K-Means con datos de lluvias de Australia

Tecnológico
de Monterrey

A01705420 - Alfonso Antonio Zazueta Bustillos
A01706155 - Manolo Ramírez Pintor
A01701969 - Enrique Guamán Herrera

Librerías:

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import sklearn
from sklearn.cluster import KMeans
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import scale
import sklearn.metrics as sm
from sklearn import datasets
from sklearn.metrics import confusion_matrix, classification_report
```

Cargar datos a memoria:

De esta manera los accesamos fácilmente llamando al mismo tiempo a Pandas

In [2]:

```
# Carga de datos
lluvias_sydney = pd.read_csv("weatherAUS_sydney.csv")
```

Revisar información de los datos:

Hacer una review de que todo ande bien en cuanto a valores y datos para que no falle.

In [3]:

```
lluvias_sydney.columns
```

Out[3]:

```
Index(['LaFech', 'MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine',
       'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am',
       'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Cloud9am', 'Cloud3pm',
       'Temp9am', 'Temp3pm', 'RainToday'],
      dtype='object')
```

In [4]:
```
lluvias_sydney.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 366 entries, 0 to 365
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   LaFech         366 non-null    int64  
 1   MinTemp        366 non-null    float64
 2   MaxTemp        366 non-null    float64
 3   Rainfall       366 non-null    float64
 4   Evaporation    366 non-null    float64
 5   Sunshine       366 non-null    float64
 6   WindGustSpeed  366 non-null    int64  
 7   WindSpeed9am   366 non-null    int64  
 8   WindSpeed3pm   366 non-null    int64  
 9   Humidity9am    366 non-null    int64  
 10  Humidity3pm    366 non-null    int64  
 11  Pressure9am    366 non-null    float64
 12  Pressure3pm    366 non-null    float64
 13  Cloud9am       366 non-null    int64  
 14  Cloud3pm       366 non-null    int64  
 15  Temp9am        366 non-null    float64
 16  Temp3pm        366 non-null    float64
 17  RainToday      366 non-null    int64  
dtypes: float64(9), int64(9)
memory usage: 51.6 KB
```

Comenzar a definir variables:

Ponemos un array para crear la variable clustering con el número de variables-1 y un random state de 5, luego utilizamos la función **.fix(var)**

In [5]:
```
X = lluvias_sydney.to_numpy()
clustering = KMeans(n_clusters=16, random_state = 5)
clustering.fit(X)
```

Out[5]:
```
KMeans(n_clusters=16, random_state=5)
```

Crear el gráfico:

Mediante numpy, creamos un array con los colores a utilizar en el gráfico, después definimos una nueva variable tipo DataFrame que obtenga los datos que ya tenemos, a continuación, definimos las columnas que tenemos disponibles y comenzamos a crear el gráfico con las funciones **.subplot(n,n,n)** y **.scatter(x, y, c, ...)**. Para finalizar, ahora ponemos un título y ya está.

Update: Por alguna razón no pude poner el color pero veo que se ponen unos de forma automática.

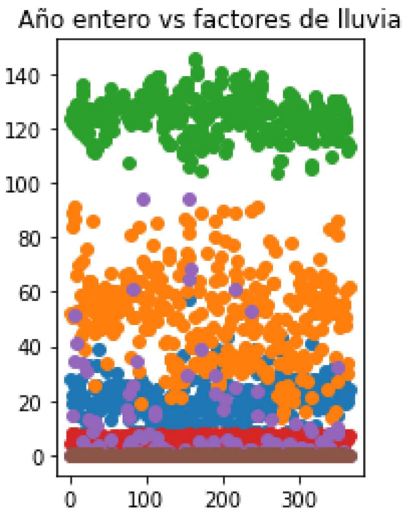
Update 2: Resté 890 en Pressure para que se viera mejor el gráfico.

In [6]:
```
color_theme = np.array(['darkgray', 'lightsalmon','powderblue'])
sydney_df = pd.DataFrame(lluvias_sydney)
sydney_df.columns=['LaFech', 'MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine

plt.subplot (1,2,2)
plt.scatter(sydney_df['LaFech'], sydney_df['WindSpeed3pm'])
plt.scatter(sydney_df['LaFech'], sydney_df['Humidity3pm'])
plt.scatter(sydney_df['LaFech'], sydney_df['Pressure3pm']-890) #Reducción para mejor vi
plt.scatter(sydney_df['LaFech'], sydney_df['Cloud3pm'])
plt.scatter(sydney_df['LaFech'], sydney_df['Rainfall'])
plt.scatter(sydney_df['LaFech'], sydney_df['RainToday'])
plt.title ("Año entero vs factores de lluvia")
```

Out[6]:

