Data Analysis - Lab 4

M. Sébastien MASCHA & M. Sauvage Pierre

ISEP Paris – September 24th, 2019

Exercice C - Optimal cluster number in exoplanet data

In this exercice, we will try to guess the optimal number of clusters to be found in an artificial data set describing the atmospheric characteristics of exoplanets.

Import of libraries

This document has been done using python on Jupyter Notebook with the librairies:

- maths for sqrt, pi, exp
- Numpy to manipulate arrays
- pandas to import csv
- matplotlib to plot graphics
- seaborn to make your charts prettier (built on top of Matplotlib)
- sklearn : tools for data mining and data analysis
- mlxtend : tools for ploting PCA

In [13]:

```
# coding: utf-8

import data

from math import sqrt,pi,exp
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns; sns.set()

import sklearn
from sklearn import metrics
from sklearn.metrics import pairwise_distances
from sklearn.cluster import KMeans
```

Question 1 - Open the file

In [14]:

```
df = pd.read_csv("data/exo4_atm_extr.csv", sep =';')
df = df.drop(['Type'], axis=1)
print(df.shape)

df.head()
```

```
(1000, 11)
```

Out[14]:

	PH2O	PHe	PCH4	PH2	PN2	PNH3	PO2	PAr	PCO2	PSO2	PK
0	0.0	8.7	1.3	87.30	0.0	2.70	0.0	0.0	0.0	0.0	0.00
1	0.0	0.0	0.0	0.00	0.0	0.00	0.0	0.0	0.0	0.0	0.02
2	0.1	7.1	1.7	86.45	0.0	1.15	0.0	0.0	0.0	3.5	0.00
3	0.0	2.7	0.0	3.70	41.5	0.00	31.3	6.6	14.2	0.0	0.00
4	0.1	11.4	1.1	86.10	0.0	0.20	0.0	0.0	0.0	1.1	0.00

Question 2 - Write down the different properties of the Calinski-Harabasz and Davies- Bouldin indexes.

Calinski-Harabasz

- Not normalized
- Better when higher
- With balanced clusters, the CH index is generally a good criterion to indicate the correct number of clusters.

Davies- Bouldin indexes

- A lower DB value means a better clustering.
- This index is not normalized.
- It favors spherical clusters.
- It is biased so that it gives lower values with less clusters.

Question 3 - Calinski-Harabasz and Davies-Bouldin

In [19]:

```
kmeans_model = KMeans(n_clusters=3, random_state=1).fit(df.values)
labels = kmeans_model.labels_
print( "Calinski-Harabasz score is :" )
metrics.calinski_harabasz_score(df.values, labels)
```

```
Calinski-Harabasz score is :
```

Out[19]:

```
2527.9845312566085
```

In [20]:

```
print( "Davies-Bouldin score is :" )
metrics.davies_bouldin_score(df.values, labels)
```

```
Davies-Bouldin score is :
```

Out[20]:

```
0.3172965483768289
```

Question 4 - PCA

In [21]:

```
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(df)

principalDf = pd.DataFrame(data = principalComponents
             , columns = ['principal component 1', 'principal component 2'])
principalDf.head(5)
```

Out[21]:

	principal component 1	principal component 2
0	62.601028	20.801273
1	-11.658641	-23.016774
2	61.723247	20.286628
3	-20.497593	-18.427451
4	61.875945	20.367130

In [ ]:

```

```