TC1002S Herramientas computacionales: el arte de la analítica

This is a notebook with all your work for the final evidence of this course

Niveles de dominio a demostrar con la evidencia

SING0202A

Interpreta interacciones entre variables relevantes en un problema, como base para la construcción de modelos bivariados basados en datos de un fenómeno investigado que le permita reproducir la respuesta del mismo. Es capaz de construir modelos bivariados que expliquen el comportamiento de un fenómeno.

Student information

- Name: Arturo Azael Godinez Rodriguez
- ID: A01641179
- My carreer: Ingeneria en Robotica y sistemas digitales

Importing libraries

```
import numpy as np                  # For array
import pandas as pd                 # For data handling
import seaborn as sns               # For advanced plotting
import matplotlib.pyplot as plt     # For showing plots
```

PART 1

Use your assigned dataset

A1 Load data

```
# Define where you are running the code: colab or local
RunInColab          = True     # (False: no  | True: yes)

# If running in colab:
if RunInColab:
    # Mount your google drive in google colab
    from google.colab import drive
    drive.mount('/content/drive')

    # Find location
    #!pwd
    #!ls
    #!ls "/content/drive/My Drive/Colab Notebooks/MachineLearningWithPython/"

    # Define path del proyecto
    Ruta            = "/content/drive/My Drive/content/drive/My Drive/"

else:
    # Define path del proyecto
    Ruta            = ""
```

```
Mounted at /content/drive
```

```
url = "drive/My Drive/Evidencia/A01641179.csv"

# Load the dataset
df  = pd.read_csv(url)
df
```

	Unnamed: 0	x1	x2
0	0	0.560778	0.829499
1	1	-0.008038	0.381122
2	2	1.541781	-0.346447
3	3	1.950773	0.068763
4	4	1.650663	-0.328446
...	...	...	...
1695	1695	-0.668243	0.772759
1696	1696	1.701226	-0.144192
1697	1697	1.941233	0.000987
1698	1698	1.986689	0.309273
1699	1699	0.288874	-0.128405

1700 rows × 3 columns

A2 Data managment

Print the first 7 rows

```
df.head(7)
```

	Unnamed: 0	x1	x2
0	0	0.560778	0.829499
1	1	-0.008038	0.381122
2	2	1.541781	-0.346447
3	3	1.950773	0.068763
4	4	1.650663	-0.328446
5	5	0.312037	0.978508
6	6	1.454846	-0.363242

Print the first 4 last rows

```
df.tail(4)
```

	Unnamed: 0	x1	x2
1696	1696	1.701226	-0.144192
1697	1697	1.941233	0.000987
1698	1698	1.986689	0.309273
1699	1699	0.288874	-0.128405

How many rows and columns are in your data?

Use the `shape` method

```
total_rows=len(df.axes[0])
print("Numero de filas: "+str(total_rows))
total_rows=len(df.axes[1])
print("Numero de columnas: "+str(total_rows))
```

```
Numero de filas: 1700
Numero de columnas: 3
```

Print the name of all columns

Use the `columns` method

```
print(df.keys())
```

```
Index(['Unnamed: 0', 'x1', 'x2'], dtype='object')
```

What is the data type in each column

Use the `dtypes` method

```
datatypes = df.dtypes
datatypes
```

```
Unnamed: 0      int64
x1            float64
x2            float64
dtype: object
```

What is the meaning of rows and columns?

```
# Your responses here

#1) Lo vemos por filas tenemos en  la primera los encabezados que nos indica las variables y despues tenemos 2 entradas de datos que esta enu

"""2) Lo vemos por columnas los datos tenemos hasta la primera columna que no tiene un encabezado que se utlizada para enumerar la entrada de
 la entrada de datos de la variable x1 y la segunda nos indica la entrada de datos de la variable x2"""

#3) Cuando lo vemos completo, son dos variables que tienen dos entradas de datos y cada entrada esta enumerada hasta llegar al 1699
```

