Machine Learning Workshop

Mentor: Nicolas Känzig

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

Contenido

Modulo 1

- Introducción ML
- Python crashcourse

Modulo 2

- Análisis de datos
- Preprocesamiento de datos
- Ejemplo ML

Modulo 3

- Modelos de ML
- Técnicas de evaluación
- Ejemplos ML

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

Aplicaciones de Machine Learning

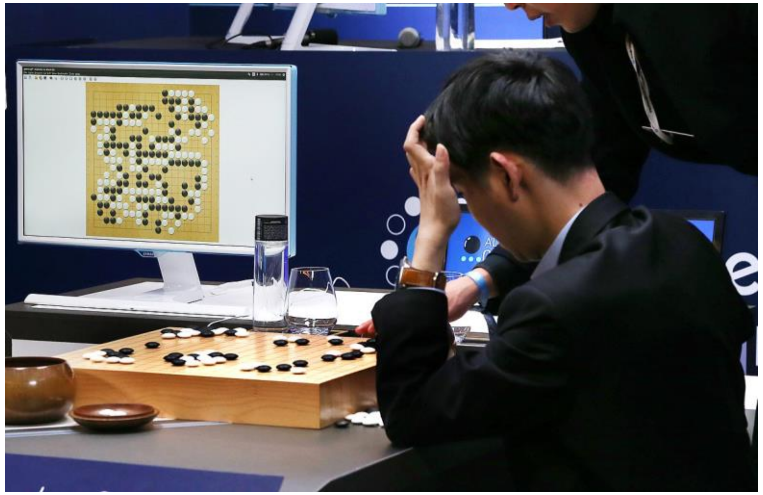
- Predicciones de ventas, precios, …
- Clasificación de textos
- Sistemas de recomendaciones
- Medicina
- Detección de fraudes
- …

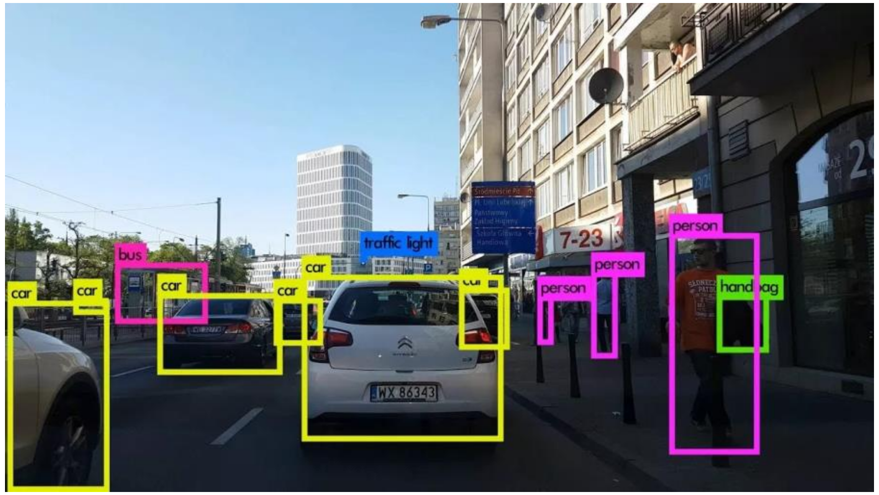
Aplicaciones de Deep Learning

- Clasificación de imágenes
- Reconocimiento de objetos
- Traducción de idiomas
- Reconocimiento de voz
- AlphaGo
- …

Google Translate

TESLA



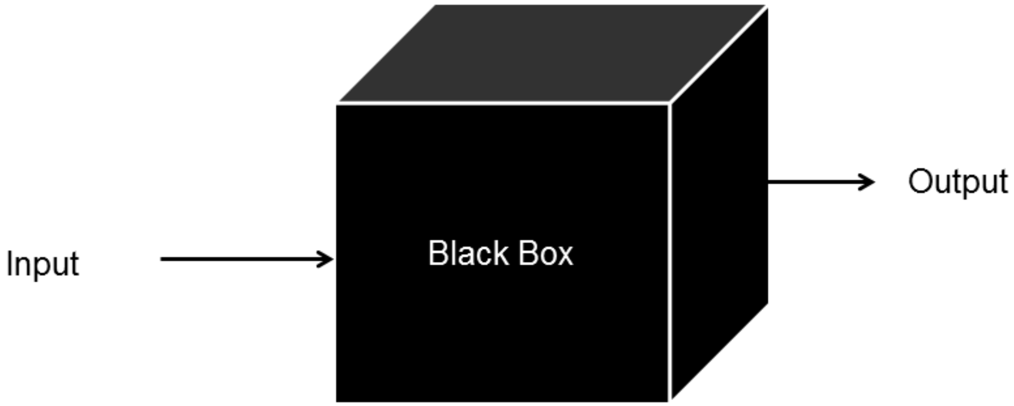


Rule-Based Systems v.s. Machine Learning

Rule-Based

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

Machine Learning



Que es Machine Learning?

