Programación para *Data Science*

Unidad 7: Análisis de datos en Python

Introducción

En este Notebook encontraréis dos conjuntos de ejercicios: un primer conjunto de **ejercicios para practicar** y un segundo conjunto de **actividades evaluables** como PEC de la asignatura.

En cuanto al conjunto de ejercicios para practicar, éstos no puntúan para la PEC, pero os recomendamos que los intentéis resolver como parte del proceso de aprendizaje. Encontraréis ejemplos de posibles soluciones a los ejercicios al propio notebook, pero es importante que intentéis resolverlos vosotros antes de consultar las soluciones. Las soluciones os permitirán validar vuestras respuestas, así como ver alternativas de resolución de las actividades. También os animamos a preguntar cualquier duda que surja sobre la resolución de los **ejercicios para practicar** en el foro del aula.

En relación a las actividades evaluables, veréis que cada una de ellas tiene asociada una puntuación que indica el peso que tiene la actividad sobre la nota de la PEC. Adicionalmente, hay un ejercicio opcional, que no tiene puntuación dentro de la PEC, pero que se valora al final del semestre de cara a conceder las matrículas de honor y redondear las notas finales. Podéis sacar la máxima nota de la PEC sin necesidad de hacer este ejercicio! El objetivo de este ejercicio es que sirva como pequeño reto para los estudiantes que quieran profundizar en el contenido de la asignatura.

Además, veréis que todas las actividades tienen una etiqueta que indica los recursos necesarios para llevarla a cabo. Hay tres posibles etiquetas:

- `NM` **Sólo materiales**: las herramientas necesarias para realizar la actividad se pueden encontrar en los materiales de la asignatura.
- `EG` **Consulta externa guiada**: la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, pero el enunciado contiene indicaciones de dónde o cómo encontrar la información adicional necesaria para resolver la actividad.
- `EI` **Consulta externa independente**: la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, y el enunciado puede no incluir la descripción de dónde o cómo encontrar esta información adicional. Será necesario que el estudiante busque esta información utilizando los recursos que se han explicado en la asignatura.

Es importante notar que estas etiquetas no indican el nivel de dificultad del ejercicio, sino únicamente la necesidad de consulta de documentación externa para su resolución. Además, recordad que las **etiquetas son informativas**, pero podréis consultar referencias externas en cualquier momento (aunque no se indique explícitamente) o puede ser que podáis hacer una actividad sin consultar ningún tipo de documentación. Por ejemplo, para resolver una actividad que sólo requiera los materiales de la asignatura, puedéis consultar referencias externas si queréis, ya sea tanto para ayudaros en la resolución como para ampliar el conocimiento!

En cuanto a la consulta de documentación externa en la resolución de los ejercicios, recordad **citar siempre la bibliografía utilizada** para resolver cada actividad.

Ejercicios para practicar

Ejercicio 1

Carga el conjunto de datos breast_cancer incorporado en la librería `sklearn`. Implementa una función, `descriu_diagnostics`, 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.

In [16]:

```
# Respuesta
```

Ejercicio 2

Representa gráficamente en un scatter plot el area de los tumores respecto a su suavidad (smoothness).

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. Podréis hacerlo incluyendo una *u* ante las comillas que delimitan la cadena de caracteres.

In [1]:

```
# Respuesta
```

Ejercicio 3

Divide los datos breast_cancer en dos subconjuntos, datos de entrenamiento y test, en una proporción 80% entrenamiento y 20% test.

In [2]:

```
# Respuesta
```

Ejercicios y preguntas teóricas para la PEC

A continuación, los ejercicios que se deben completar en esta PEC y que forman parte de la evaluación de esta unidad.

Pregunta 1

Investiga y describe el funcionamiento del algoritmo de clustering *Leave One Out*. Pon un ejemplo de aplicación del algoritmo para resolver un problema de análisis en el ámbito de la salud. Recordad que hay que citar las referencias consultadas para responder la pregunta, y que la respuesta que proporcionéis debe ser original (redactada por vosotros mismos, después de haber leído y entendido las referencias que consideréis oportunas). **(1 punto)** `EG`

Respuesta

Pregunta 2

Debemos evitar evaluar los modelos con los mismos datos que se han utilizado para el aprendizaje. De no hacerlo, podríamos favorecer el problema del **sobre-ajuste**. ¿En qué consiste este problema y qué consecuencias puede tener? **(0,5 puntos)** `EI`

Respuesta

Ejercicio 1

En los siguientes ejercicios de la PEC usaremos el conjunto de datos del dataset `breast_cancer` de *sklearn*. Crea una tabla con los estadísticos descriptivos de todas las características de los tumores de la base de datos `breast_cancer` agrupados por tipo (benigno/maligno).