```
Text(0.5, 1.0, 'Año entero vs factores de lluvia')
```



Evaluar resultados:

Desafortunadamente no supe cómo realizar esta parte pero estuve intentando varias cosas para hacerlo funcionar, metí datos y me sacó listas interminables con advertencias, mejor lo dejé sin funcionar. 😟

In [7]:

```
# relabel = np.choose (clustering.labels_,[2,0,1]).astype(np.int64)
# print (classification_report(y, relabel))
```

- Pruebas -

Experimentos con las primeras variables recibidas (contiene errores)

In [8]:

```
iris=datasets.load_iris()
iris.data
```

Out[8]:

```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
[4.8, 3. , 1.4, 0.1],
[4.3, 3. , 1.1, 0.1],
[5.8, 4. , 1.2, 0.2],
[5.7, 4.4, 1.5, 0.4],
[5.4, 3.9, 1.3, 0.4],
[5.1, 3.5, 1.4, 0.3],
[5.7, 3.8, 1.7, 0.3],
[5.1, 3.8, 1.5, 0.3],
[5.4, 3.4, 1.7, 0.2],
[5.1, 3.7, 1.5, 0.4],
[4.6, 3.6, 1. , 0.2],
[5.1, 3.3, 1.7, 0.5],
[4.8, 3.4, 1.9, 0.2],
[5. , 3. , 1.6, 0.2],
[5. , 3.4, 1.6, 0.4],
[5.2, 3.5, 1.5, 0.2],
[5.2, 3.4, 1.4, 0.2],
[4.7, 3.2, 1.6, 0.2],
[4.8, 3.1, 1.6, 0.2],
[5.4, 3.4, 1.5, 0.4],
[5.2, 4.1, 1.5, 0.1],
[5.5, 4.2, 1.4, 0.2],
[4.9, 3.1, 1.5, 0.2],
[5. , 3.2, 1.2, 0.2],
[5.5, 3.5, 1.3, 0.2],
[4.9, 3.6, 1.4, 0.1],
[4.4, 3. , 1.3, 0.2],
[5.1, 3.4, 1.5, 0.2],
[5. , 3.5, 1.3, 0.3],
[4.5, 2.3, 1.3, 0.3],
[4.4, 3.2, 1.3, 0.2],
[5. , 3.5, 1.6, 0.6],
[5.1, 3.8, 1.9, 0.4],
[4.8, 3. , 1.4, 0.3],
[5.1, 3.8, 1.6, 0.2],
[4.6, 3.2, 1.4, 0.2],
[5.3, 3.7, 1.5, 0.2],
[5. , 3.3, 1.4, 0.2],
[7. , 3.2, 4.7, 1.4],
[6.4, 3.2, 4.5, 1.5],
[6.9, 3.1, 4.9, 1.5],
[5.5, 2.3, 4. , 1.3],
[6.5, 2.8, 4.6, 1.5],
[5.7, 2.8, 4.5, 1.3],
[6.3, 3.3, 4.7, 1.6],
[4.9, 2.4, 3.3, 1. ],
[6.6, 2.9, 4.6, 1.3],
[5.2, 2.7, 3.9, 1.4],
[5. , 2. , 3.5, 1. ],
[5.9, 3. , 4.2, 1.5],
[6. , 2.2, 4. , 1. ],
[6.1, 2.9, 4.7, 1.4],
[5.6, 2.9, 3.6, 1.3],
[6.7, 3.1, 4.4, 1.4],
[5.6, 3. , 4.5, 1.5],
[5.8, 2.7, 4.1, 1. ],
[6.2, 2.2, 4.5, 1.5],
[5.6, 2.5, 3.9, 1.1],
[5.9, 3.2, 4.8, 1.8],
[6.1, 2.8, 4. , 1.3],
[6.3, 2.5, 4.9, 1.5],
[6.1, 2.8, 4.7, 1.2],
[6.4, 2.9, 4.3, 1.3],
[6.6, 3. , 4.4, 1.4],
[6.8, 2.8, 4.8, 1.4],
       [6.7, 3. , 5. , 1.7],
       [6. , 2.9, 4.5, 1.5],
       [5.7, 2.6, 3.5, 1. ],
       [5.5, 2.4, 3.8, 1.1],
       [5.5, 2.4, 3.7, 1. ],
       [5.8, 2.7, 3.9, 1.2],
       [6. , 2.7, 5.1, 1.6],
       [5.4, 3. , 4.5, 1.5],
       [6. , 3.4, 4.5, 1.6],
       [6.7, 3.1, 4.7, 1.5],
       [6.3, 2.3, 4.4, 1.3],
       [5.6, 3. , 4.1, 1.3],
       [5.5, 2.5, 4. , 1.3],
       [5.5, 2.6, 4.4, 1.2],
       [6.1, 3. , 4.6, 1.4],
       [5.8, 2.6, 4. , 1.2],
       [5. , 2.3, 3.3, 1. ],
