Machine Learning Workshop

Nicolas Känzig

Email: nkaenzig@gmail.com
Workshop Repository: https://github.com/nkaenzig/ml-workshop

Contenido

Modulo 1

- Introducción ML
- Python

Modulo 2

- Análisis de datos
- Preprocesamiento de datos

Modulo 3

- Modelos de ML
- Técnicas de evaluación

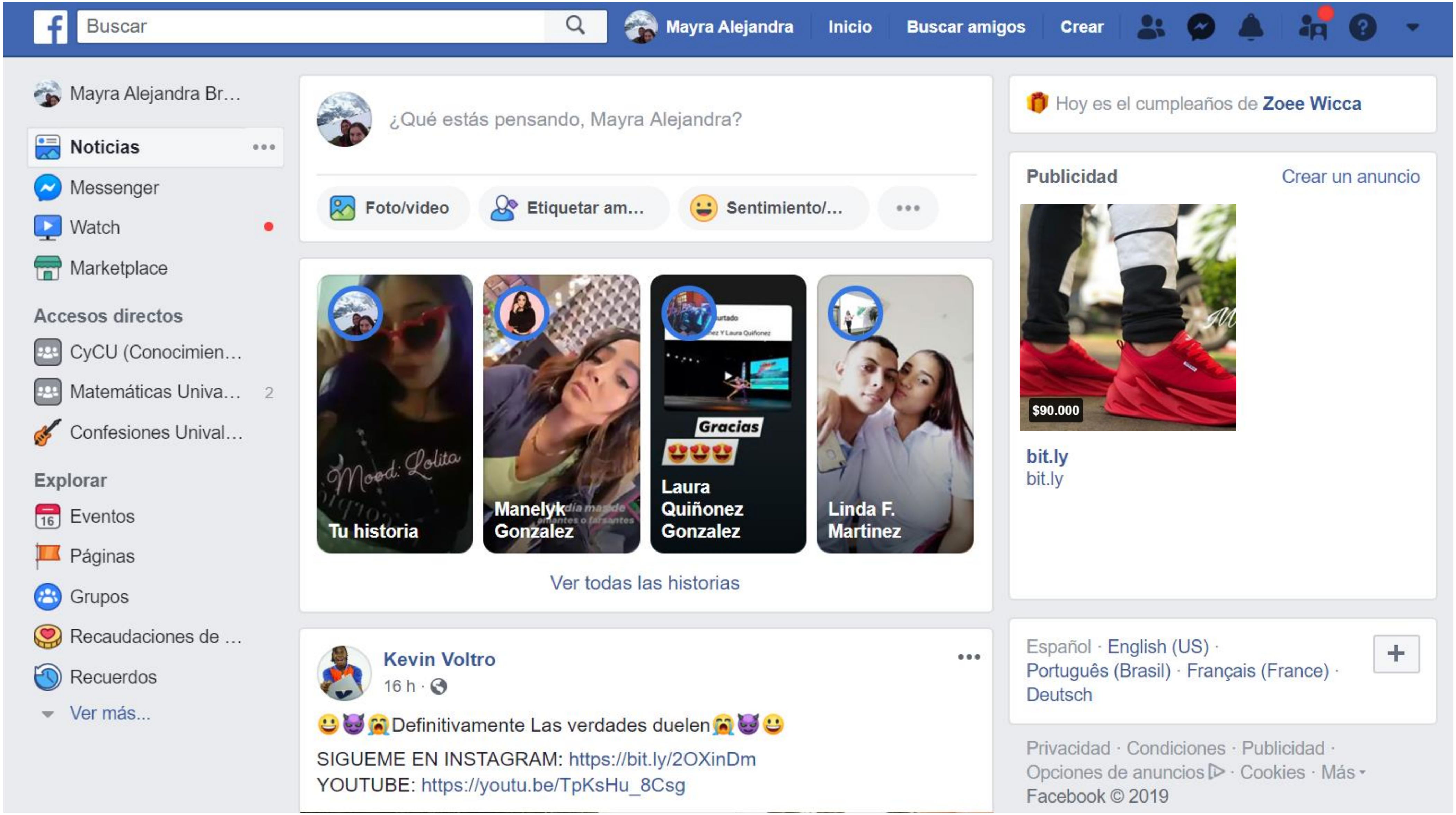
Machine Learning Introducción

Terminologías

- **Artificial Intelligence (AI)**
  - **Machine Learning**
    - Algoritmos que aprenden de datos
  - **Deep Learning**
    - Subconjunto de Machine Learning
    - Redes neuronales artificiales
  - **General AI**
    - Pensar, razonar, generalizar, curiosidad, …
    - El futuro

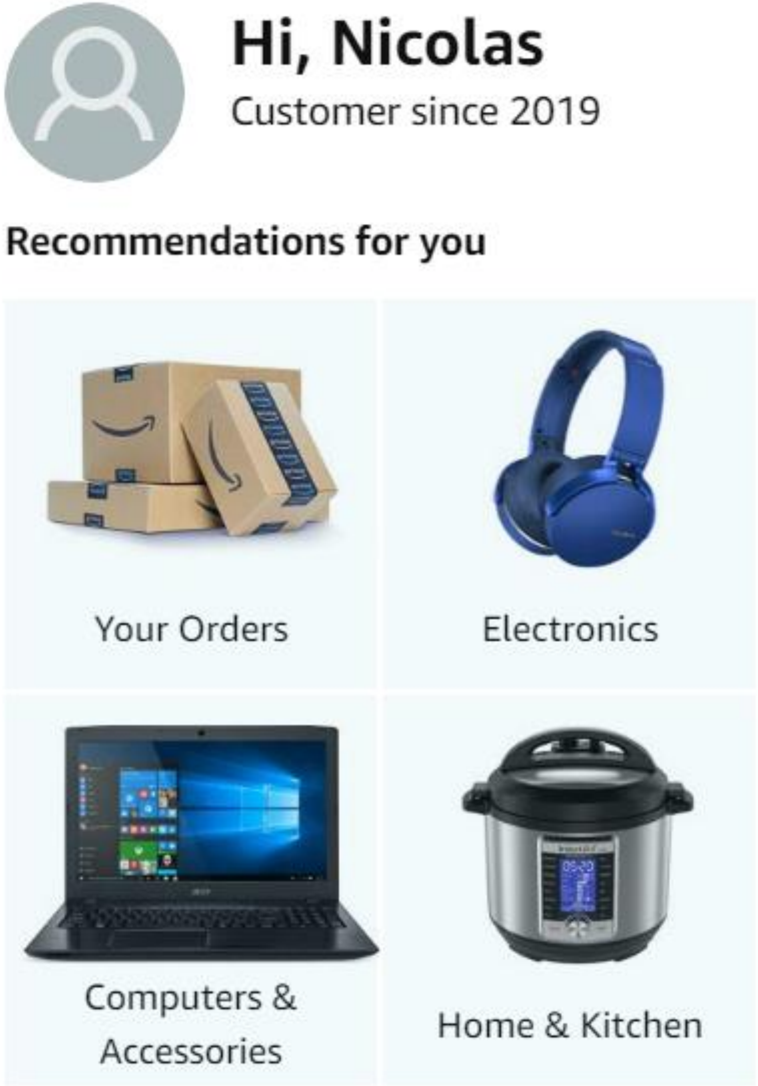
Terminologías

- **Artificial Intelligence (AI)**
  - **Machine Learning**
    - Algoritmos que aprenden de datos
  - **Deep Learning**
    - Subconjunto de Machine Learning
    - Redes neuronales artificiales
  - **General AI**
    - Pensar, razonar, generalizar, curiosidad, …
    - El futuro



😀😈😭Definitivamente Las verdades duelen😭😈😀

SIGUEME EN INSTAGRAM: https://bit.ly/2OXinDm
YOUTUBE: https://youtu.be/TpKsHu_8Csg



Hi, Nicolas

Customer since 2019

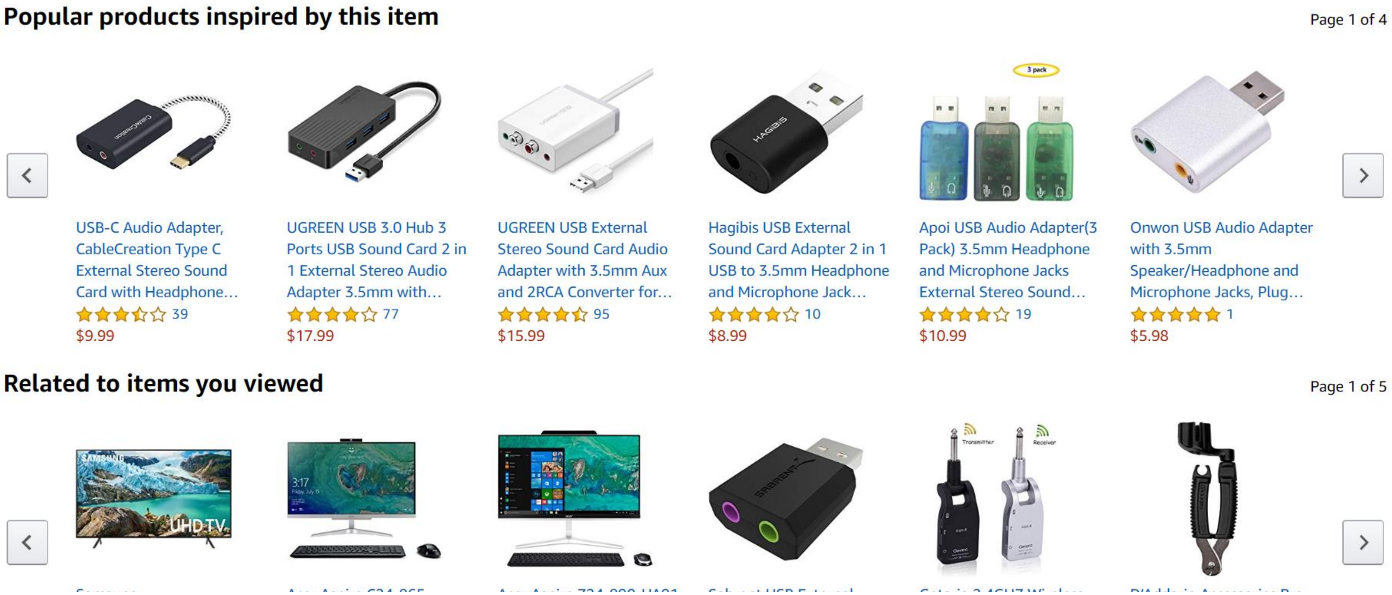
Recommendations for you

Your Orders

Electronics

Computers & Accessories

Home & Kitchen



Popular products inspired by this item Page 1 of 4

USB-C Audio Adapter, CableCreation Type C External Stereo Sound Card with Headphone… 39 $9.99

UGREEN USB 3.0 Hub 3 Ports USB Sound Card 2 in 1 External Stereo Audio Adapter 3.5mm with… 77 $17.99