$$f(x, \theta)$$

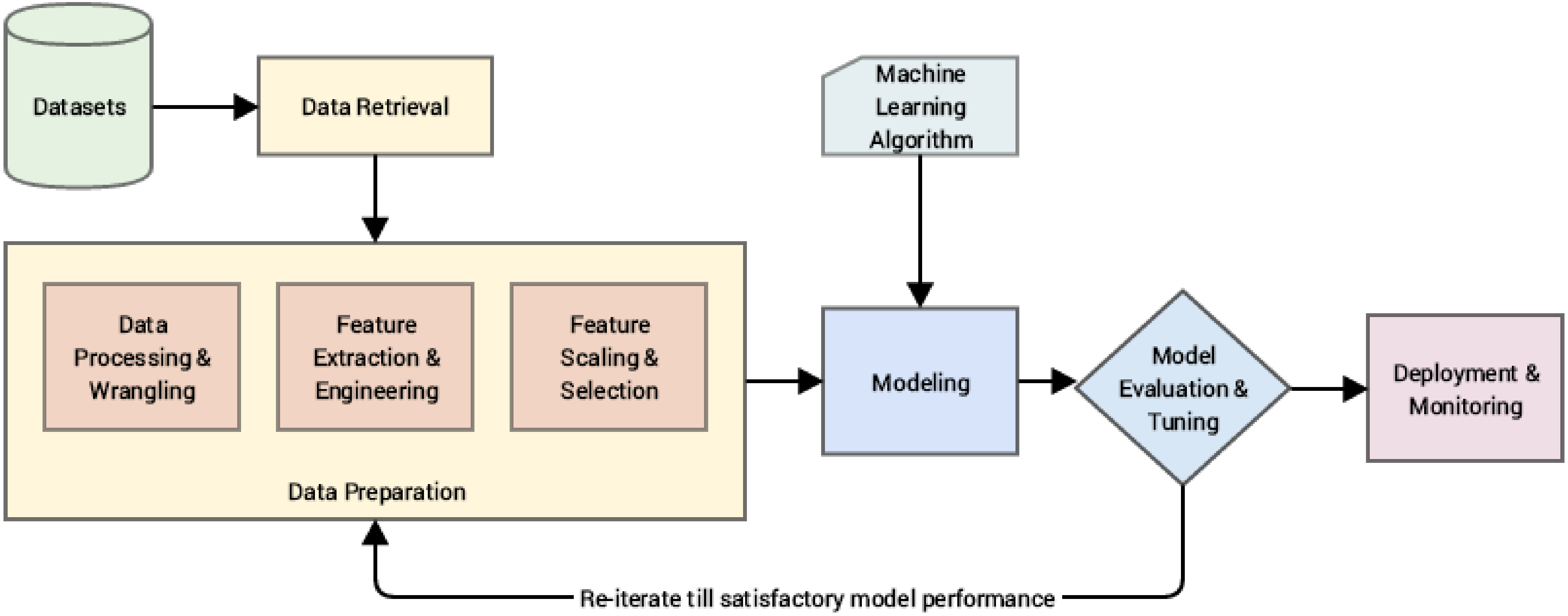
$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, L(x, y, \theta)$$

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

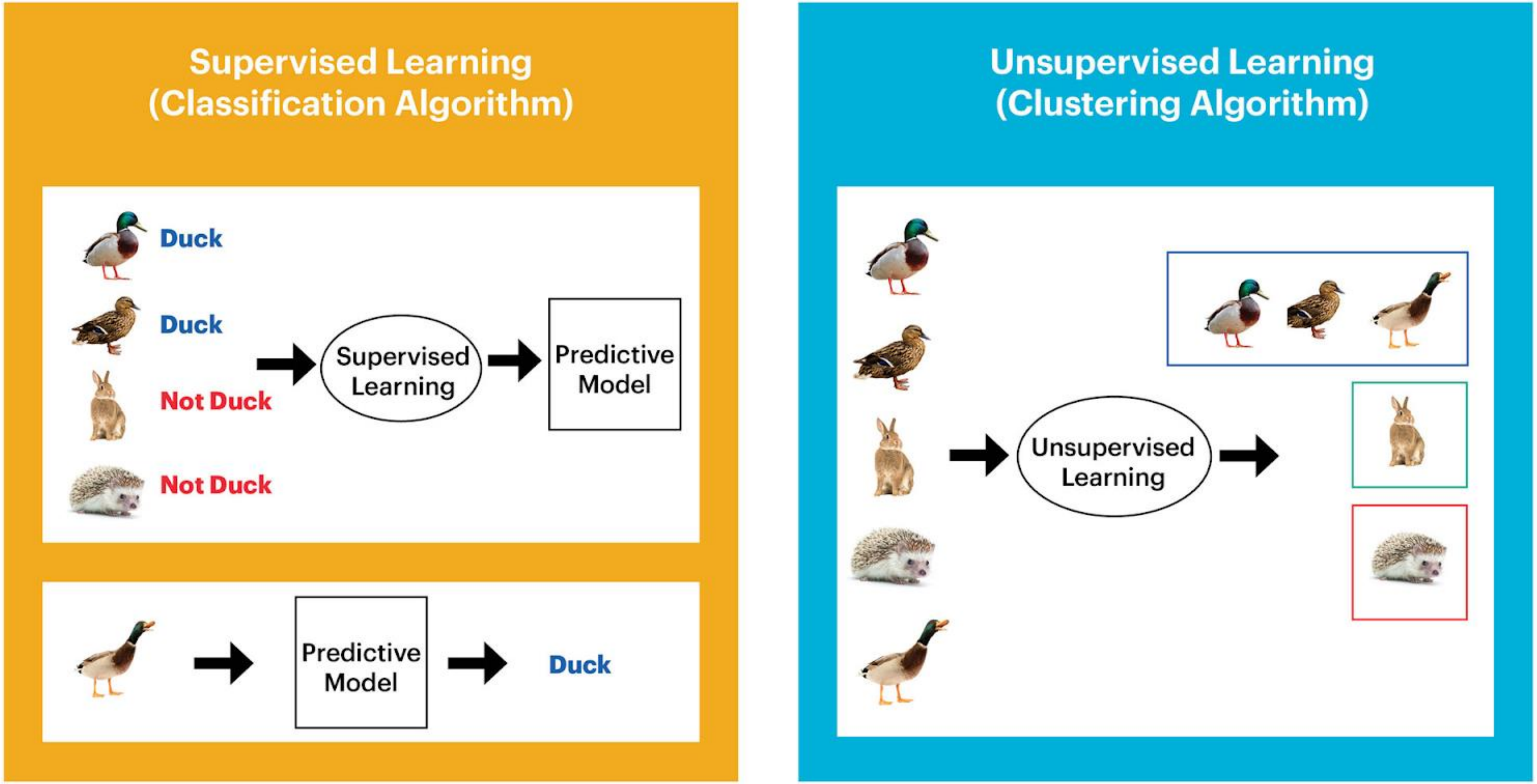
- Machine Learning
  - Estadística
  - Optimización
  - Algebra lineal
  - Matemática numérica
  - Computer Science
  - …

HEY Machine Learning WHY DO YOU ALWAYS WEAR THAT MASK?
Machine Learning
Machine Learning
Machine Learning
Statistics
LET'S KEEP THIS ON.
Machine Learning