       [5.6, 2.7, 4.2, 1.3],
       [5.7, 3. , 4.2, 1.2],
       [5.7, 2.9, 4.2, 1.3],
       [6.2, 2.9, 4.3, 1.3],
       [5.1, 2.5, 3. , 1.1],
       [5.7, 2.8, 4.1, 1.3],
       [6.3, 3.3, 6. , 2.5],
       [5.8, 2.7, 5.1, 1.9],
       [7.1, 3. , 5.9, 2.1],
       [6.3, 2.9, 5.6, 1.8],
       [6.5, 3. , 5.8, 2.2],
       [7.6, 3. , 6.6, 2.1],
       [4.9, 2.5, 4.5, 1.7],
       [7.3, 2.9, 6.3, 1.8],
       [6.7, 2.5, 5.8, 1.8],
       [7.2, 3.6, 6.1, 2.5],
       [6.5, 3.2, 5.1, 2. ],
       [6.4, 2.7, 5.3, 1.9],
       [6.8, 3. , 5.5, 2.1],
       [5.7, 2.5, 5. , 2. ],
       [5.8, 2.8, 5.1, 2.4],
       [6.4, 3.2, 5.3, 2.3],
       [6.5, 3. , 5.5, 1.8],
       [7.7, 3.8, 6.7, 2.2],
       [7.7, 2.6, 6.9, 2.3],
       [6. , 2.2, 5. , 1.5],
       [6.9, 3.2, 5.7, 2.3],
       [5.6, 2.8, 4.9, 2. ],
       [7.7, 2.8, 6.7, 2. ],
       [6.3, 2.7, 4.9, 1.8],
       [6.7, 3.3, 5.7, 2.1],
       [7.2, 3.2, 6. , 1.8],
       [6.2, 2.8, 4.8, 1.8],
       [6.1, 3. , 4.9, 1.8],
       [6.4, 2.8, 5.6, 2.1],
       [7.2, 3. , 5.8, 1.6],
       [7.4, 2.8, 6.1, 1.9],
       [7.9, 3.8, 6.4, 2. ],
       [6.4, 2.8, 5.6, 2.2],
       [6.3, 2.8, 5.1, 1.5],
       [6.1, 2.6, 5.6, 1.4],
       [7.7, 3. , 6.1, 2.3],
       [6.3, 3.4, 5.6, 2.4],
       [6.4, 3.1, 5.5, 1.8],
       [6. , 3. , 4.8, 1.8],
       [6.9, 3.1, 5.4, 2.1],
       [6.7, 3.1, 5.6, 2.4],
       [6.9, 3.1, 5.1, 2.3],
       [5.8, 2.7, 5.1, 1.9],
       [6.8, 3.2, 5.9, 2.3],
       [6.7, 3.3, 5.7, 2.5],
       [6.7, 3. , 5.2, 2.3],
       [6.3, 2.5, 5. , 1.9],
       [6.5, 3. , 5.2, 2. ],
       [6.2, 3.4, 5.4, 2.3],
       [5.9, 3. , 5.1, 1.8]])
```

In [9]:

```
iris.target
```

Out[9]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

In [10]:

```
X
```

Out[10]:

```
array([[  1. ,  17.9,  25.6, ...,  22.9,  24.2,   0. ],
       [  2. ,  18. ,  25.4, ...,  22.3,  23.5,   0. ],
       [  3. ,  20.2,  24.6, ...,  22.2,  19.9,   0. ],
       ...,
       [364. ,  22.6,  36.6, ...,  28.1,  31.8,   0. ],
       [365. ,  23.9,  33.3, ...,  27.3,  32.1,   0. ],
       [366. ,  24.1,  30. , ...,  27.7,  26.4,   0. ]])
```

In [11]:

```
# lluvias_sydney.data()
```

In [12]:

```
sydney_df['LaFech']
```

Out[12]:

```
0        1
1        2
2        3
3        4
4        5
      ...
361    362
362    363
363    364
364    365
365    366
Name: LaFech, Length: 366, dtype: int64
```

In [13]:

```
sydney_df['RainToday']
```

Out[13]:

```
0      0
1      0
2      0
3      1
4      1
      ..
361    0
362    0
363    0
364    0
365    0
Name: RainToday, Length: 366, dtype: int64
```