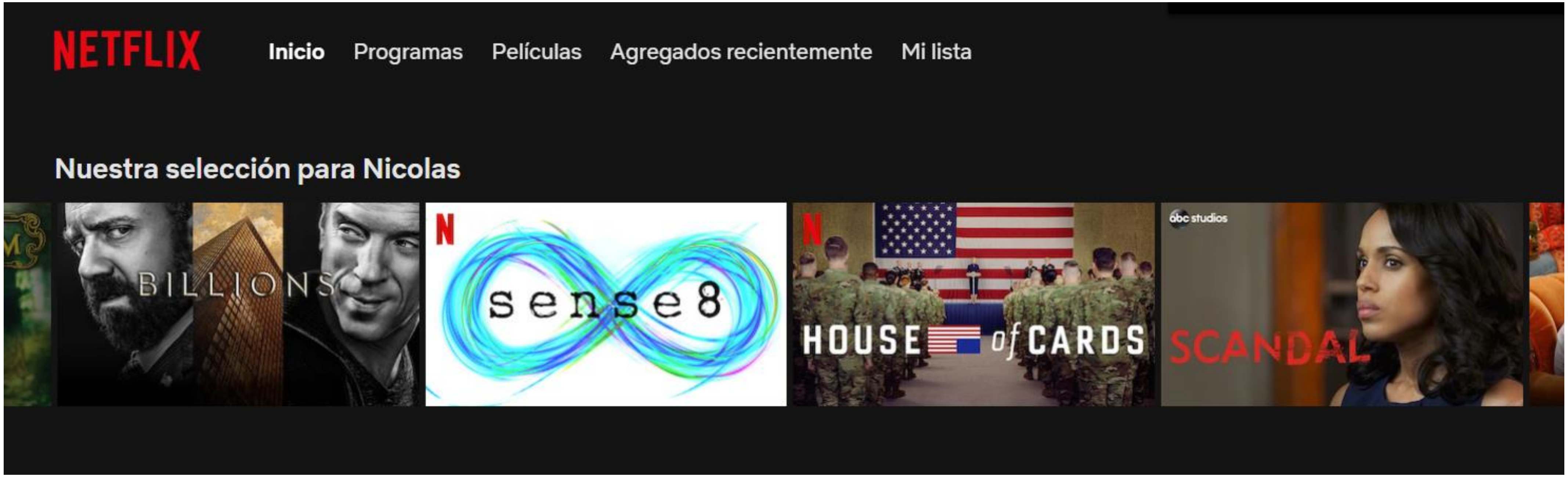
UGREEN USB External Stereo Sound Card Audio Adapter with 3.5mm Aux and 2RCA Converter for… 95 $15.99

Hagibis USB External Sound Card Adapter 2 in 1 USB to 3.5mm Headphone and Microphone Jack… 10 $8.99

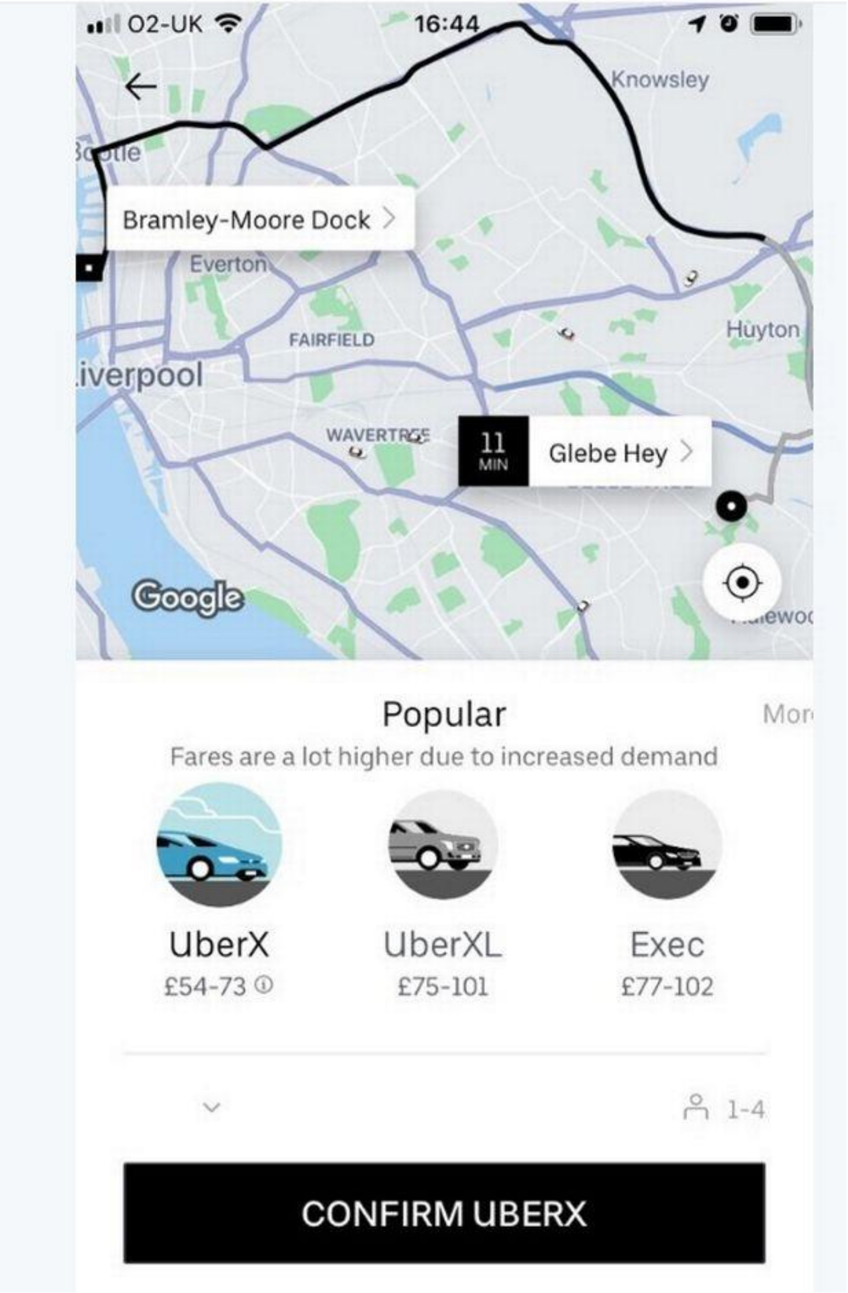
Apoi USB Audio Adapter(3 Pack) 3.5mm Headphone and Microphone Jacks External Stereo Sound… 19 $10.99

Onwon USB Audio Adapter with 3.5mm Speaker/Headphone and Microphone Jacks, Plug… 1 $5.98

Related to items you viewed Page 1 of 5


NETFLIX
Inicio
Programas
Películas
Agregados recientemente
Mi lista
Nuestra selección para Nicolas
BILLIONS
N
sense8
N
HOUSE of CARDS
abc studios
SCANDAL

Photo



Bramley-Moore Dock

11 MIN Glebe Hey

Popular

More

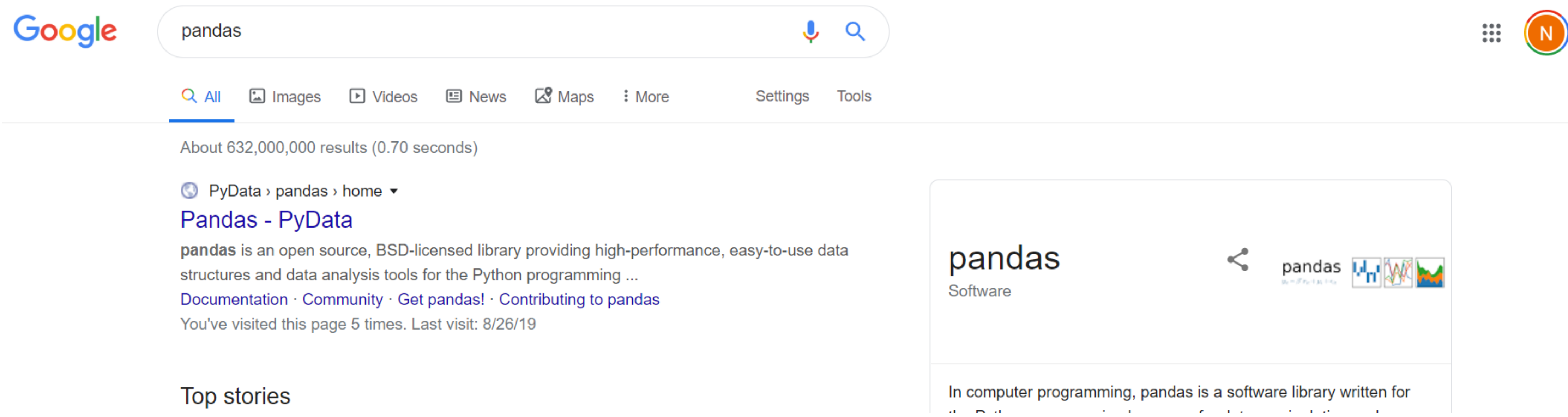
Fares are a lot higher due to increased demand

UberX	UberXL	Exec
£54-73	£75-101	£77-102

1-4

CONFIRM UBERX

Uber



About 632,000,000 results (0.70 seconds)

PyData › pandas › home

Pandas - PyData

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming ...

Documentation · Community · Get pandas! · Contributing to pandas

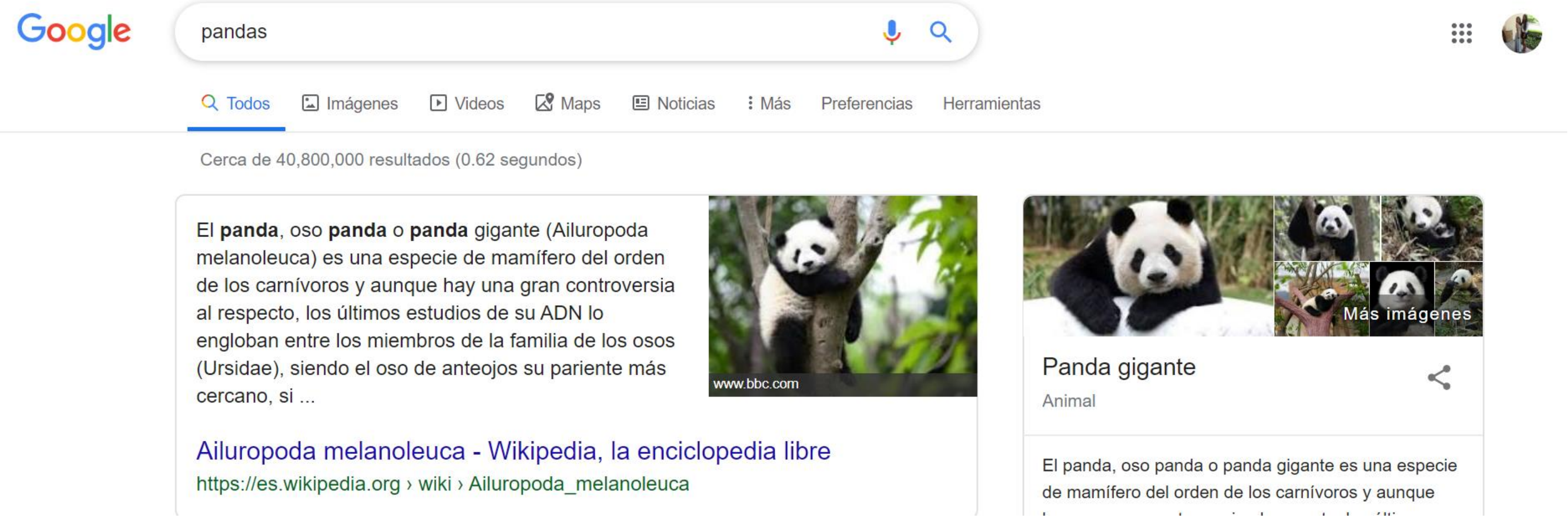
You've visited this page 5 times. Last visit: 8/26/19

Top stories

pandas

Software

In computer programming, pandas is a software library written for



Cerca de 40,800,000 resultados (0.62 segundos)

El **panda**, oso **panda** o **panda** gigante (Ailuropoda melanoleuca) es una especie de mamífero del orden de los carnívoros y aunque hay una gran controversia al respecto, los últimos estudios de su ADN lo engloban entre los miembros de la familia de los osos (Ursidae), siendo el oso de anteojos su pariente más cercano, si ...

