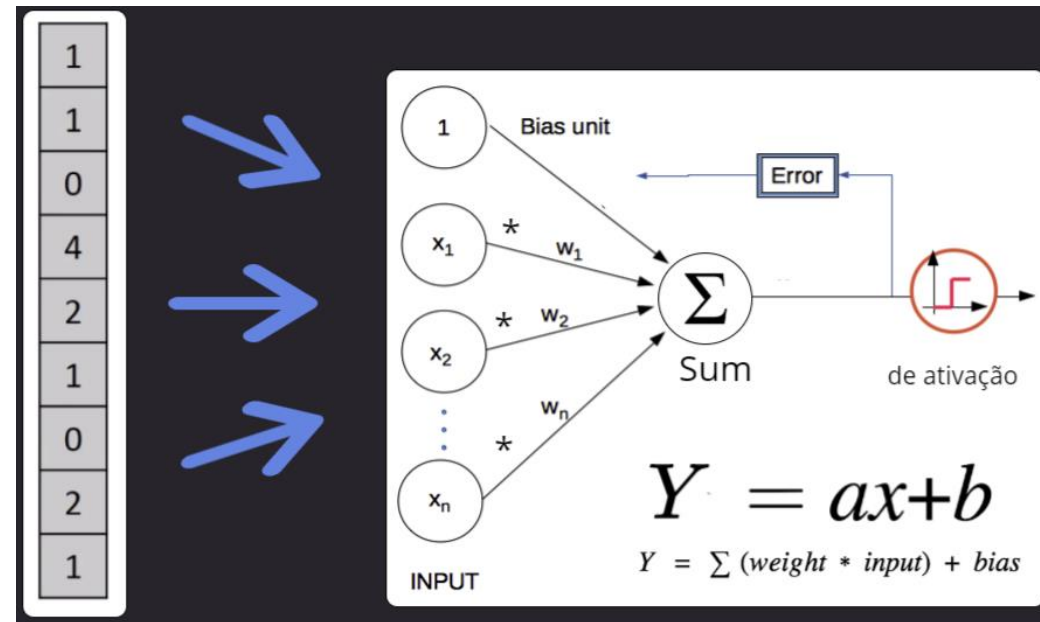
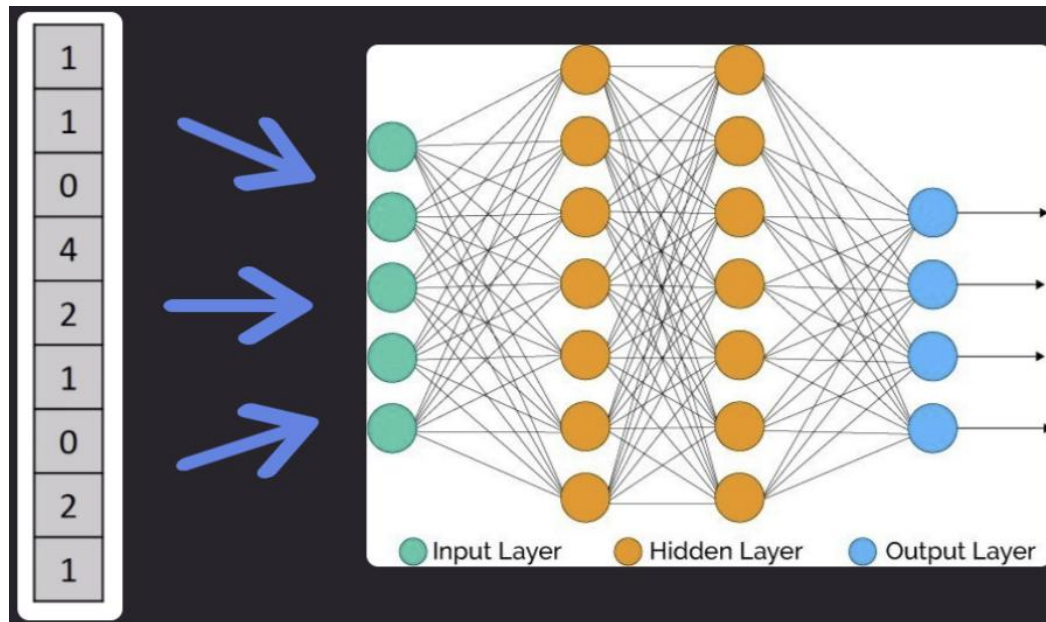
Deep Learning – Convolutional Neural Network