Realizad un test estadístico ttest para comparar el error de la simetría (*symmetry error*) y tersura (*mean smoothness*) de los tumores benignos con las de los malignos. Dado el resultado obtenido, indica qué características podrían ser más relevantes para diagnosticar un tumor maligno.

Nota: Consulta este enlace si quieres ampliar conocimientos sobre el test estadístico ttest. Para realizar el test estadístico

automáticamente podéis utilizar el método `ttest_ind` de *Scipy.stats* **(1 punto)** EG

In [22]:

```
# Respuesta
```

Respuesta

Ejercicio 2

Representar mediante [boxplots] (https://en.wikipedia.org/wiki/Boxplots) como varían la textura, el area, la concavidad y la simetría de los tumores según el tipo (*maligna* / *benigna*). Recuerda ajustar los parámetros de la visualización para facilitar la lectura e interpretación de la gráfica que generen. **(2 puntos)** NM

Nota: Crea un dataframe con los datos de `diagnostics.data` i `diagnostics.target` para dibujar los boxplots.

In [1]:

```
# Importamos las librerías

import numpy as np
import pandas as pd
import seaborn as sns  # librería opcional
import matplotlib.pyplot as plt
from sklearn import datasets

# Importamos el dataset
diagnostics = datasets.load_breast_cancer()
```

In [7]:

```
## Respuesta
```

Ejercicio 3

Ahora aplica un Principal Component Analysis para reducir el número de variables predictivas de los tipos de tumor. NM

a) ¿Qué porción de la variabilidad total la explica el primer componente? **(1.5 puntos)**

b) Representa en un diagrama de barras qué porción de la variabilidad total explica cada uno de los tres primeros componentes. **(0.5 puntos)**

In [5]:

```
## Respuesta
```

Ejercicio 4

El objetivo de un modelo de **regresión lineal** es encontrar una relación entre una o más características (variables independientes) y una variable objetivo continua (variable dependiente). Cuando sólo usamos una característica predictiva se le llama **Regresión Lineal Univariada** y si hay múltiples predictores se llama **Regresión Lineal Múltiple**.

Cuando la variable dependiente es una variable binaria que contiene datos codificados como 1 (sí, éxito, etc.) o 0 (no, fallo, etc.), como es el caso del dataset `breast_cancer`, tendremos que emplear una regresión logística.

Crea un modelo de regresión logística que estime la probabilidad de que un tumor sea maligno dada su área y su textura. Muestra la matriz de confusión del modelo obtenido.

Nota: Quizás te interesa explorar la función sklearn.metrics.confusion_matrix() **(1,5 puntos)** EI

In [8]:

```
# Respuesta
```

Ejercicio 5

El **Multi-layer Perceptron (MLP)** es un algoritmo de aprendizaje supervisado que aprende una función mediante el entrenamiento de un modelo que asocia n dimensiones de entrada (*predictores*) con dimensiones de salida (*targets*). Teniendo en cuenta un conjunto de funciones y un objetivo, puede aprender un aproximador de funciones no lineales por clasificación o regresión. Es diferente de la regresión logística, ya que entre la capa de entrada y la de salida, puede haber una o más capas no lineales, llamadas capas ocultas.

Aplica un clasificador basado en un MLP para predecir los tipos de tumor utilizando el área, la textura, la simetría y la concavidad, como atributos y utilizando el 70% de las muestras de entrenamiento y el 30% de test. Debéis utilizar la función sklearn.neural_network.MLPClassifier.

Utiliza tres niveles (*layers*) con 30 neuronas por nivel. ¿Qué valor de precisión obtenemos en un modelo basado en un MLP? **(2 puntos)** EG

In [9]:

```
# Respuesta
```

Ejercicio Opcional

Aplicad un clasificador basado en un árbol de decicisión de un máximo de 3 niveles de profundidad para predecir los tipos de tumor utilizando el area, la tersura y la simetría como atributos y utilizando 60% de las muestras de entrenamiento y el 40% de test. Debéis utilizar la función sklearn.tree.DecisionTreeClassifier.

¿Qué valor de precisión obtenemos en un modelo basado en un árbol de decisión? Representa el árbol de decisión y expórtalo a un archivo PDF. Explora el árbol resultante al PDF y responde a la siguiente pregunta: Dados estos datos, cómo diagnosticarías a un paciente que presenta un tumor de area 400, textura 12.18, y un valor de lisura de 19 puntos?