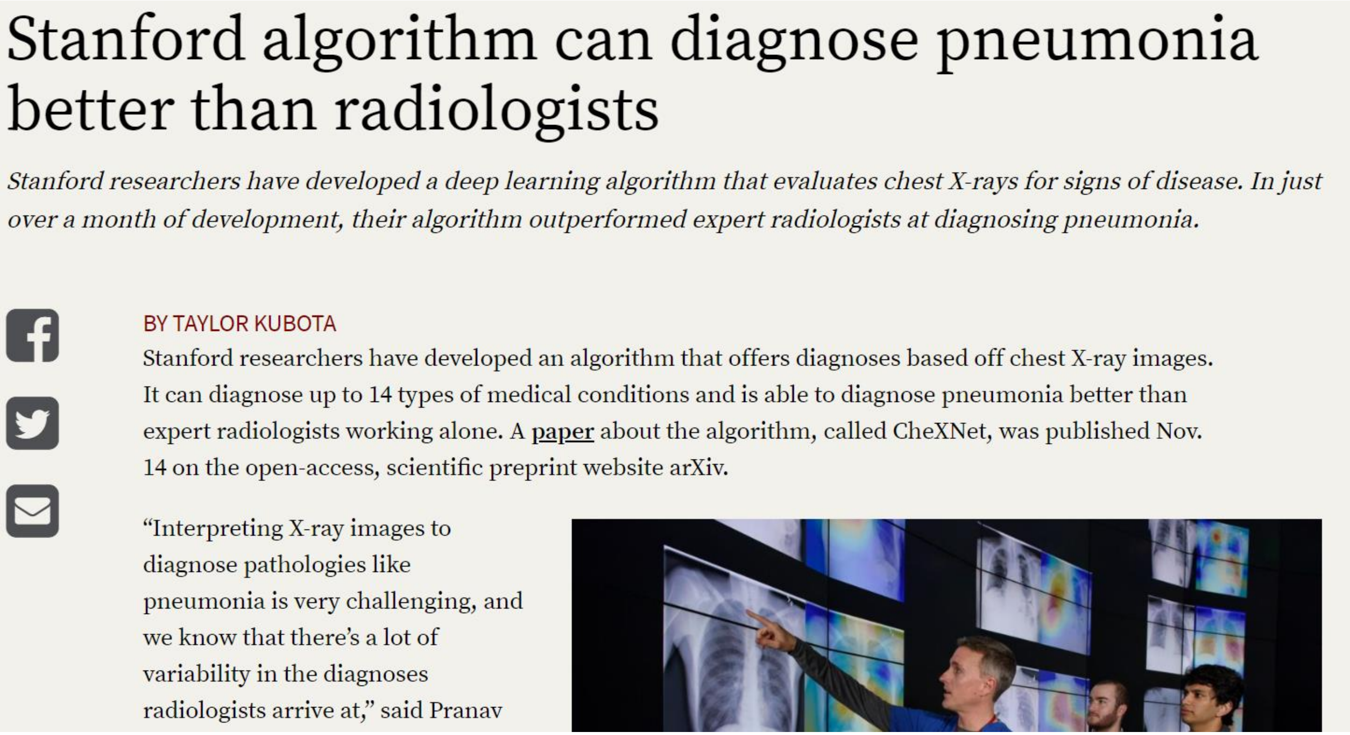
Ailuropoda melanoleuca - Wikipedia, la enciclopedia libre

https://es.wikipedia.org › wiki › Ailuropoda_melanoleuca

Panda gigante

Animal

El panda, oso panda o panda gigante es una especie de mamífero del orden de los carnívoros y aunque



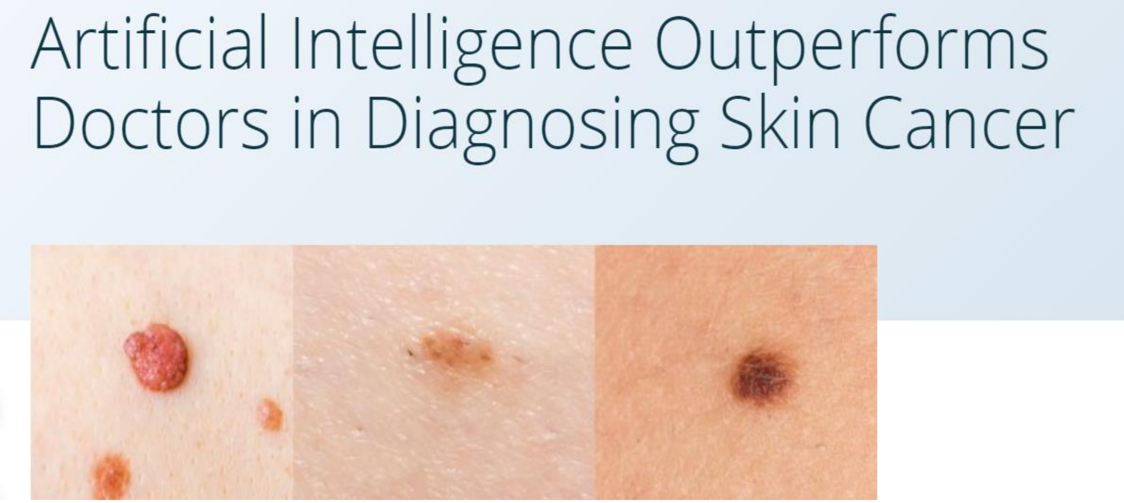
Stanford algorithm can diagnose pneumonia better than radiologists

Stanford researchers have developed a deep learning algorithm that evaluates chest X-rays for signs of disease. In just over a month of development, their algorithm outperformed expert radiologists at diagnosing pneumonia.

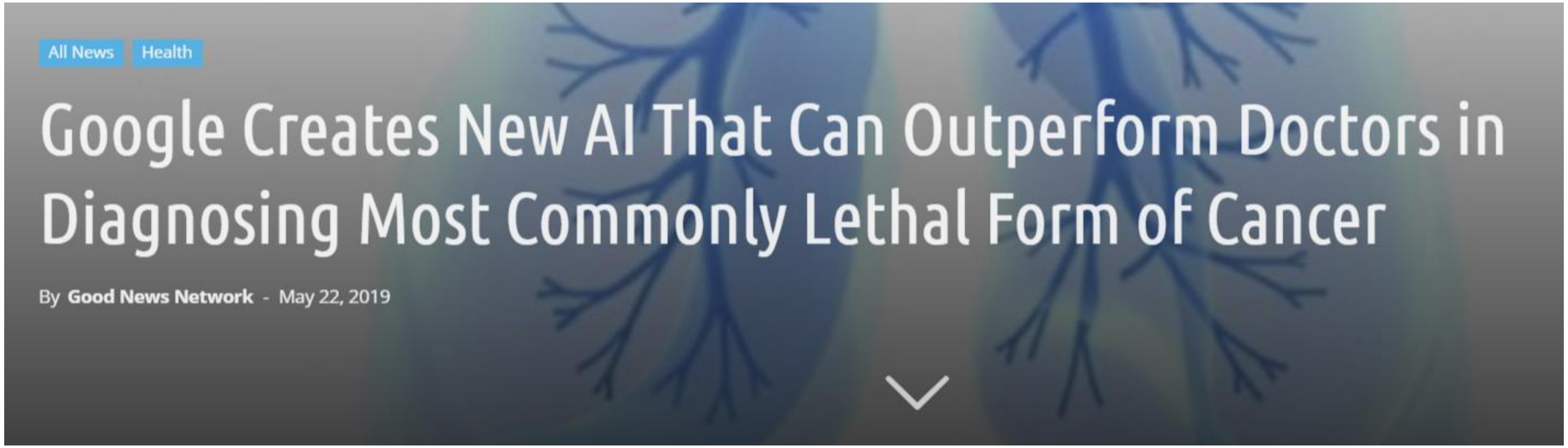
BY TAYLOR KUBOTA

Stanford researchers have developed an algorithm that offers diagnoses based off chest X-ray images. It can diagnose up to 14 types of medical conditions and is able to diagnose pneumonia better than expert radiologists working alone. A paper about the algorithm, called CheXNet, was published Nov. 14 on the open-access, scientific preprint website arXiv.

"Interpreting X-ray images to diagnose pathologies like pneumonia is very challenging, and we know that there's a lot of variability in the diagnoses radiologists arrive at," said Pranav



Artificial Intelligence Outperforms Doctors in Diagnosing Skin Cancer



Google Creates New AI That Can Outperform Doctors in Diagnosing Most Commonly Lethal Form of Cancer

By **Good News Network** - May 22, 2019

Que es Machine Learning?

$$f(x, \theta)$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} L(x, y, \theta)$$

e.g. $L(x, y, \theta) = |f(x, \theta) - y|^2$

- Machine Learning
  - Optimización
  - Matemática numérica
  - Estadística
  - Algebra lineal
  - Computer Science
  - …

Frameworks

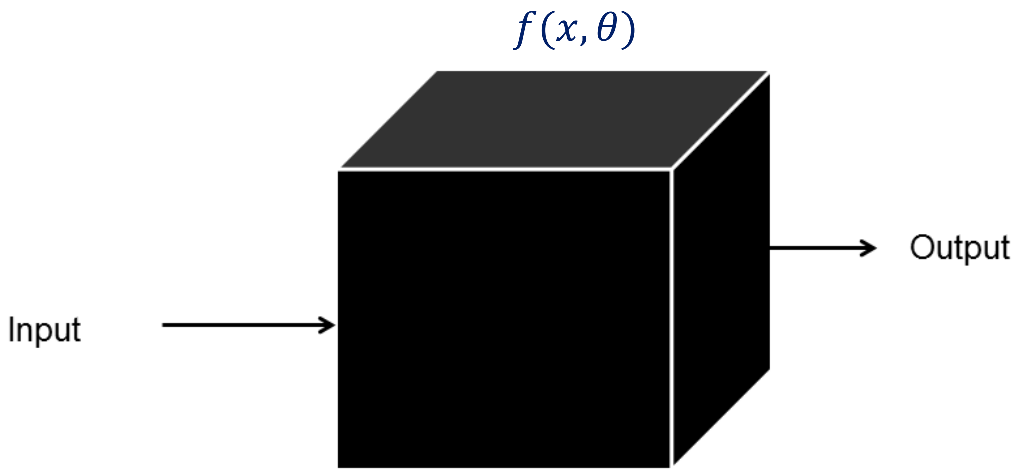
- Implementación de modelos
- Algoritmos de optimización
- Computación rápida y paralela
  - Python: API
  - C/C++: Algoritmos