Proceso

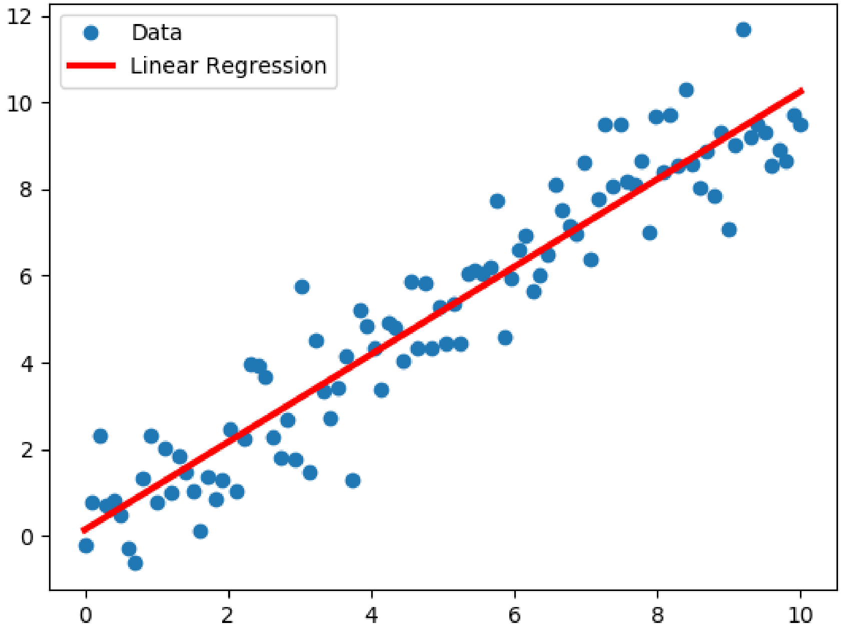

Datasets
Data Retrieval
Machine Learning Algorithm
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate till satisfactory model performance