Nota: Tal vez la función tree de *sklearn* sea de utilidad. EG

In [2]:

```
# Respuesta
```

Respuesta

Soluciones a los ejercicios para practicar

Exercici 1

Carga el conjunto de datos breast_cancer incorporado en la librería `sklearn`. Implementa una función, `descriu_diagnostics`, 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.

In [3]:

```
from sklearn import datasets

# Carga el dataset breast_cancer
```

```
diagnostics = datasets.load_breast_cancer()

def describe_diagnostics():
    #Inicialitzem el diccionari
    diccionario = {}
    #Assignem a cada atribut el valor corresponent segons al dataset breast_cancer
    diccionario["categorias"] = diagnostics.target_names
    diccionario["atributos"] = diagnostics.feature_names
    diccionario["num_muestras"] = len(diagnostics.data)

    return diccionario

print(describe_diagnostics())
```

```
{'categorias': array(['malignant', 'benign'], dtype='<U9'), 'atributos': array(['mean radius',
'mean texture', 'mean perimeter', 'mean area',
       'mean smoothness', 'mean compactness', 'mean concavity',
       'mean concave points', 'mean symmetry', 'mean fractal dimension',
       'radius error', 'texture error', 'perimeter error', 'area error',
       'smoothness error', 'compactness error', 'concavity error',
       'concave points error', 'symmetry error',
       'fractal dimension error', 'worst radius', 'worst texture',
       'worst perimeter', 'worst area', 'worst smoothness',
       'worst compactness', 'worst concavity', 'worst concave points',
       'worst symmetry', 'worst fractal dimension'], dtype='<U23'), 'num_muestras': 569}
```

Exercici 2

Representa gráficamente en un scatter plot el area de los tumores respecto a su suavidad (smoothness).

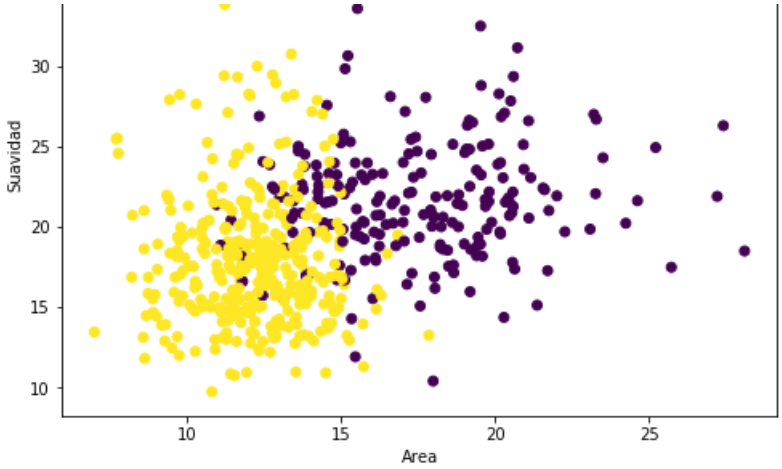
Nota: para poder incluir acentos en los textos de las etiquetas o del título del plot, es necesario indicar explícitamente que las cadenas de caracteres son unicode. Puede hacerlo incluyendo una *u* ante las comillas que delimitan la cadena de caracteres.

In [14]:

```
%matplotlib inline

#Importamos las librerías

import matplotlib.pyplot as plt
from sklearn import datasets

# Importamos el dataset

diagnostics=datasets.load_breast_cancer()

Area = diagnostics.data[:,0]

Smoothness = diagnostics.data[:,1]

Y = diagnostics.target

# Creamos la figura

plt.figure(1, figsize=(8,6))

plt.clf()

# Asignamos colores según el tipo

plt.scatter(Area,Smoothness,c=Y)
plt.xlabel(u'Area')
plt.ylabel(u'Suavidad')
```

Out[14]:

```
Text(0, 0.5, 'Suavidad')
```



Ejercicio 3

Divide los datos breast_cancer en dos subconjuntos, datos de entrenamiento y test, en una proporción 80% entrenamiento y 20% test.

In [15]:

```
from sklearn.model_selection import train_test_split

diagnostics=datasets.load_breast_cancer()

# Dividimos los datos:  entrenamiento(80%) y test (20%) mediante la función train_test_split
train,test = train_test_split(diagnostics.data, test_size=0.2)

# El atributo test_size=0.2 divide el dataset en entrenamiento=80% y test=20%
print(train.shape)
print(test.shape)
```

```
(455, 30)
(114, 30)
```

In [ ]:

```

```