Traditional Programming vs. Machine Learning

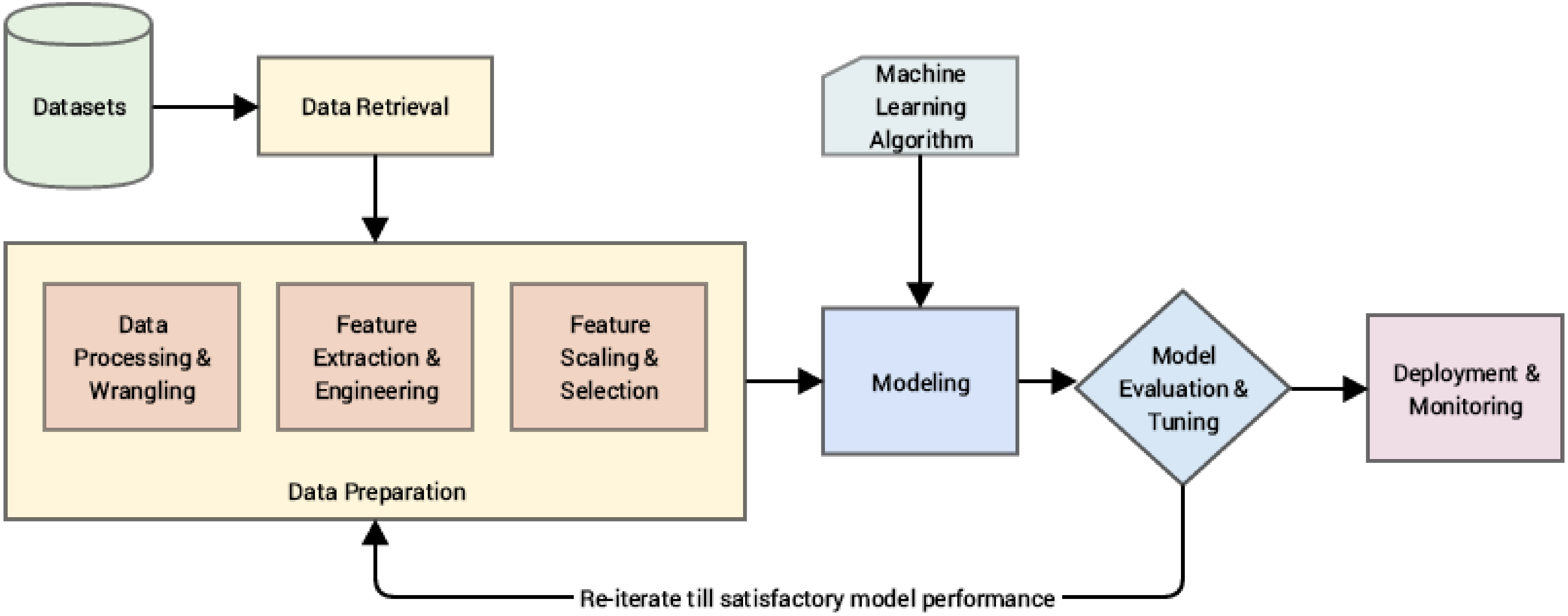
Traditional Programming / Rule-Based

```
if condition1:
    # Do something
elif condition2:
    # Do something else
else:
    # Default action
```

Machine Learning



Workflow



Datasets

Data Retrieval

Machine Learning Algorithm

Data Processing & Wrangling

Feature Extraction & Engineering

Feature Scaling & Selection

Data Preparation

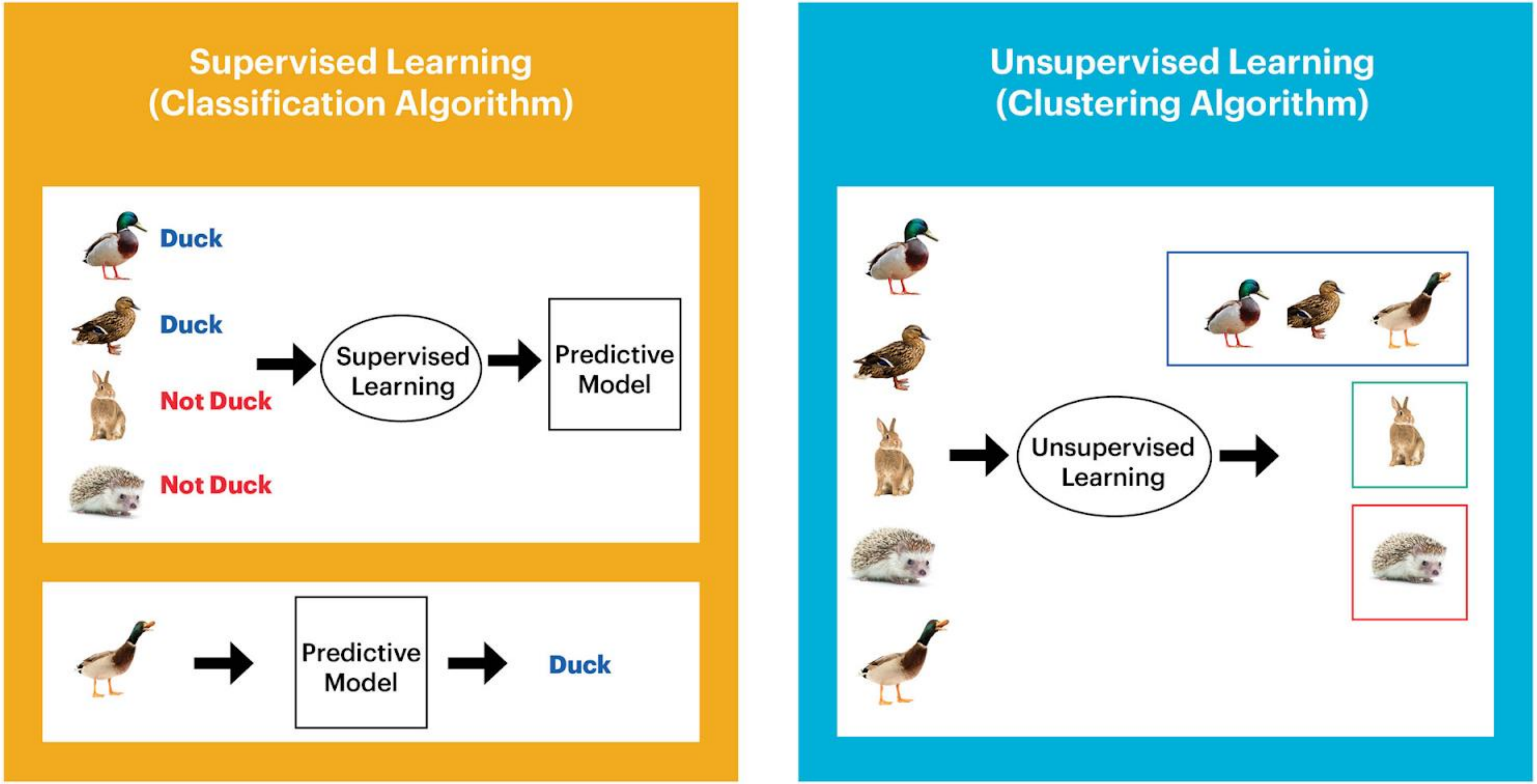
Modeling

Model Evaluation & Tuning

Deployment & Monitoring

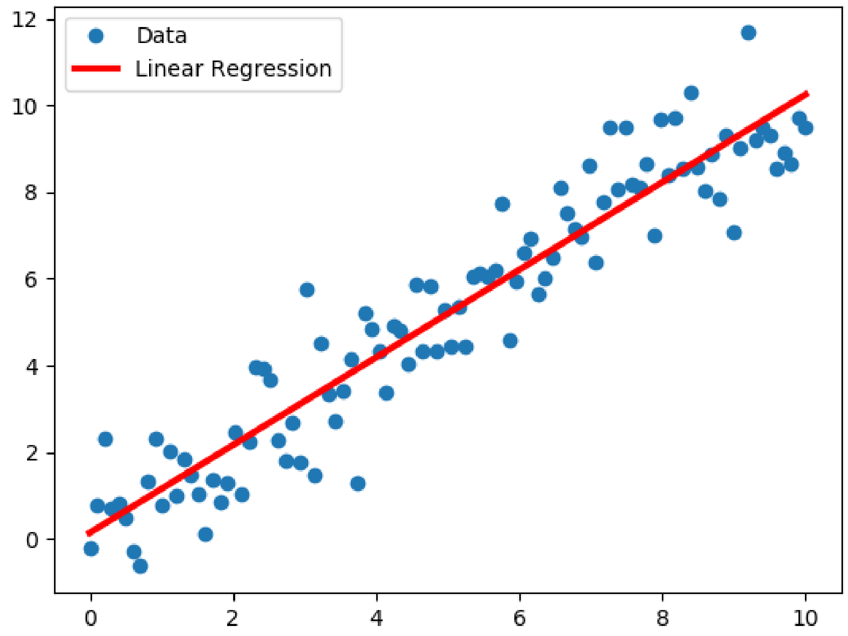
Re-iterate till satisfactory model performance