Supervised v.s. Unsupervised Learning



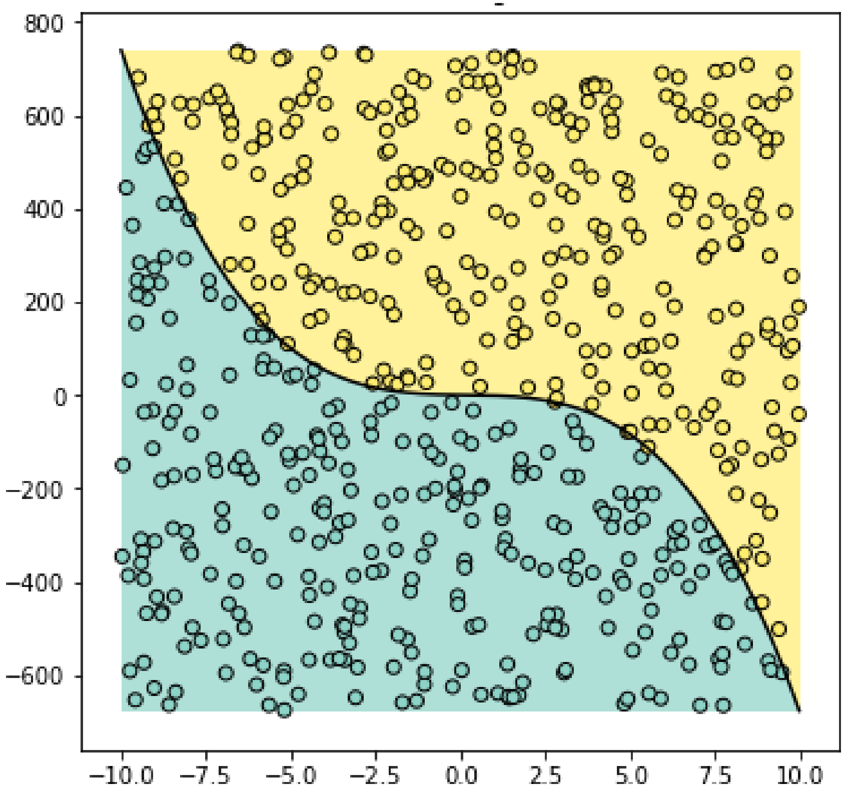
Supervised Learning

Regression

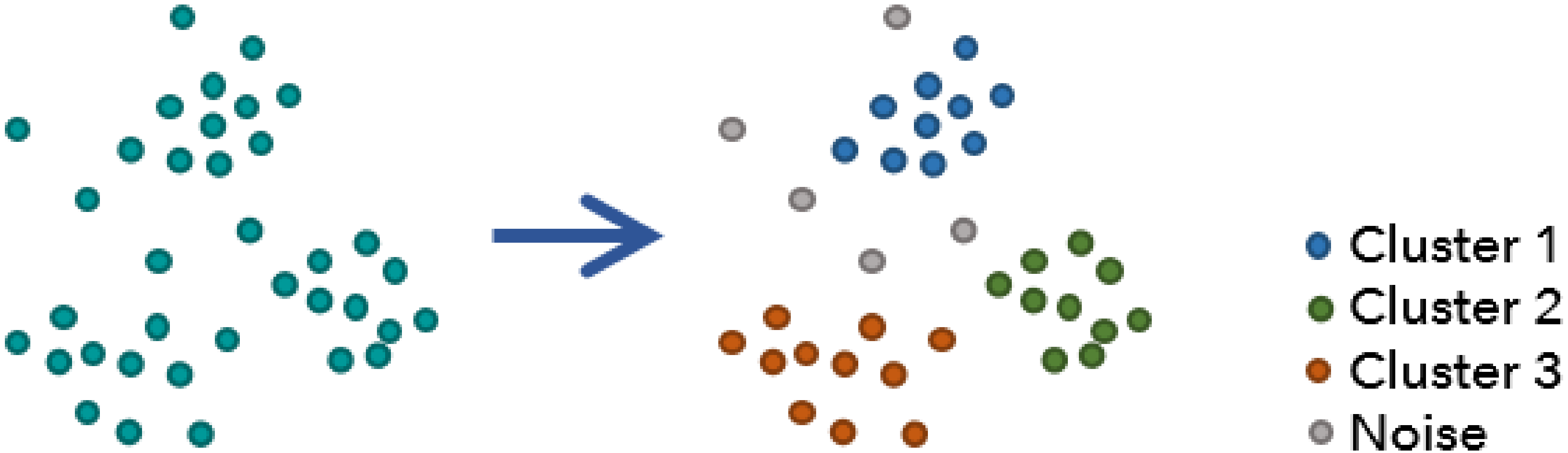


$$f(x,\theta) = \theta_0 + \theta_1 x$$

Classification

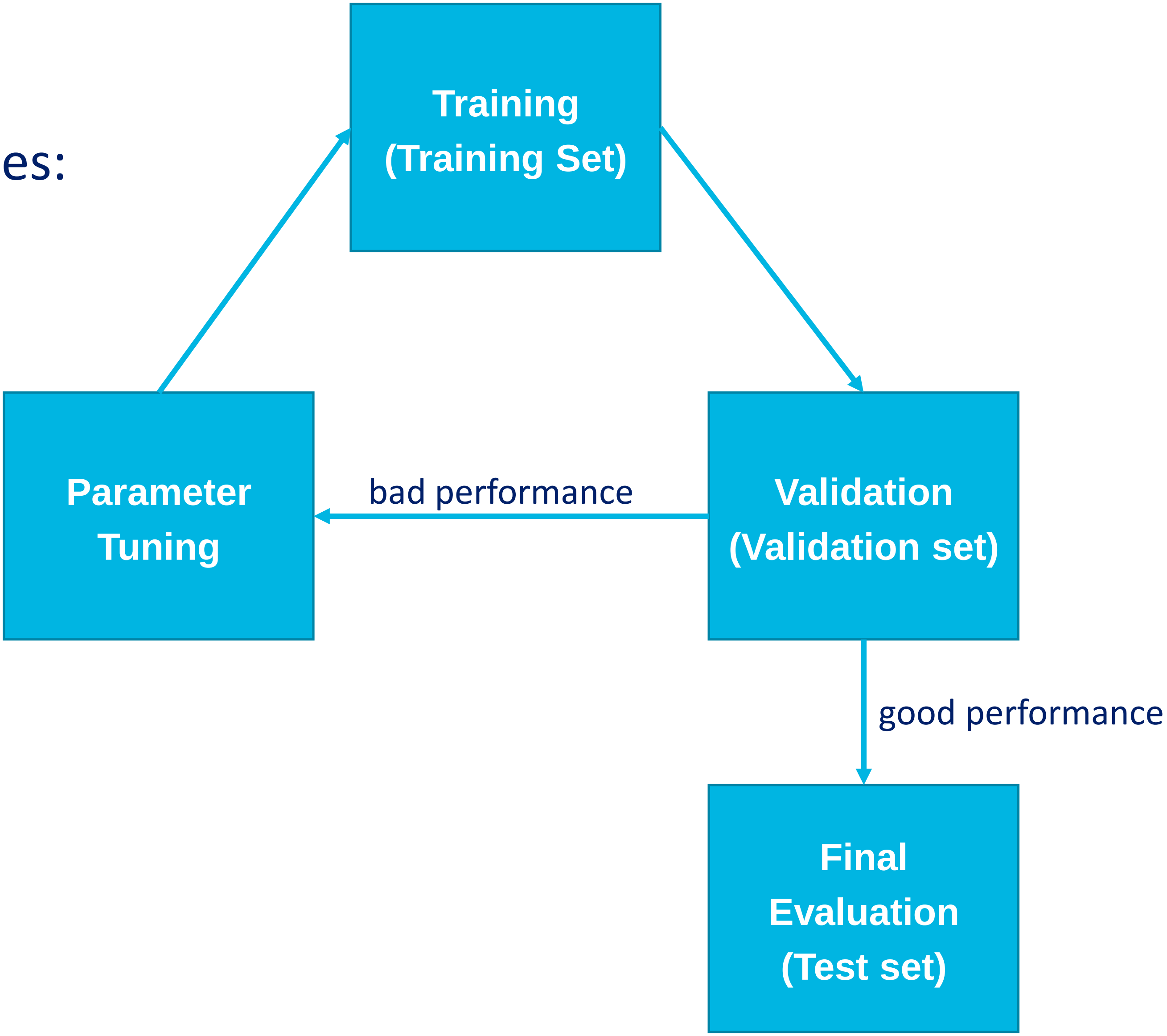


Unsupervised Learning (Clustering)