Print a statistical summary of your columns

```
print(df['x1'].mean()) #Calculate the average
print(df['x2'].mean()) #Calculate the average
df.describe()
```

```
0.5011117528162025
0.25267001656880317
```

	Unnamed: 0	x1	x2
count	1700.000000	1700.000000	1700.000000
mean	849.500000	0.501112	0.252670
std	490.892045	0.868836	0.496579
min	0.000000	-1.152887	-0.626515
25%	424.750000	-0.033787	-0.202554
50%	849.500000	0.496433	0.252857
75%	1274.250000	1.036949	0.704387
max	1699.000000	2.125947	1.108099

```
# 1) What is the minumum and maximum values of each variable
df.max()

# 2) What is the mean and standar deviation of each variable
df.mean() #Calculate the average
print(df['x1'].std()) #Standar deviation
print(df['x2'].std()) #Standar deviation


# 3) What the 25%, 50% and 75% represent?
"""El porcentaje que nos arroja signfica que es el percentil
que esta debajo de este por el ejemplo el 25%,
el percentil 25% es el valor que está por debajo del 25% de los datos """
```

```
0.8688361068000848
```

Rename the columns using the same name with capital letters

```
25%, (el percentil 25% es el valor que está por debajo del 25% de los datos
```

```
df.rename(columns={'x1': 'X1', 'x2': 'X2'})
```

	Unnamed: 0	X1	X2
0	0	0.560778	0.829499
1	1	-0.008038	0.381122
2	2	1.541781	-0.346447
3	3	1.950773	0.068763
4	4	1.650663	-0.328446
...	...	...	...
1695	1695	-0.668243	0.772759
1696	1696	1.701226	-0.144192
1697	1697	1.941233	0.000987
1698	1698	1.986689	0.309273
1699	1699	0.288874	-0.128405

1700 rows × 3 columns

Rename the columns to their original names

```
df.rename(columns={'x1': 'x1', 'x2': 'x2'})
```

	Unnamed: 0	x1	x2
0	0	0.560778	0.829499
1	1	-0.008038	0.381122
2	2	1.541781	-0.346447
3	3	1.950773	0.068763
4	4	1.650663	-0.328446
...	...	...	...
1695	1695	-0.668243	0.772759
1696	1696	1.701226	-0.144192
1697	1697	1.941233	0.000987
1698	1698	1.986689	0.309273
1699	1699	0.288874	-0.128405

1700 rows × 3 columns

Use two different alternatives to get one of the columns

```
columna_nombre = df["x1"]
columna_nombre
```

```
0       0.560778
1      -0.008038
2       1.541781
3       1.950773
4       1.650663
          ...
1695   -0.668243
1696    1.701226
1697    1.941233
1698    1.986689
1699    0.288874
Name: x1, Length: 1700, dtype: float64
```

```
columna_nombre = df.loc[:, "x2"]
columna_nombre
```

```
0       0.829499
1       0.381122
2      -0.346447
3       0.068763
4      -0.328446
          ...
1695    0.772759
1696   -0.144192
1697    0.000987
1698    0.309273
1699   -0.128405
Name: x2, Length: 1700, dtype: float64
```

Get a slice of your data set: second and thrid columns and rows from 62 to 72

```
columnas_seleccionadas = df[["x1", "x2"]]
columnas_seleccionadas.iloc[62:73]
```

	x1	x2
62	0.888268	0.085673
63	0.906120	0.387119
64	-0.368457	0.883192
65	0.173414	-0.167169
66	0.695985	0.827573
67	1.932337	0.270719
68	0.046146	0.541350
69	0.373738	0.889317
70	0.378543	0.945828
71	-0.109662	0.995726
72	0.998030	0.025771

