K-means clustering

The notebook aims to study and implement a k-means clustering using "sklearn". A synthetic dataset will be used to identify clusters automatically using the K-means method.

Acknowledgments

- Inquiries: mauricio.antelis@tec.mx

Importing libraries

In [ ]:

```
# Import the packages that we will be using
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

# Note: specific functions of the "sklearn" package will be imported when nee
```

Importing data

In [ ]:

```
# Dataset url
path = "/home/alex/TC1002S/NotebooksStudents/A01639643/K_means/datasets/Synth

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

Undertanding and preprocessing the data

1. Get a general 'feel' of the data

In [ ]:

```
# Print the dataframe
df
```

Out[ ]:

	x1	x2	x3	x4	x5	x6
0	1.914825	-1.380503	-3.609674	4.236011	-5.158681	5.712978
1	1.356415	9.767893	7.263659	8.750819	5.568930	-6.039122
2	1.185186	11.528344	9.999419	7.890027	7.308210	-8.899397
3	-1.739155	12.648965	7.965588	7.850296	10.235743	-10.175542
4	7.890985	-3.210880	-7.672016	2.438106	3.310904	-3.308334

	x1	x2	x3	x4	x5	x6
...	...	...	...	...	...	...
1019	3.685106	-1.715503	-5.674443	6.510551	-0.121862	-6.166649
1020	-7.014173	-9.697874	4.093272	-0.590262	-9.882245	2.339336
1021	-2.993762	7.528182	7.877165	8.895835	9.318544	-7.445100
1022	4.576644	-1.720788	-6.581909	4.745839	1.497980	-4.828975
1023	2.616634	0.274593	-5.521864	9.582110	0.878266	-8.274990

In [ ]:
```
# get the number of observations and variables
print("The number of observations are: ", df.shape[0])
print("The number of variables are: ", df.shape[1])
```

```
The number of observations are:  1024
The number of variables are:  6
```

1. Drop rows with any missing values

In [ ]:
```
# Drop rows with NaN values if existing
df.dropna().describe()
print("The amount of NaN values in the dataset is: \n", df.isnull().sum())
df.notnull().sum()

# Print the new shape
df
```

```
The amount of NaN values in the dataset is:
 x1    0
x2    0
x3    0
x4    0
x5    0
x6    0
dtype: int64
```

Out[ ]:

	x1	x2	x3	x4	x5	x6
0	1.914825	-1.380503	-3.609674	4.236011	-5.158681	5.712978
1	1.356415	9.767893	7.263659	8.750819	5.568930	-6.039122
2	1.185186	11.528344	9.999419	7.890027	7.308210	-8.899397
3	-1.739155	12.648965	7.965588	7.850296	10.235743	-10.175542
4	7.890985	-3.210880	-7.672016	2.438106	3.310904	-3.308334
...	...	...	...	...	...	...
1019	3.685106	-1.715503	-5.674443	6.510551	-0.121862	-6.166649
1020	-7.014173	-9.697874	4.093272	-0.590262	-9.882245	2.339336
1021	-2.993762	7.528182	7.877165	8.895835	9.318544	-7.445100
1022	4.576644	-1.720788	-6.581909	4.745839	1.497980	-4.828975
1023	2.616634	0.274593	-5.521864	9.582110	0.878266	-8.274990