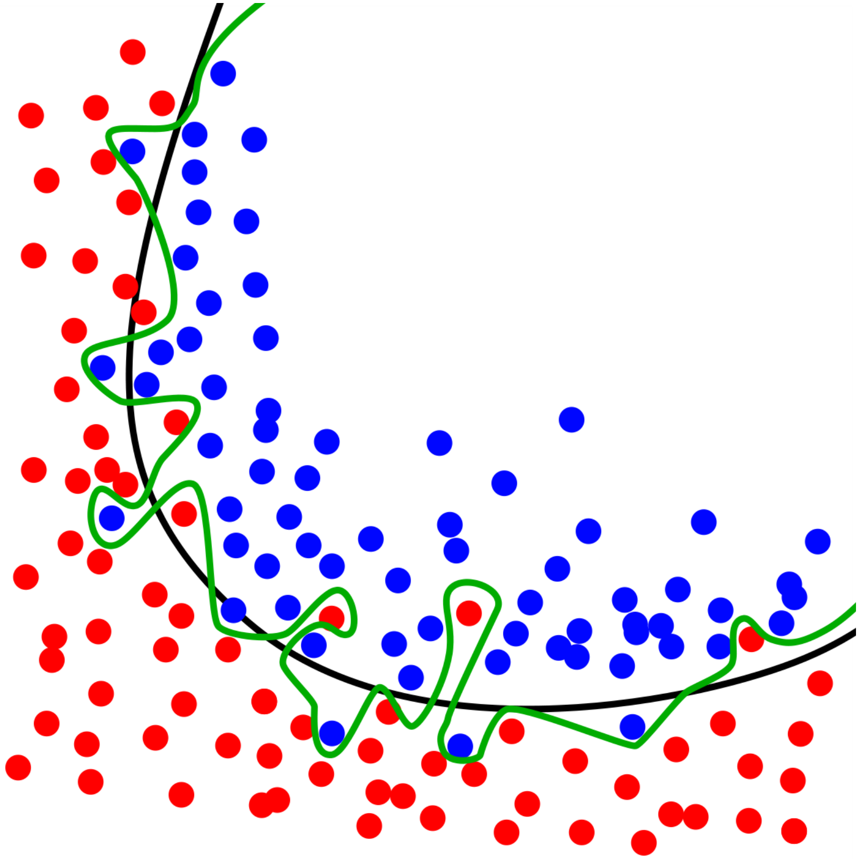


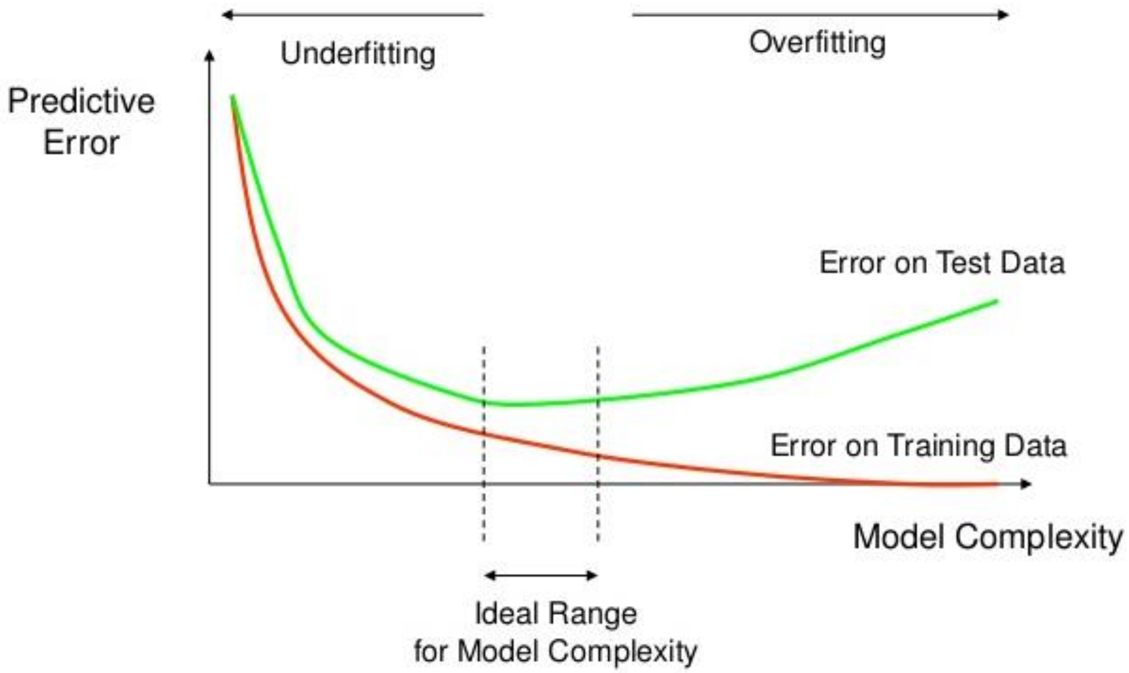
Datasets

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

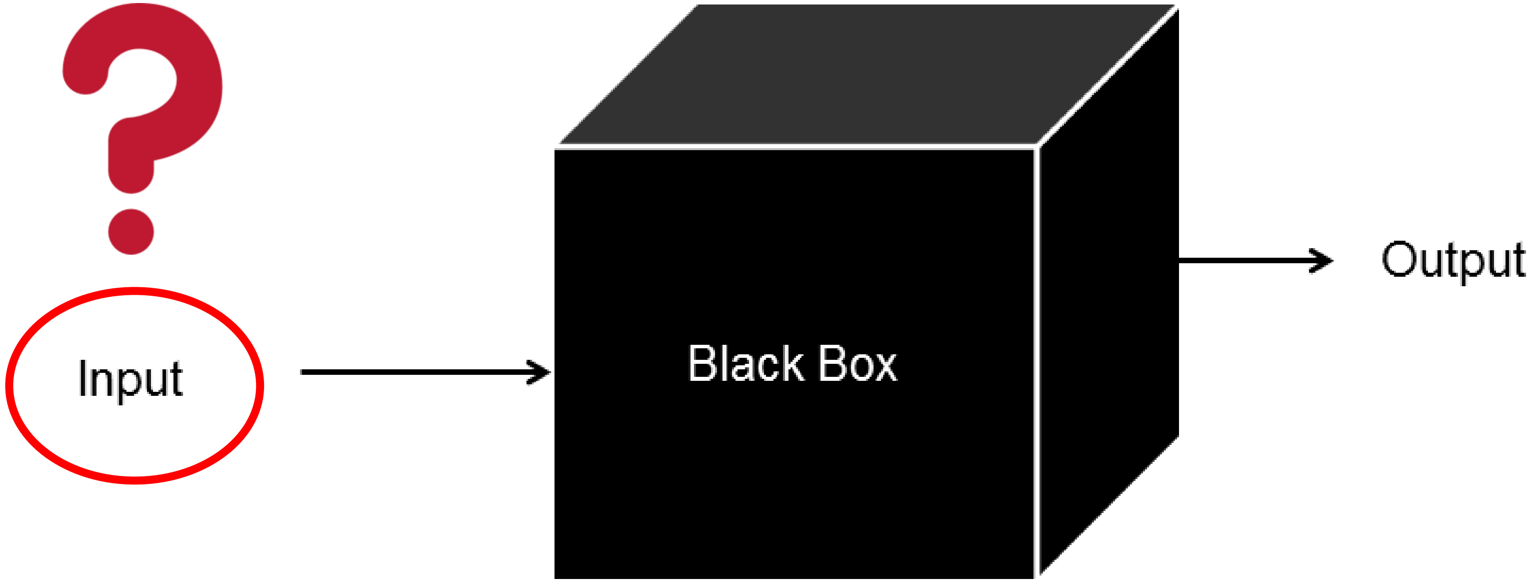


Overfitting



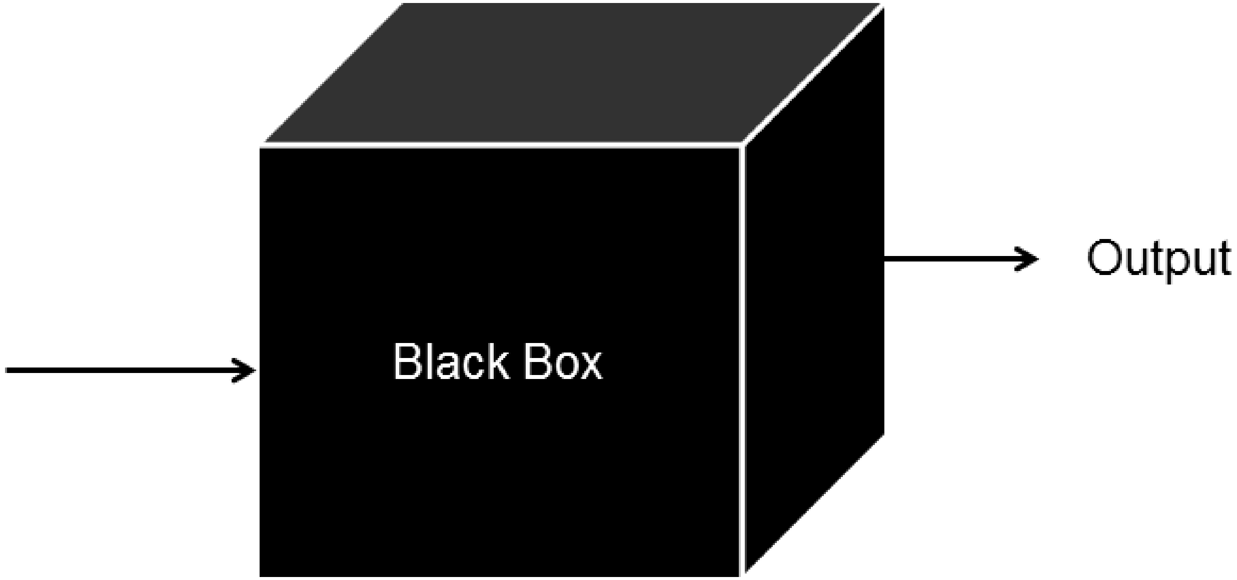