For the second and thrid columns, calculate the number of null and not null values and verify that their sum equals the total number of rows

```
# calculate the number of null and not null values
num_null = columnas_seleccionadas.isnull().sum().sum()
print(num_null)
num_not_null = columnas_seleccionadas.notnull().sum().sum()
print(num_not_null)

# verify that their sum equals the total number of rows
num_rows = df.shape[0]
num_rows
num_null + num_not_null == num_rows
```

```
0
3400
False
```

Discard the last column

```
df1=df.drop(df.tail(3).index,inplace = True)
print(df)
```

```
      Unnamed: 0        x1        x2
0              0  0.560778  0.829499
1              1 -0.008038  0.381122
2              2  1.541781 -0.346447
3              3  1.950773  0.068763
4              4  1.650663 -0.328446
...          ...       ...       ...
1689        1689  0.341108 -0.129838
1690        1690  0.148309  0.000448
1691        1691  1.606403 -0.191328
1692        1692 -0.773588  0.563455
```

```
1693        1693  1.539968 -0.326243

[1694 rows x 3 columns]
```

Questions

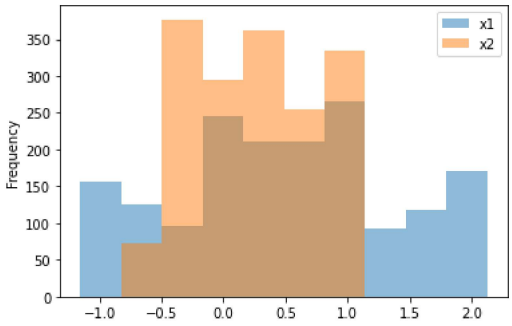
Based on the previos results, provide a description of yout dataset

Your response: Viendo los datos anteriores, pudimos tener varios datos a varacion de las funciones que tenemos con tan solo dos variables

A3 Data visualization

Plot in the same figure the histogram of the two variables

```
columnas_seleccionadas1 = df[["x1", "x2"]]
columnas_seleccionadas1.plot.hist(alpha=0.5)
plt.show()
```

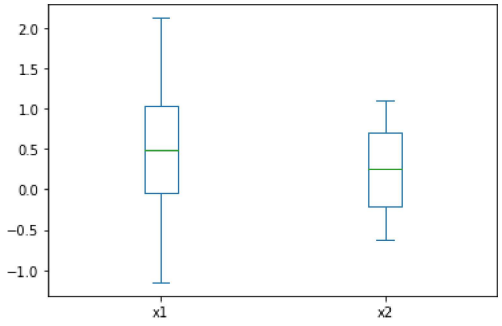


Based on this plots, provide a description of your data:

Your response here: La tabla muestra que los datos con la x2 son mayores que la variable x1 por una gran diferencia

Plot in the same figure the boxplot of the two variables

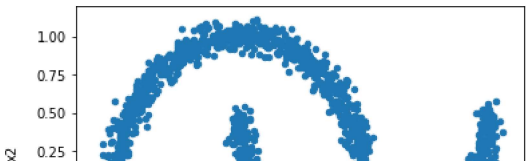
```
columnas_seleccionadas1.plot.box()
plt.show()
```



Scatter plot of the two variables

```
columnas_seleccionadas1.plot.scatter(x='x1', y='x2')
plt.show()
```

```
/usr/local/lib/python3.9/dist-packages/pandas/plotting/_matplotlib/core.py:1114: UserWarning: No data f
  scatter = ax.scatter(
```



Questions

Based on the previos plots, provide a description of yout dataset

Your response: Lo primero que todo tenemos lo que parecen 2 funciones cuadraticas, si seguimos observando que empiezan donde terminan como quiero decir esto los datos de la primera funcion cuadrada empiezan en 0 y terminan en 0.Ademas tenemos que ninguna de estas funciones cuadraticas se toca. Y tenemos en un análisis 4 cluster.