Supervised vs. Unsupervised Learning



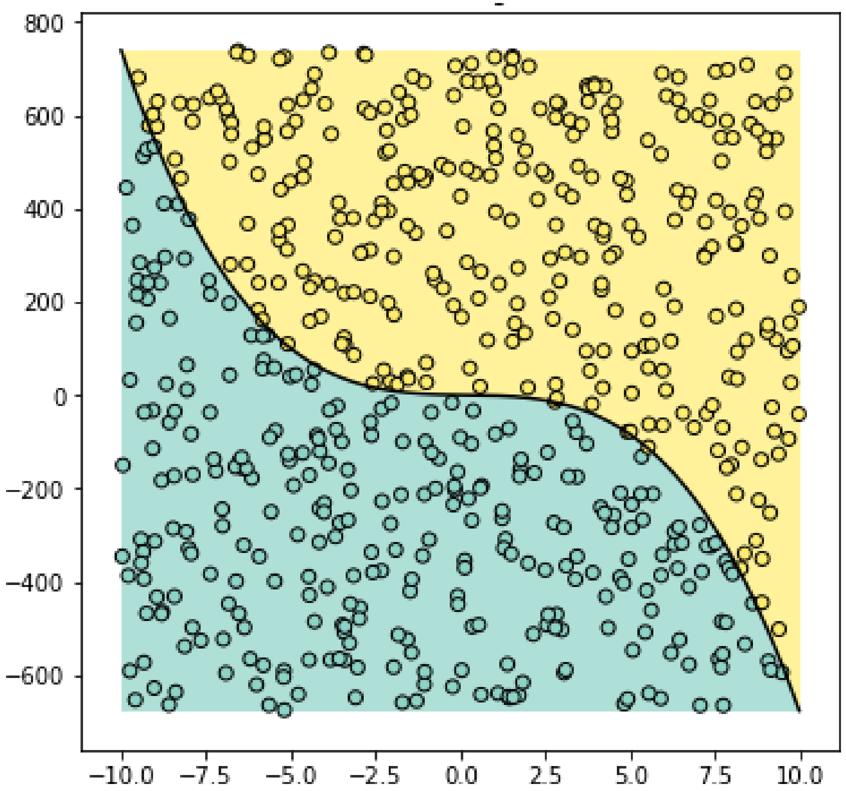
Supervised Learning

Regression

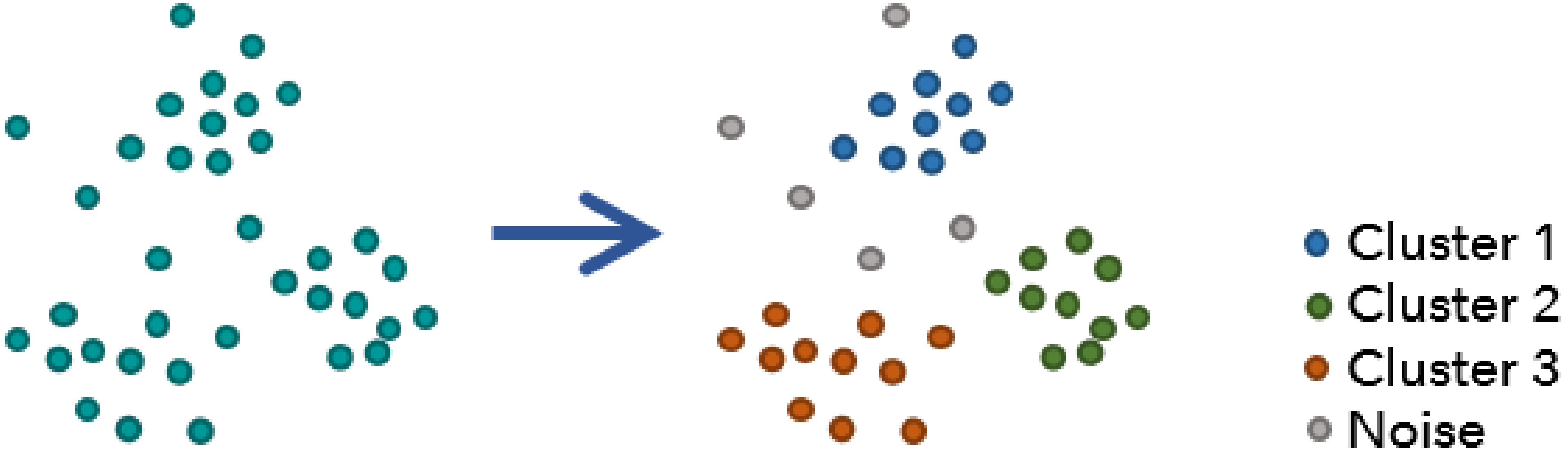


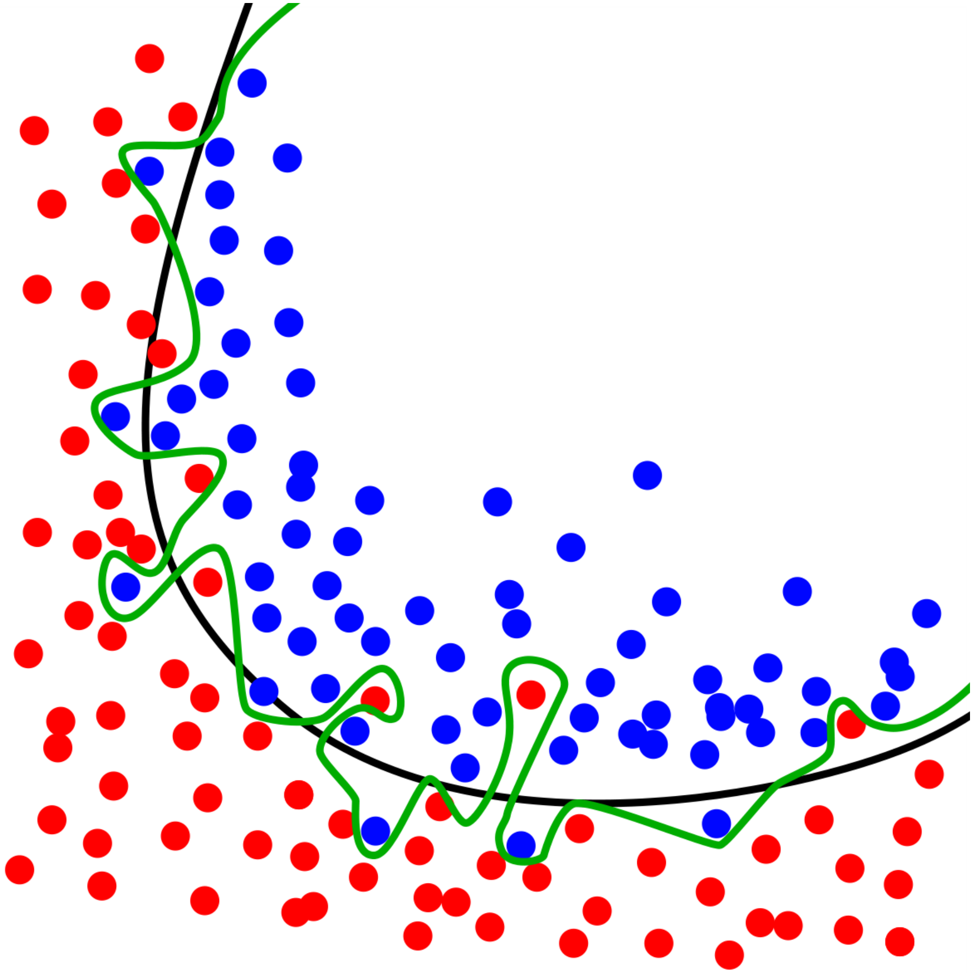
$$f(x,\theta) = \theta_0 + \theta_1 x$$

Classification

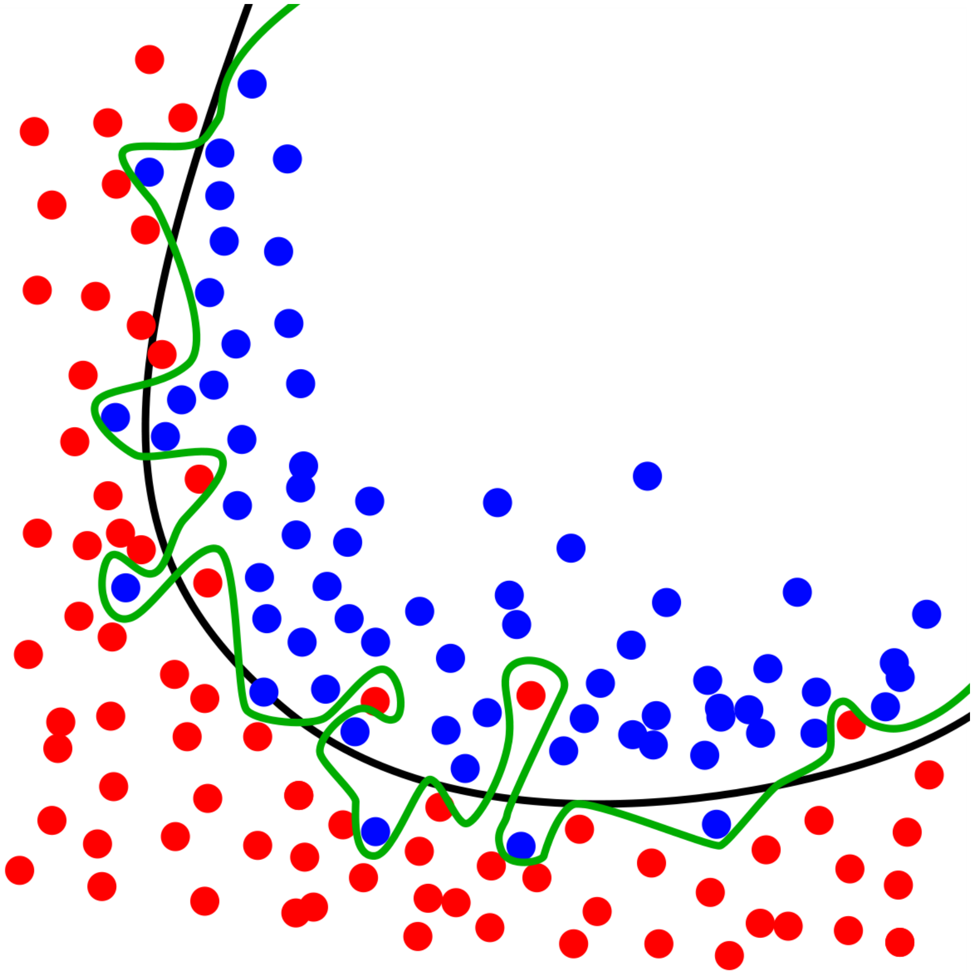


Unsupervised Learning (Clustering)