Que es el Input / Formato de los datos?



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, …)
- Texto
- Imágenes (Deep Learning)
- Audio (Deep Learning)

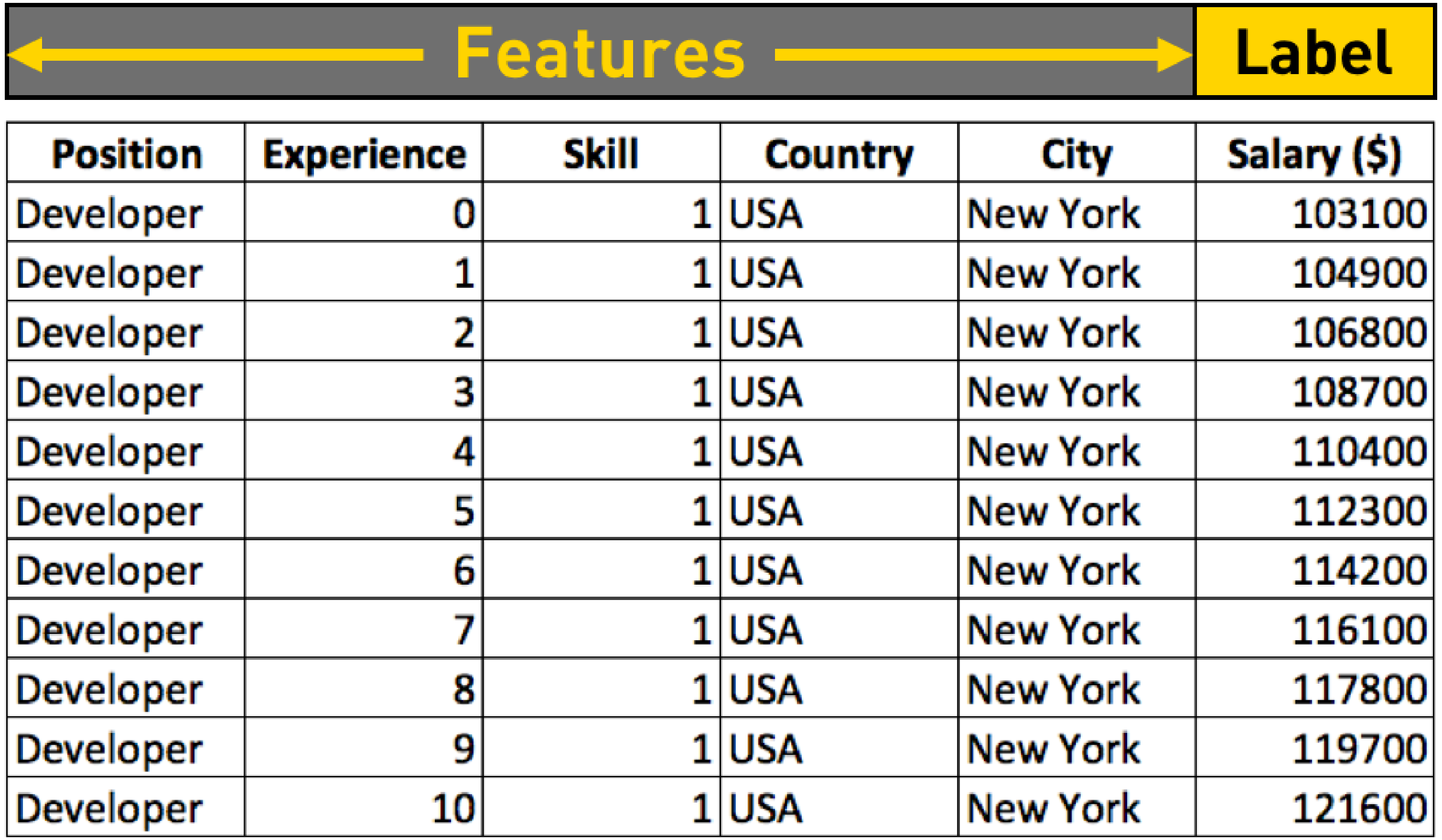
$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

