Programación para Data Science

Unidad 7: Análisis de datos en Python - Ejercicios y preguntas

Ejercicio 0

Cargad el conjunto de datos Iris incorporado en la librería `sklearn`.

In [1]:

```
# Respuesta
```

Ejercicio 1

Implementad una función, `describe_iris`, que devuelva un diccionario con la siguiente estructura:

```
{
   "categorias": [],
   "atributos": [],
   "num_muestras": 0
}
```

categorias debe ser un array con el nombre de los **targets** del dataset. **atributos** debe ser un array con el nombre de los **atributos** y finalmente, **num_muestras** debe indicar el número **total de muestras** del dataset. **(0.5 puntos)**

In [50]:

```
# Respuesta

def describe_iris():
    # Código a completar
    return
```

Ejercicio 2

Representad gráficamente en un **scatter plot** la longitud de los sépalos frente a la longitud de los pétalos. **(1.5 puntos)**

Nota: para poder incluir acentos en los textos de las etiquetas o del título del **plot**, es necesario indicar explícitamente que las cadenas de caracteres son **unicode**. Podéis hacerlo incluyendo una `u` delante de las comillas que delimitan la cadena de caracteres.

In [3]:

```
# Ejemplo de cadena de caracateres unicode especificada explícitamente.
print u"pétalo"
```

```
pétalo
```

In [51]:

```
# Respuesta
```

Pregunta 1

En el Notebook de explicación hemos utilizado un clasificador **k nearest neighbors**. Describid a grandes rasgos cómo funciona este clasificador. **(1 punto)**

Respuesta:

Ejercicio 3

Aplicad el clasificador **KNeighborsClassifier** para predecir el tipo de especie de iris utilizando la longitud y ancho de los pétalos como atributos y utilizando 20 muestras de aprendizaje y 20 muestras de test (podéis usar cualquier partición de muestras de aprendizaje y de test). **(1 punto)**

In [52]:

```
# Respuesta
```

Ejercicio 4

Visualizad gráficamente el clasificador aprendido en el ejercicio anterior mostrando tanto las muestras usadas para el aprendizaje como las muestras utilizadas para el test. Utilizad colores para mostrar la clase (`target`) a la que pertenecen las muestras de aprendizaje y las de test. **(1 punto)**

Pista: podéis utilizar el código que hemos visto en el Notebook de explicación, añadiendo una línea que permita visualizar las muestras de test y cambiando el marcador para diferenciarlas de las muestras de aprendizaje.

In [53]:

```
# Respuesta
```

Pregunta 2

Comparad la precisión del nuevo clasificador aprendido en el ejercicio 3 con la precisión del clasificador del Notebook de explicación. ¿Podemos observar alguna diferencia? En caso afirmativo, explicad brevemente por qué. **(1 punto)**

Respuesta:

Ejercicio 5

Implementad una función, `kfold_bins` , que nos devuelva las longitudes de los conjuntos de aprendizaje y test aplicando la función **KFold** a los datos del dataset Iris utilizando **K = 4**. El formato de salida de la función debería ser:

```
[ (longitud_apredizaje1, longitud_test1), ..., (longitud_apredizajeK, longitud_testK) ]
```

(1 punto)

Respuesta:

In [54]:

```
# Respuesta

def kfold_bins(k = 1):
    # Código a completar
    return
```

Ejercicio 6

Aplicad el algoritmo de **clustering KMeans** tal como hemos visto en el Notebook de explicación, pero esta vez utilizando los siguientes parámetros:

```
Número de clusters: 3
Método de inicialización de los puntos centrales: 'random'
Número de iteraciones para la selección de puntos centrales: 1
```

Visualizad gráficamente el resulado. **(2 puntos)**

In [55]:

```
# Respuesta
```

Pregunta 3

Ejecutad 10 veces la celda anterior y observad el resultado final. ¿Obtenemos siempre los mismos clusters? Explicad brevemente como pueden afectar los dos nuevos parámetros introducidos en el ejercicio anterior al resultado final. **(1 punto)**

Respuesta:

Ejercicio Opcional

Usad el clasificador **nearest centroid** https://en.wikipedia.org/wiki/Nearest_centroid_classifier (https://en.wikipedia.org/wiki/Nearest_centroid_classifier) para predecir el tipo de especies de iris. Visualizad de forma gráfica el clasificador aprendido.

Pista. Podéis utilizar la librería **sklearn**.