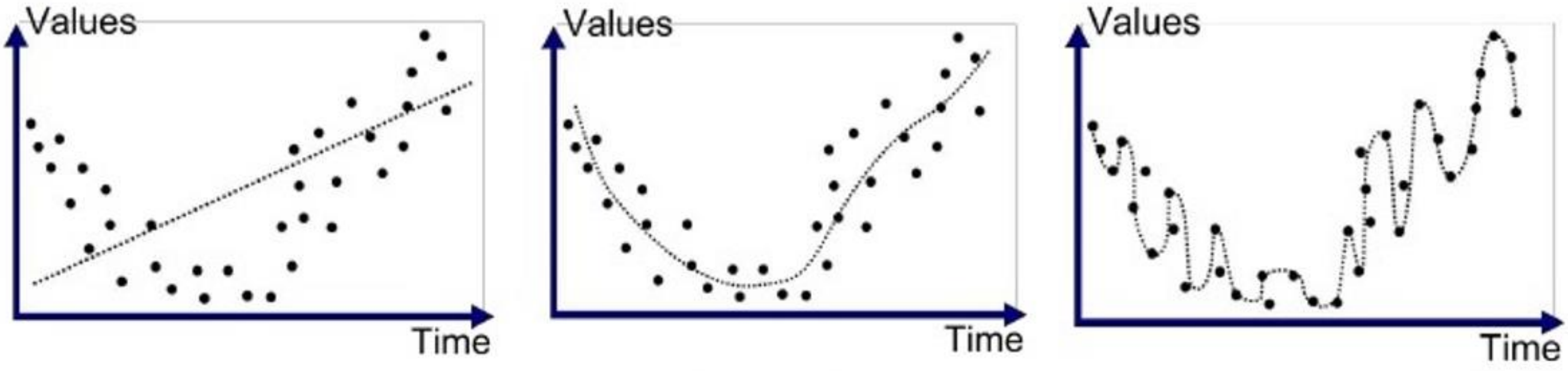




Overfitting



Underfitting vs. Overfitting



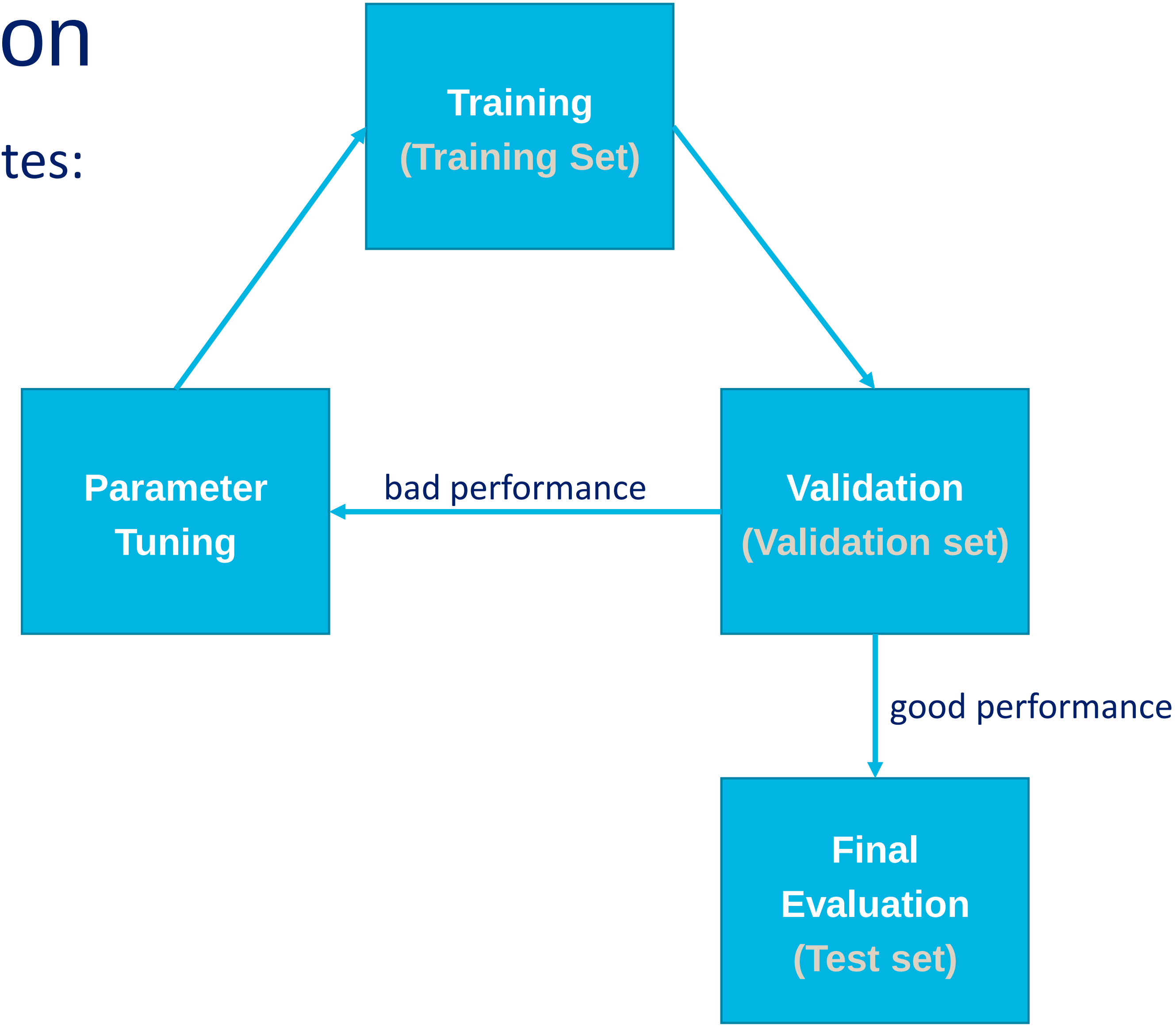
Underfitted

Good Fit/Robust

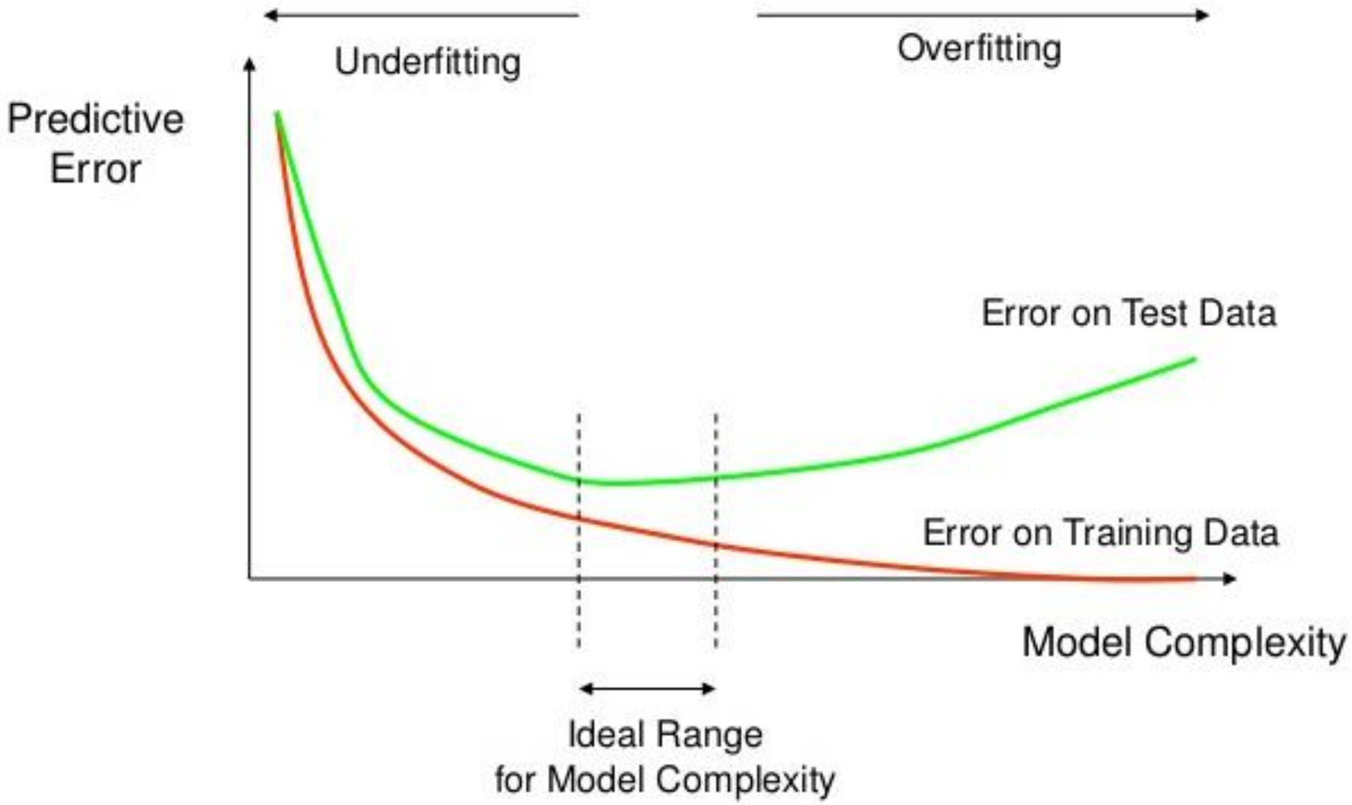
Overfitted

Training vs. Validation

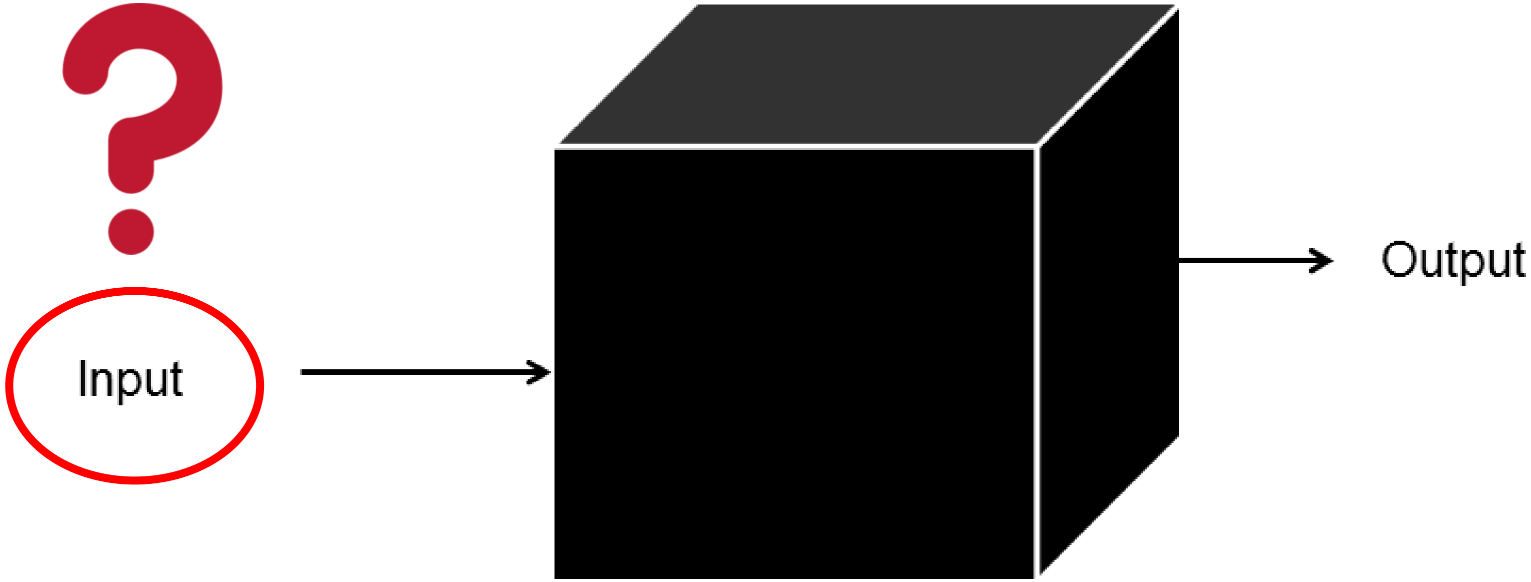
- División de los datos en 3 partes:
    - Training set (70%)
    - Validation set (20%)
    - Test set (10%)



Overfitting vs. Model Complexity

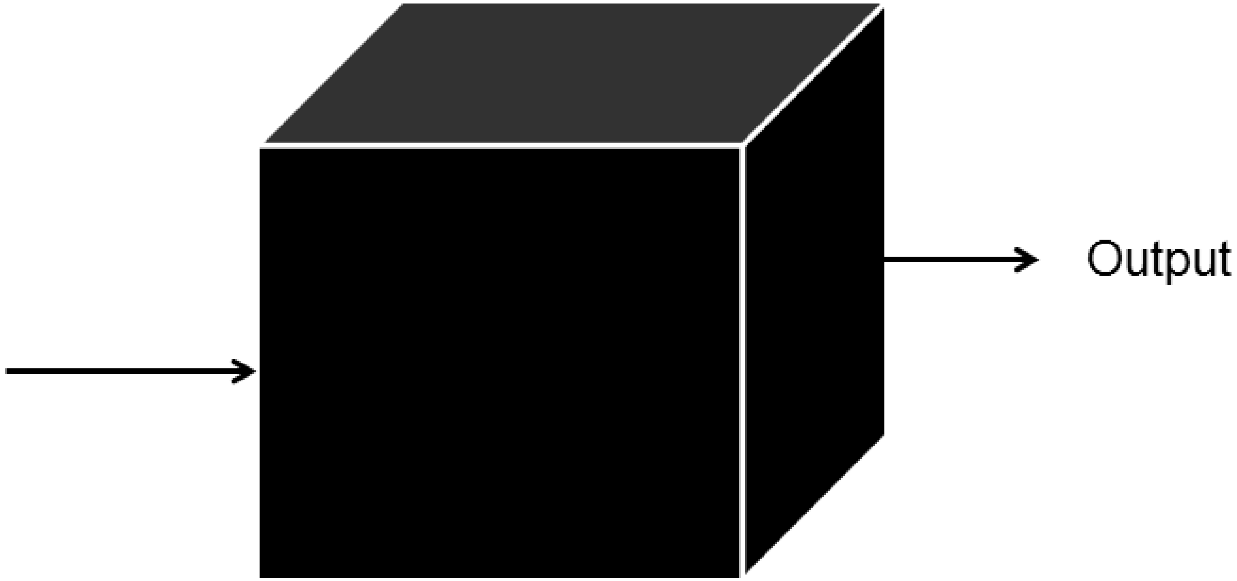


Que es el Input / Formato de los datos?



