

# part1

April 7, 2021

## 1 Machine Learning Project 1 ( Part 1 )

### 1.1 UNICAMP

### 1.2 Dataset 1

Load Datasets

```
[22]: import pandas as pd
import numpy as np

df = pd.read_table("cluster.dat",header=None, sep="\s+")
datos = pd.DataFrame(df)

datos
```

```
[22]:      0      1
0    1555.0  28.65
1    1490.0  27.55
2    1445.0  28.35
3    1415.0  28.80
4    1375.0  28.05
..     ...   ...
568  3420.0  27.95
569  3465.0  26.85
570  3525.0  26.00
571  3570.0  26.15
572  3440.0  25.60

[573 rows x 2 columns]
```

Transformamos el dataset a un array de nummpy

```
[23]: datos=datos.to_numpy()
datos
```

```
[23]: array([[1555. ,  28.65],
          [1490. ,  27.55],
          [1445. ,  28.35],
```

```
...,
[3525. , 26. ],
[3570. , 26.15],
[3440. , 25.6 ]])
```

Shuffle data

```
[24]: np.random.shuffle(datos)
      datos
```

```
[24]: array([[3385. , 20. ],
[2075. , 8.75],
[ 830. , 18.5 ]],
...,
[1675. , 11.5 ],
[1860. , 3.45],
[1300. , 29.15]])
```

Split data in train and test: - 90% train - 10% test

```
[25]: datos_train = datos[:int(0.9*len(datos))]
      datos_test = datos[int(0.9*len(datos)):]

      print("Train", datos_train.shape)
      print("Test", datos_test.shape)
```

Train (515, 2)

Test (58, 2)

```
[26]: x_train = datos_train[:,0]
      y_train = datos_train[:,1]
```

### 1.2.1 Normalization

Min max Scaler

```
[29]: def normalization(x,y,x_max, x_min , y_max, y_min):
      print("Normalizing using parameters: x", x_max,"\t",x_min)
      print("Normalizing using parameters: y", y_max,"\t",y_min)
      x=(x-x_min)/(x_max-x_min)
      y=(y-y_min)/(y_max-y_min)
      return x,y

      x_max = x_train.max()
      x_min = x_train.min()
      y_max = y_train.max()
      y_min = y_train.min()
```

```
[31]: x_n_train, y_n_train = normalization(x_train,y_train,x_max,x_min,y_max,y_min)
      print("x_train",x_n_train[:3] , "...")
      print("y_train",y_n_train[:3] , "...")
```

Normalizing using parameters: x 3635.0                      335.0

Normalizing using parameters: y 28.8      1.95

x\_train [0.92424242 0.52727273 0.15            ] ...

y\_train [0.67225326 0.25325885 0.61638734] ...