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

Matriz = Tabula
Features & Label = Columnas

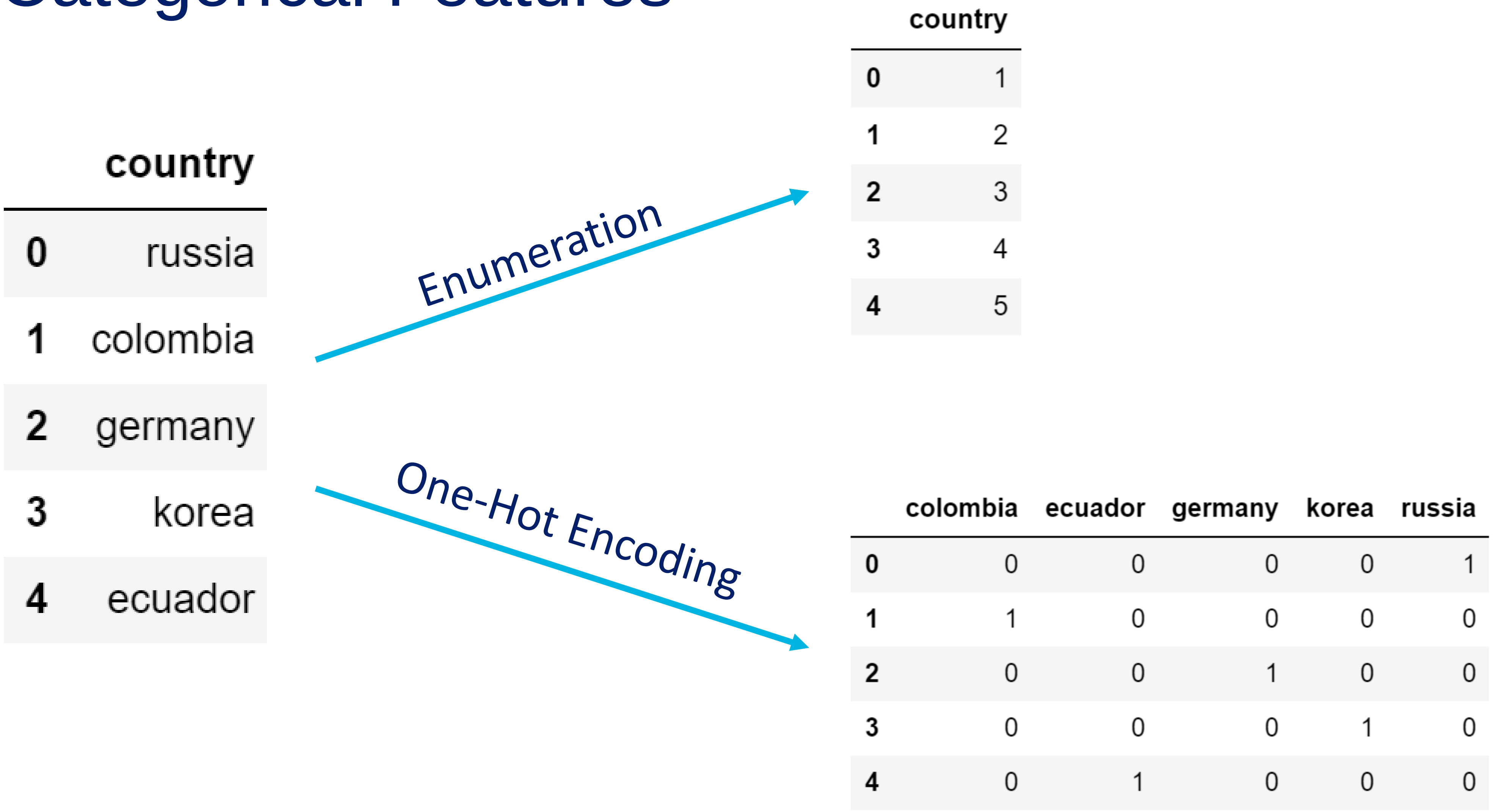
$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

m x n
n: # features
m: # samples



Features					Label
Position	**Experience**	**Skill**	**Country**	**City**	**Salary ($)**
Developer	0	1	USA	New York	103100
Developer	1	1	USA	New York	104900
Developer	2	1	USA	New York	106800
Developer	3	1	USA	New York	108700
Developer	4	1	USA	New York	110400
Developer	5	1	USA	New York	112300
Developer	6	1	USA	New York	114200
Developer	7	1	USA	New York	116100
Developer	8	1	USA	New York	117800
Developer	9	1	USA	New York	119700
Developer	10	1	USA	New York	121600

Categorical Features



	country
0	russia
1	colombia
2	germany
3	korea
4	ecuador

Enumeration

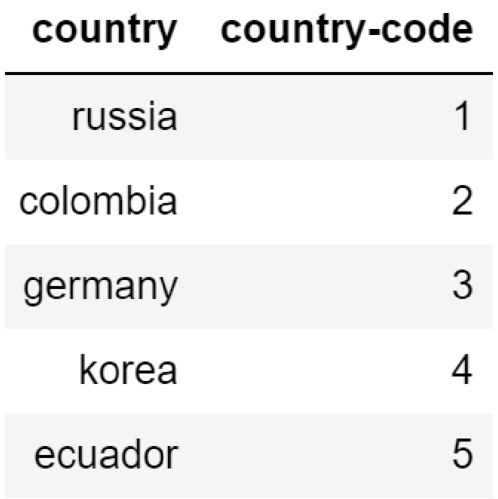
	country
0	1
1	2
2	3
3	4
4	5

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

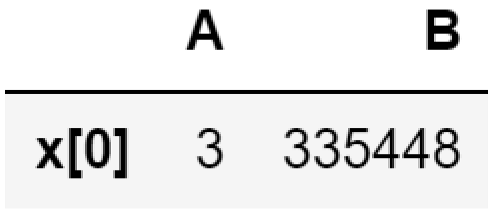
- **Enumeration**
  - Distancia euclidiana da falsa información
    - $|Russia\ - Colombia| = |1 - 2| = 1$
    - $|Colombia\ - Ecuador| = |2 - 5| = 3$



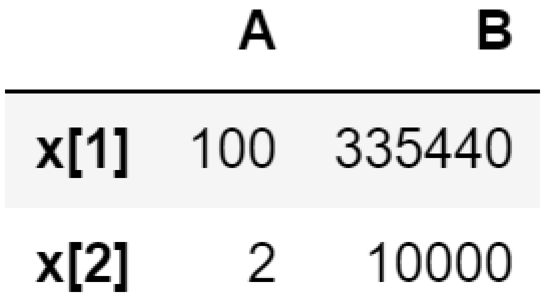
country	country-code
russia	1
colombia	2
germany	3
korea	4
ecuador	5

- **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
    - La distancia euclidiana entre todos los puntos se aproxima a una constante
    - Uso de memoria

Similaridad ?