Que es el Input?

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$



$a_{ij} \in \mathbb{R}$

Datasets

- Tabulas (Excel, CSV, SQL, ...)
- Textos
- Imágenes (Deep Learning)
- Audio (Deep Learning)
- ...

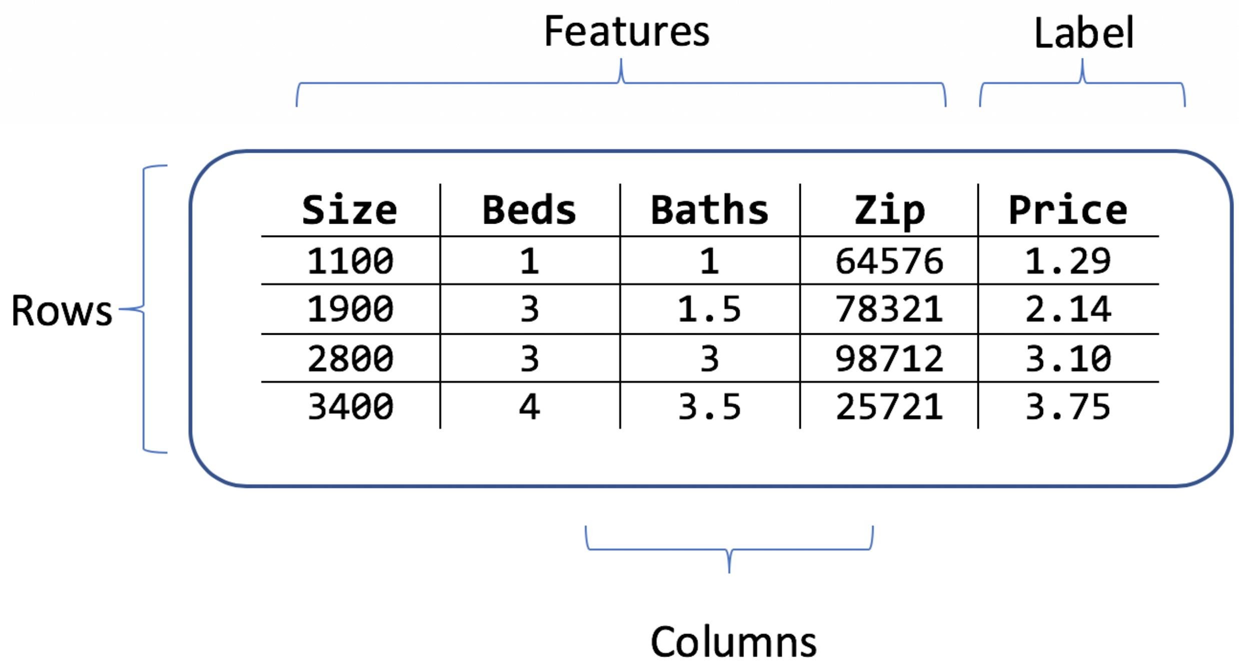
$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

$$a_{ij} \in \mathbb{R}$$

Features vs. Labels

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

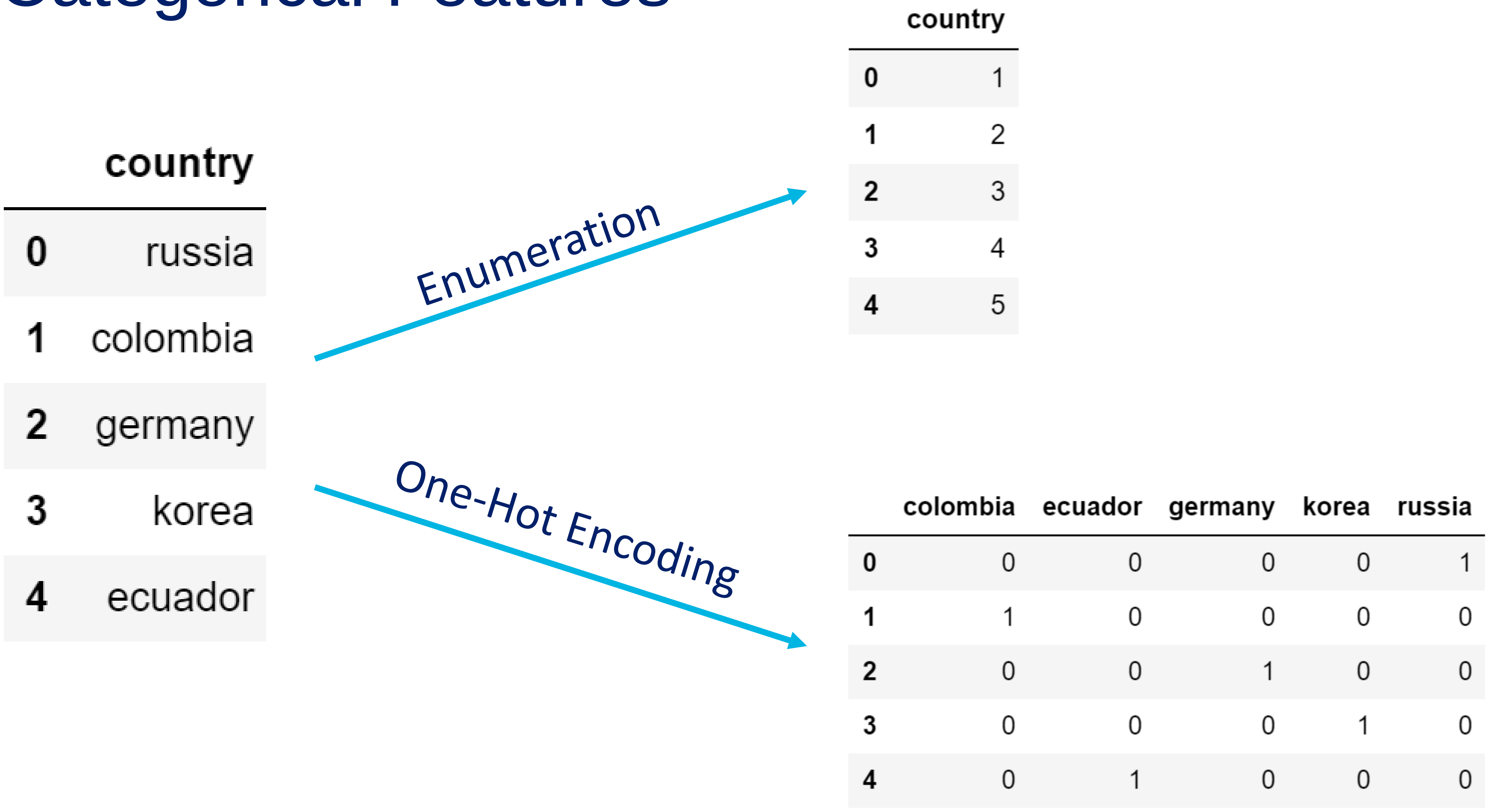
m x n
n: # features
m: # samples


Features
Label
Rows
Columns

Size	Beds	Baths	Zip	Price
1100	1	1	64576	1.29
1900	3	1.5	78321	2.14
2800	3	3	98712	3.10
3400	4	3.5	25721	3.75

Columns = Features/Label
Rows = Samples

Categorical Features



	country
0	russia
1	colombia
2	germany
3	korea
4	ecuador

Enumeration

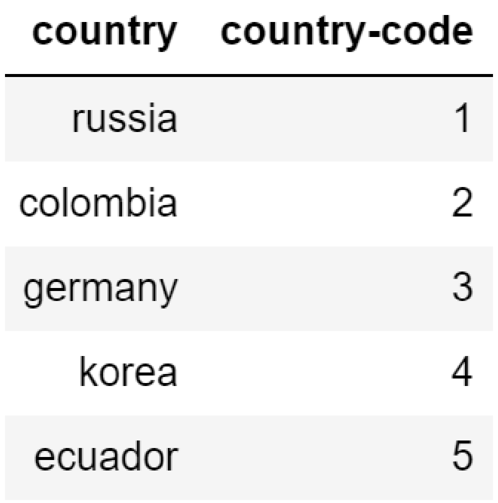
	country
0	1
1	2
2	3
3	4
4	5

One-Hot Encoding

	colombia	ecuador	germany	korea	russia
0	0	0	0	0	1
1	1	0	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0
4	0	1	0	0	0

Desventajas

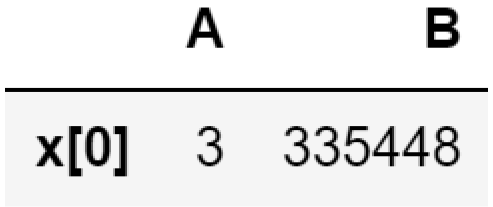
- **Enumeration**
  - Distancia euclidiana da falsa información
    - $|Russia\ - Colombia| = |1 - 2| = 1$
    - $|Colombia\ - Ecuador| = |2 - 5| = 3$



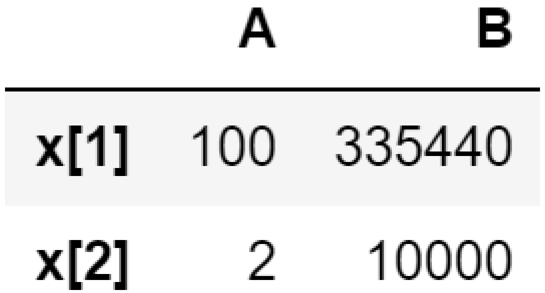
country	country-code
russia	1
colombia	2
germany	3
korea	4
ecuador	5

- **One-Hot Encoding**
  - "The curse of dimensionality" ("La maldición de la dimensionalidad")
    - 10’000 categorías → 10’000 columnas nuevas
    - Sparsity: Casi todos los valores de la matriz son 0
    - Uso de memoria

Similaridad ?

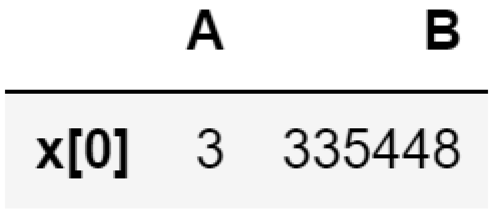


	A	B
x[0]	3	335448

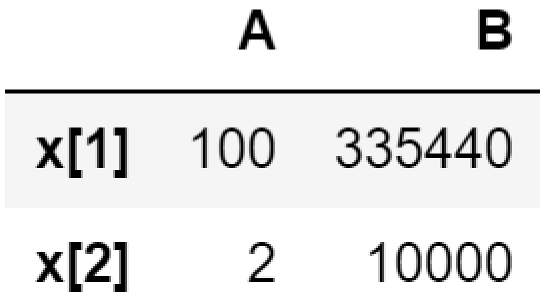


	A	B
x[1]	100	335440
x[2]	2	10000

Similaridad ?