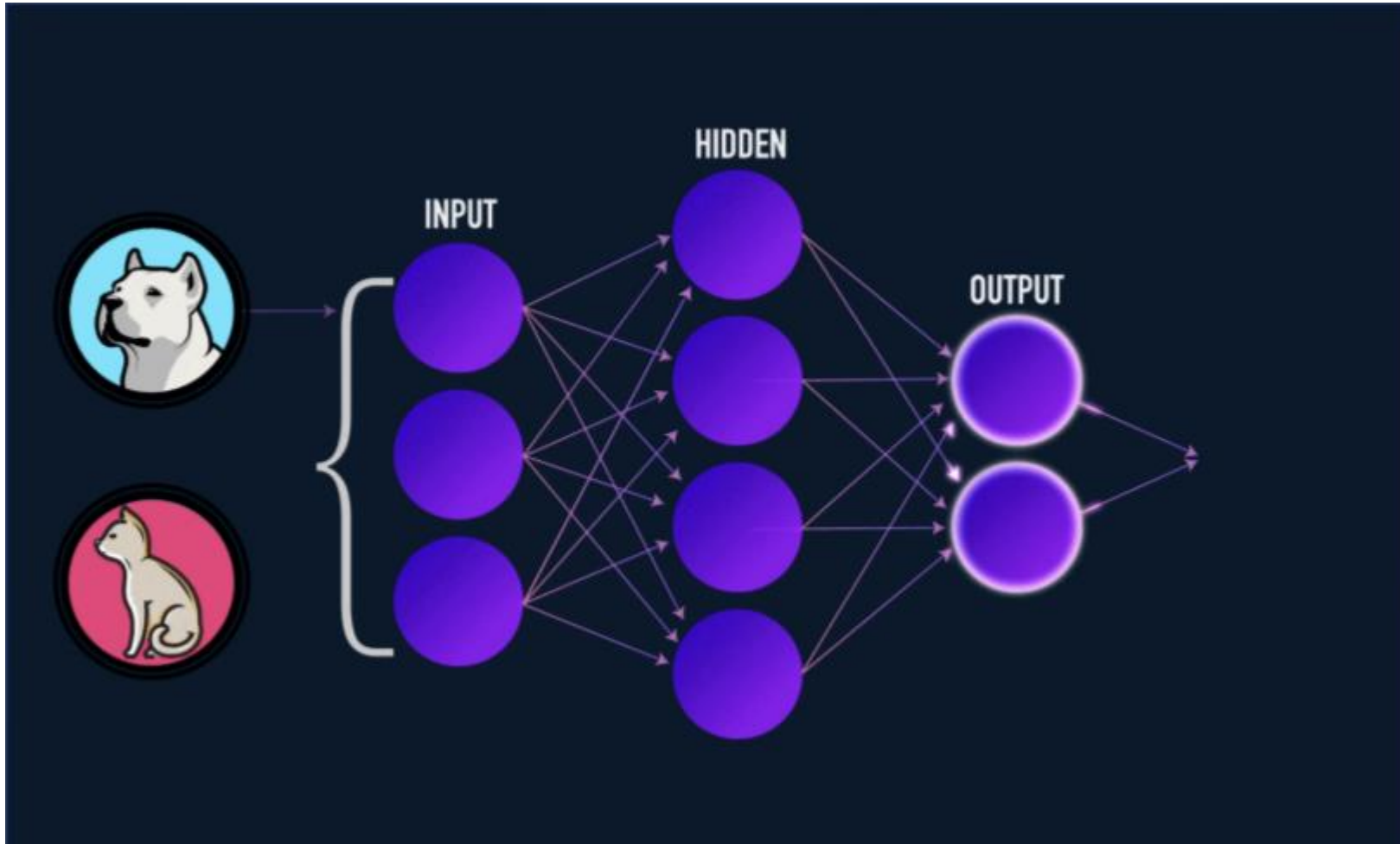
Deep Learning – Convolutional Neural Network



Deep Learning – Convolutional Neural Network



Deep Learning – Convolutional Neural Network



Deep Learning

Arquiteturas de Deep Learning:

- Restricted Boltzmann Machines (RBM)
 - Classificação de imagens
 - Reconhecimento de fala
- Autoencoders
 - Fraudes
 - Análise de comportamento
- Convolutional Neural Networks (CNN)
 - Classificação de imagens
- Recurrent Neural Networks (RNN)
 - Classificação de sentimento
 - Geração de texto
 - Predição de preço de ações
- Generative Adversarial Networks (GANs)
 - Geração de imagem, audio, video, etc.
 - <https://thispersondoesnotexist.com/>

Meus heróis:



Deep Learning

Peter Norvig – Diretor de pesquisa no Google



Yoshua Bengio – Universidade de Montreal



Geoffrey E. Hinton – Google Brain e Universidade de Toronto

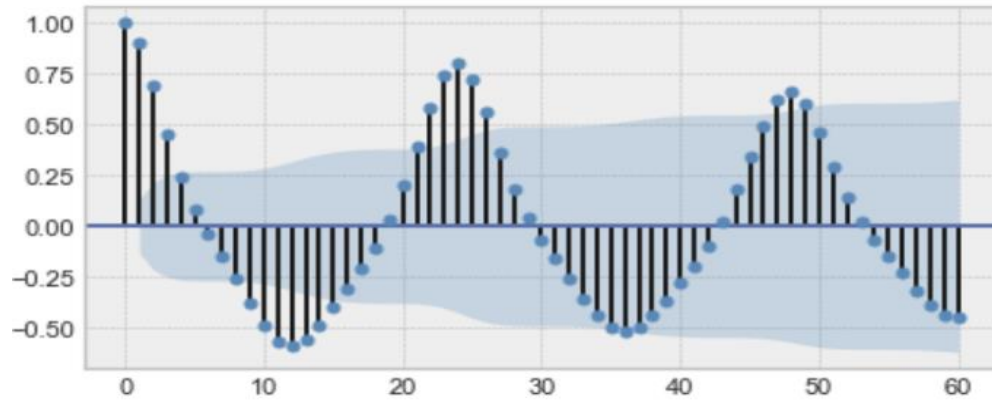


Yann LeCun – Facebook e Courant Institute



Time Series

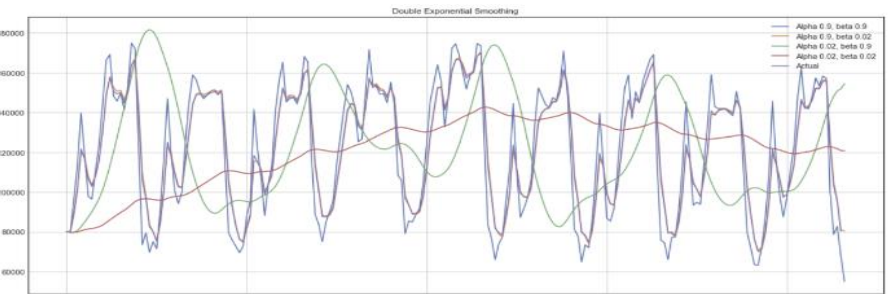
Autocorrelation



Médias móveis

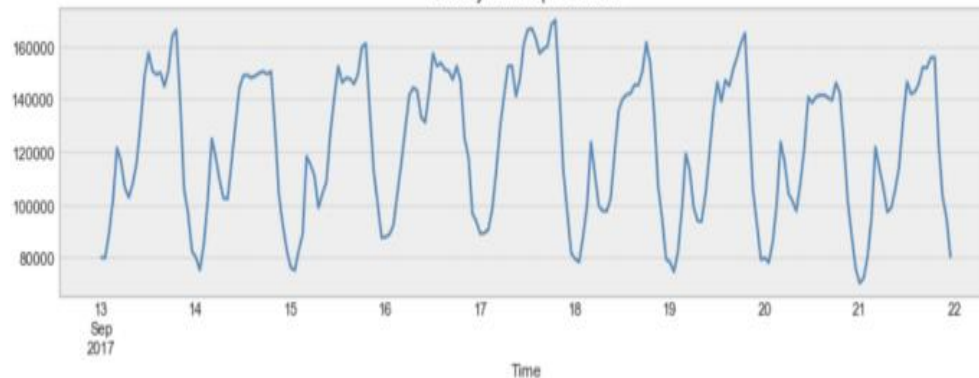


Exponential smoothing



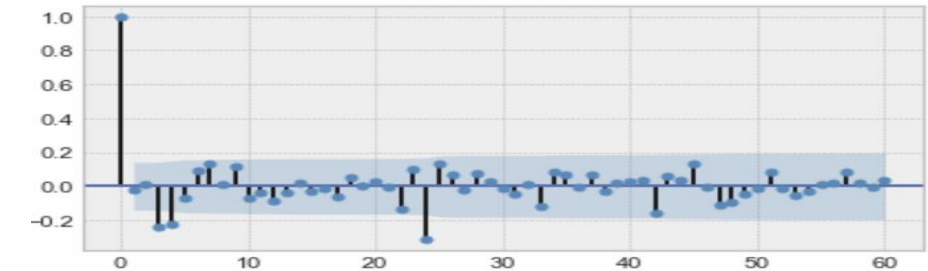
Seasonality and Stationarity (constant media and variance)

Time Series Analysis Plots
Dickey-Fuller: p=0.00000



ARIMA

Autocorrelation



Análise de Sentimento

Leitura sugerida:

<http://www.akitaonrails.com/2014/05/02/off-topic-carreira-em-programacao-codificar-nao-e-programar#.U2hY5PldUvz>

Várias técnicas e maneiras, também, várias bibliotecas

Bibliotecas para a lingua portuguesa:

- SentiLex-PT02
- propbankbr_v1_26112011
- oplexicon_v3.0
-

Análise de sentimento - Tradutor

```
import nltk.corpus
from textblob import TextBlob as tb

frase = tb("Este é um teste de textblob")
type(frase)
frase.tokens

# Lê o texto e quebra em sentenças
sent_tokenizer=nltk.data.load('tokenizers/punkt/portuguese.pickle')
rt = open("D:/Netbiis/Curso_AS/noticia1.txt", "r")
raw_text = rt.read()
sentences = sent_tokenizer.tokenize(raw_text)

# Mostra as sentenças em português
for sent in sentences:
    print(",", sent , ">>")

# Traduz as sentenças para o inglês
for sent_en in sentences:
    fi = tb(sent_en)
    se = fi.translate(from_lang = "pt", to = "en")
    print(",", se , ">>")
```

Análise de Sentimento – Análise das palavras de uma frase - SentiLex



Análise de Sentimento – Análise de frases com Naive Bayes e Random Forest



Perguntas ???

Obrigado!!!