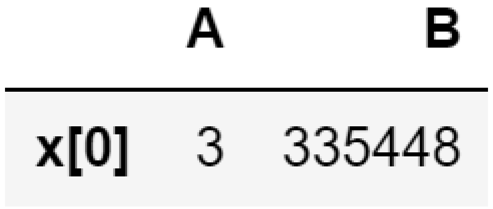


	A	B
x[0]	3	335448

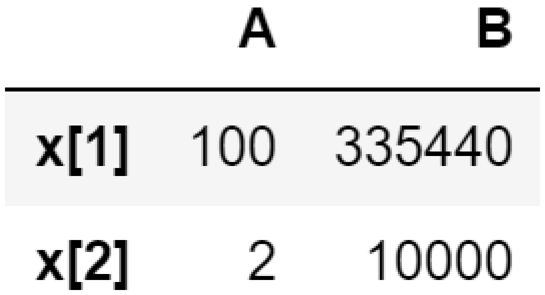


	A	B
x[1]	100	335440
x[2]	2	10000

Similaridad ?



	A	B
x[0]	3	335448



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

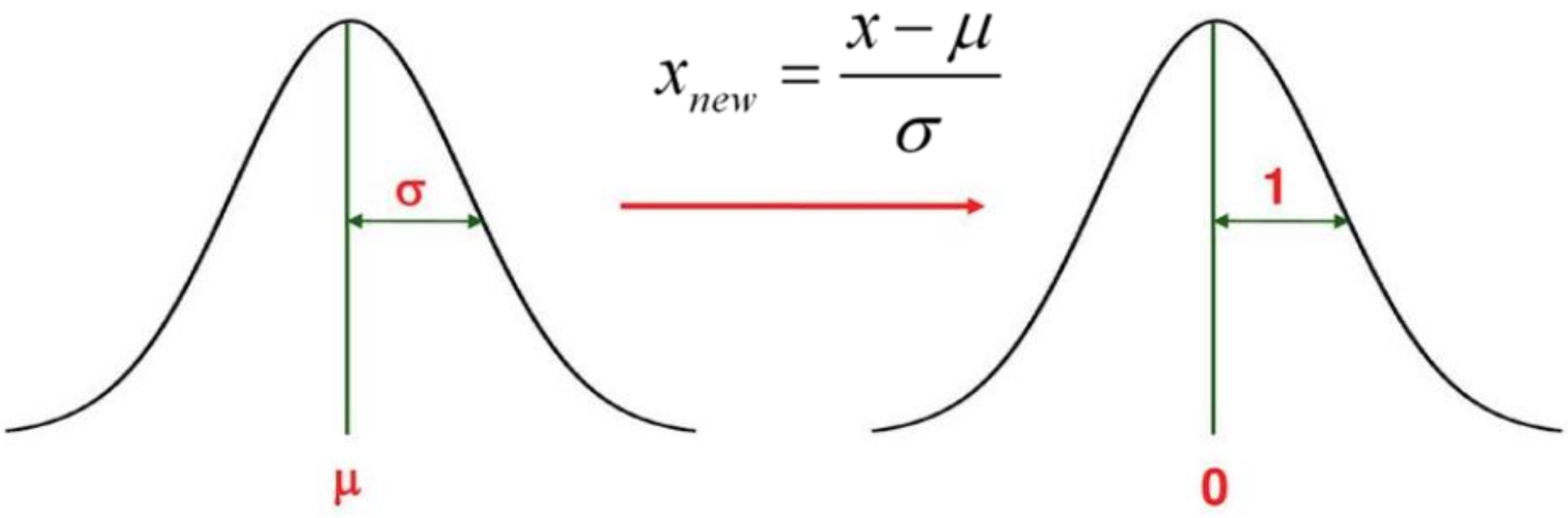
Distancia Euclidiana

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

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

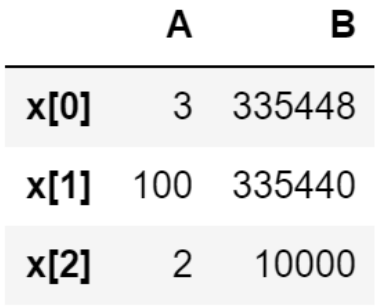
Y si A son [metros] y B son [milimetros] ?

Standardization

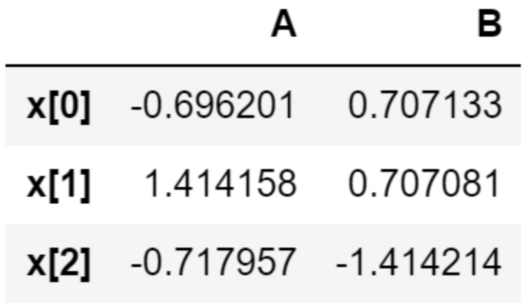


$$x_{new} = \frac{x - \mu}{\sigma}$$

[-1, 1]



	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

Dificultades

- Entender los datos
- Definir la tarea
- Datos en un formato adecuado para entrenar modelos
- Conseguir suficiente datos
- Seleccionar un modelo y encontrar los mejores parámetros
- Computación / Memoria

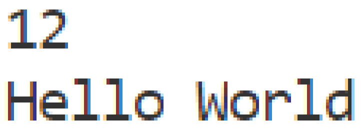
Python crashcourse

Code Example 1

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



```
12
Hello World
```

- No datatype declaration
- Variable assignment with `=`
  - First assignment creates variable
- Comments with #
- Logical operators are words: `and, or, not`
- Special use of + for string concatenation
- Printing command: `print()`
- Scope declaration with indentations (no {})

Naming Rules

- 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
```

- Floats

```
x = 3.256
y = 5 / 2.0 # result is 2.5
```

- Strings

```
x = "Machine Learning"
y = ‘Machine Learning’
```

- Boolean

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

Conditional Branching

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

Loops

```
# 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

```
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
```

$A \cap B$

$A \cup B$

$A \backslash \mathrm{B}$

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

p1 = Person("John", 36)

print(p1.name)
print(p1.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))
```

Files

```
# 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")

# Method B:
fp = open(filepath, 'w')
fp.write("test\n")
fp.close()
```

Exception Handling

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

String Methods

```
age = 21
print("Alejandra is {} years old".format(age))

price = 100000.2356412
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
```

```
import pandas as pd
import numpy as np
import tensorflow as tf

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

Virtual Environments

```
pip install virtualenv
virtualenv myenv
source myenv/bin/activate
pip install pandas
```

- Creates an isolated Python environment
  - Helps to avoid dependency conflicts