	A	B
x[0]	3	335448



	A	B
x[1]	100	335440
x[2]	2	10000

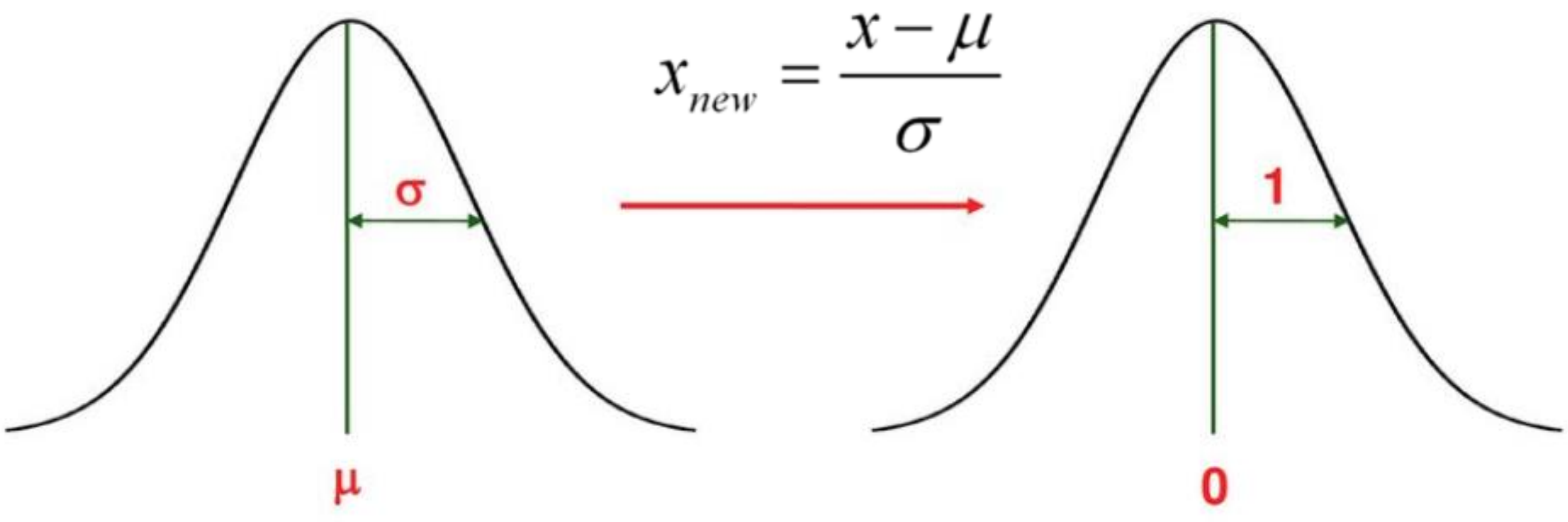
Distancia Euclidiana

$$dist(x[0], x[1]) = 97.33$$

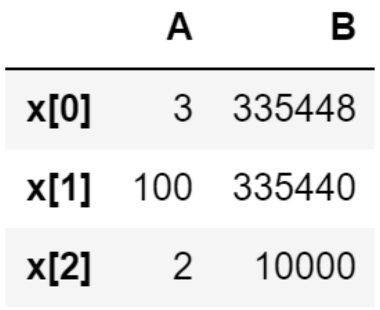
$$dist(x[0], x[2]) = 325448$$

Y si A son [km] y B son [mm] ?

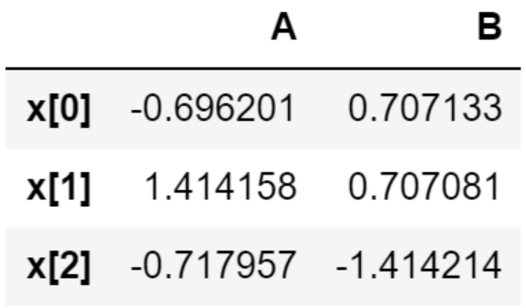
Standardización



$$x_{new} = \frac{x - \mu}{\sigma}$$



	A	B
x[0]	3	335448
x[1]	100	335440
x[2]	2	10000



	A	B
x[0]	-0.696201	0.707133
x[1]	1.414158	0.707081
x[2]	-0.717957	-1.414214

[-1, 1]

Dificultades

- Entender los datos
- Datos en un formato adecuado para entrenar modelos (Matriz)
- Definir la tarea (Regression, Classification, …)
- Conseguir suficiente datos
- Conseguir Labels
- Seleccionar un modelo y encontrar los mejores parámetros
- Prevenir Overfitting
- Computación / Memoria

Python

Porque Python?

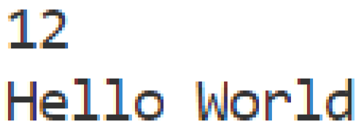
- La lenguaje mas usada en la industria (ML & DS)
- Python es muy sencillo de enseñar y de usar
- Muchas Liberarías / Frameworks para ML
- Instalación es muy fácil

Code Example 1

```
# A comment.
x = 34 - 23
y = "Hello" # Another comment.
z = 3.45
if z < 5 or y == "Hello" and not z > x:
    x += 1
    y = y + " World"
print(x)
print(y)
```

Code Example 1

```
# A comment.
x = 34 - 23
y = "Hello" # Another comment.
z = 3.45
if z < 5 or y == "Hello" and not z > x:
    x += 1
    y = y + " World"
print(x)
print(y)
```



```
12
Hello World
```

- No hay declaración de tipo de datos
- Asignación de variables con “=”
  - Primera asignación crea variable
- Comentarios: #
- Operadores lógicos son palabras:
  - `and, or, not`
- Special use of + for string concatenation
- Printing command: `print()`
- Scope declaration with indentations (no {})

Naming Conventions

- Case sensitive

```
Name = "Alejandra"
name = "Jorge"
```

  - Upper case no es muy común

- Snake case for variables

```
a_variable_with_a_long_name = 22
```

- CamelCase for class names

```
class MyClassName
```

- Reserved words

```
and, or, not, assert, break, class, continue, def, del, elif, else, except,
exec, finally, for, from, global, if, import, in, is, lambda, pass, print,
raise, return, try, while
```

Basic Datatypes

- Integers

```
x = 1
y = 5 / 2 # result is 2.5 for Python3, 2 for Python2
```

- Floats

```
x = 3.256
```

- Strings

```
x = "Machine Learning"
```

- Boolean

```
x = True
y = False
```

Conditional Branching

```python
if condition_a:
    # do something
elif condition_b:
    # do something else
else:
    # default action
```

Loops

```python
# For-Loop
for i in range(10):
    print(i)

# While-loop
i=0
while i < 10:
    print(i)
    i += 1
```

Complex Datatypes

- Lists

```
x = [2, "ML", 2, 3.75, [1, "a"]]
```

- Tuples

```
x = (2, "ML", 2, 3.75, [1, "a"]) # immutable
```

- Dictionaries

```
x = {"name": "Alejandra", "age": 21}
```

- Sets

```
x = {"Alejandra", "Jorge", "Maria"} # not ordered
```

Lists

```
x = [2, "ML", 3.75]

# Add element to List
x.append(5) # [2, "ML", 3.75, 5]


# List concatenation
y = [2, 1]
z = x + y # [2, "ML", 3.75, 5, 2, 1]
```

Lists

```python
x = [2, "ML", 3.75, 5]

# Indexing
x[0] # 2
x[-1] # 2
x[1:] # ["ML", 3.75, 5]
x[:2] # [2, "ML"]
x[1:3] # ["ML", 3.75]

# Check if contains element
if "ML" in x:
    # do something
```

Lists

```
x = [1.2, 200.53, 55, 2.44, 77]

# Loop through elements ("foreach")
for value in x:
    # do something

# List comprehension
a = [round(value) for value in x] # [1, 200, 55, 2, 77]
b = [value for value in x if value > 50] # [200.53, 55, 77]
c = [value if value > 50 else -1 for value in x] # [-1, 200.53, 55, -1, 77]
```

Tuples

- Same as List, but immutable

```
x = (2, "ML", 2, 3.75, [1, "a"])

>>> x[2] = "test"
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: "tuple" object does not support item
assignment
```

Dictionaries

```python
x = {"name": "Alejandra", "age": 21}
x["age"] = 5 # overrides current value assigned to key "age"
del x["name"] # deletes the key “name“ and its value
keys = x.keys()
values = x.values()

# iterate over keys
for key in x:
    # do something

# iterate over keys & values
for key, value in x.items():
    # do something
```

Sets

```
x = {"A", "B", "C"}
x.add("D") # adds D to set
x.add("D") # won"t change set, as D already exists
x.update(["E", "F", "G"]) # adds multiple elements to set

x.remove("A")
x.remove("Z") # raises error
x.dicard("Z") # no error

a = set([1, 2, 3])
b = set([2, 3, 4])
intersection = a.intersection(b) # or a&b
union = a.union(b) # or a|b
difference = a.difference(b) # or a-b
```

$A \cap B$

$A \cup B$

$A \backslash \mathrm{B}$

Iterators

```
>>> itr = iter([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> next(itr)
0
>>> next(itr)
1
```

```
>>> range(10)
range(0, 10)
```

```
for i in range(10):
    print(i)
```

```
>>> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Functions

```
def calculate_sum(a,b):
    return a+b

f = lambda x: x*2
f(4) # 8

a = ["bla_4", "bla_2", "bla_8"]
sorted(a, key=lambda x: x[-1])
```

Classes

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

p = Person("John", 36)

print(p.name)
print(p.age)
```

Type Conversion

```
x = 2.54
int(x) # 2
float(2.0) # 2
str(x) # "2"

x_list = [2, 2, 2, 55, 12, 3]
x_set = set(x_list)
y_list = list(x_set)

indices = list(range(20))
```

File I/O

```
# Read file line-by-line
with open(filepath, "r") as fp:
  for line in fp:
    print(line)

# Write line to file
with open(filepath, "w") as fp:
  fp.write("test\n")
```

Exception Handling

```
try:
  x = 1/0
except ZeroDivisionError:
  print("Division by zero exception")
except:
  print("Any other exception")
```

String Methods

```
age = 21
print(f"Alejandra is {age} years old") # Python >= 3.6
print("Alejandra is {} years old".format(age))

price = 100000.2356412
print(f"The house costs {price:.{2}f} USD") # Python >= 3.6
print("This house costs {:.2f} USD".format(price))

csv = "12;Test;987.11"
csv_splitted = csv.split(";") # ["12", "Test", "987.11"]
csv_joined = splitted.join(";") # csv == csv_joined

str_with_spaces = " test string "
str_striped = str_with_spaces.strip() # "test string“
```

https://docs.python.org/3/library/stdtypes.html?highlight=upper#string-methods

Libraries

- Install/Uninstall modules:

```
pip install pandas
pip install pandas==0.21.0
pip uninstall pandas
```

```python
import pandas as pd
import numpy as np

# load mylibrary.py
import sys
sys.path.insert(1, "/path/to/application/app/library_folder")
import mylibrary
```