Haz doble clic (o ingresa) para editar

A4 Kmeans

Do Kmeans clustering assuming a number of clusters accorging to your scatter plot

```
from sklearn.cluster import KMeans

# Define number of clusters
K  = 4 # Let's assume there are 2,3,4,5...? clusters/groups

#Crear el objeto o el modelo de MachineKMeans
km = KMeans(n_clusters =K,n_init="auto")

# Do K-means clustering (assing each point in the dataset to a cluster)
yestimated = km.fit_predict(df)


# Print estimated cluster of each point in the dataset
yestimated
```

```
array([0, 0, 0, ..., 1, 1, 1], dtype=int32)
```

Add to your dataset a column with the assigned cluster to each data point

```
df2 = df.assign(yestimated=yestimated)
```

Print the number associated to each cluster

```
df2
```

	Unnamed: 0	x1	x2	yestimated
0	0	0.560778	0.829499	0

Print the centroids

```
km.cluster_centers_
```

```
array([[2.15500000e+02, 4.91786683e-01, 2.64386874e-01],
       [1.48650000e+03, 5.02389001e-01, 2.75846065e-01],
       [6.45500000e+02, 4.81779154e-01, 2.32316050e-01],
       [1.06950000e+03, 5.21545560e-01, 2.39266468e-01]])
```

1689	1689	0.341108	-0.129838	1

Print the intertia metric

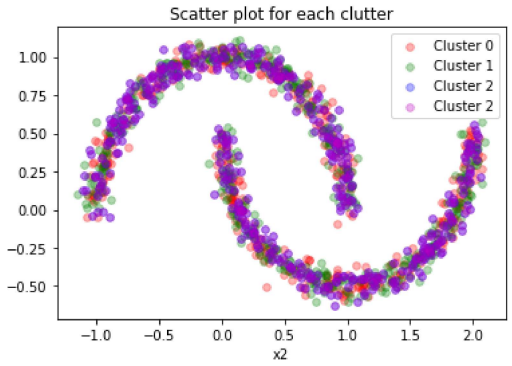
1691	1691	1.606403	0.191328	1

```
km.inertia_
```

```
25337803.046920665
```

Plot a scatter plot of your data assigned to each cluster. Also plot the centroids

```
# Get a dataframe with the data of each clsuter

df_1 = df2[df2.yestimated==0]
df_2 = df2[df2.yestimated==1]
df_3 = df2[df2.yestimated==2]
df_4 = df2[df2.yestimated==2]

# Scatter plot of each cluster

plt.scatter(df_1.x1, df_1.x2, label="Cluster 0", c='r',marker='o',s=32,alpha=0.3)
plt.scatter(df_2.x1, df_2.x2, label="Cluster 1", c='g',marker='o',s=32,alpha=0.3)
plt.scatter(df_3.x1, df_3.x2, label="Cluster 2", c='b',marker='o',s=32,alpha=0.3)
plt.scatter(df_4.x1, df_4.x2, label="Cluster 2", c='m',marker='o',s=32,alpha=0.3)

plt.title("Scatter plot for each clutter")
plt.xlabel('x1')
plt.xlabel('x2')
plt.legend()
plt.show()
```



Questions

Provides a detailed description of your results

Your response: Al momento de plotter los datos podemos ver la relacion que tienen con los datos inciales que realmente se asemejan por no decir que son identicos a simple vista.

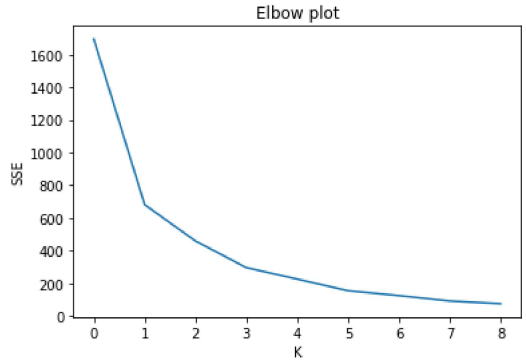
A5 Elbow plot

Compute the Elbow plot

```
#Intialize a list to hold sum of squared error (sse)
sse = []
```

```
# Define values of k
k_rng=range(1,10)

# For each k
for k in k_rng:
  km = KMeans(n_clusters=k,n_init="auto")
  km.fit_predict(df[['x1','x2']])
  sse.append(km.inertia_)


plt.title('Elbow plot')
plt.xlabel('K')
plt.ylabel('SSE')
plt.plot(sse)
```

```
[<matplotlib.lines.Line2D at 0x7f493c298400>]
```



Questions

What is the best number of clusters K? (argue your response)

Your response: Haciendo la simulacion con diversos cluster se puede observar que el numero 4 es el mejor a elegir para este caso porque al momento de ver la grafica tenemos que SSE tiende a bajar con el paso de K en otras palabras se dice que baja el error. Ademas se habian seleccionado por visualmente ya que son 4 puntos que se acercan entre si.

Does this number of clusters agree with your inital guess? (argue your response)

Your response: Si, gracias a lo anterior dicho por ver graficamente se pudo obtener.

PART 2

Load and do clustering using the "digits" dataset

1) Load the dataset using the "load_digits()" function from "sklearn.datasets"

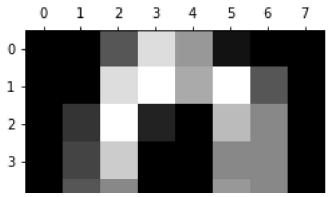
```
from sklearn.datasets import load_digits

digits = load_digits()
print(digits.data.shape)
```

```
(1797, 64)
```

2) Plot some of the observations

```
plt.gray()
plt.matshow(digits.images[0])
plt.show()
```

```
<Figure size 432x288 with 0 Axes>
```



3) Do K means clustering

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import scale

data = scale(digits.data)
n_samples, n_features = data.shape
n_digits = len(np.unique(digits.target))
labels = digits.target

kmeans = KMeans(init='k-means++', n_clusters=n_digits, n_init=10)
kmeans.fit(data)
print(f'K-Means clustering digits data: {kmeans.labels_}')
```

```
K-Means clustering on the handwritten digits data: [1 3 4 ... 3 7 7]
```

4) Verify your results in any of the observations

```
df = pd.DataFrame(labels)
df
```

	0
0	0
1	1
2	2
3	3
4	4
...	...
1792	9
1793	0
1794	8
1795	9
1796	8

1797 rows × 1 columns

```
df = pd.DataFrame(kmeans.labels_)
df
```

	0
0	1

Questions

Provides a detailed description of your results.

Your response: Revisando los datos muchos de los datos coinciden con los datos en el clustering dando de forma acertiva, nuestra forma de verificar los datos obtenidos.

PART 3

Descipcion de tu percepcion del nivel de desarrollo de la subcompetencia

SING0202A Interpretación de variables

Escribe tu description del nivel de logro del siguiente criterio de la subcompetencia

Interpreta interacciones. Interpreta interacciones entre variables relevantes en un problema, como base para la construcción de modelos bivariados basados en datos de un fenómeno investigado que le permita reproducir la respuesta del mismo.

Tu respuesta: Durante la semana de la clase pudimos ver diversos archivos con datos algunas si pertenecian con un encabezado alguna variable algunas otras no lo hacian y teniamos que hacer un acomodo de las variable, con todo ello pudimos intrepetarlas con las graficas juntando dos variables y de ahi obtener una agrupacion para una aprendizaje no supervisado.

Escribe tu description del nivel de logro del siguiente criterio de la subcompetencia

Construcción de modelos. Es capaz de construir modelos bivariados que expliquen el comportamiento de un fenómeno.

Tu respuesta: Durante el proceso de creacion de los ultimos archvios pudimos lograr que una inteligencia fuera aprendiendo con tan solo darle los datos o como tambien es conocido el aprendizaje no supervisado para la obtencion de procesos sencillos o tediosos, tan solo con la agrupacion de datos a base de clustering.