Programación para *Data Science*

PEC 7 - Introducción al análisis de datos en Python

En este Notebook encontraréis el conjunto de actividades evaluables como PEC de la asignatura. Veréis que cada una de ellas tiene asociada una puntuación, que indica el peso que tiene la actividad sobre la nota final 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. Podréis sacar la máxima nota de la PAC 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.

Veréis que todas las actividades de la PEC 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 independiente**: 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.

Ejercicio 1

Estudia el siguiente código y descríbe paso a paso su funcionamiento. **(0.5 puntos)** `NM`

```
import numpy as np
data = np.vstack([ages, weight, children, gross, properties ]).T

data_normalised = (data - data.mean(axis=0) )/ data.std(axis=0)

pca = PCA(n_components=2)
pca.fit(data_normalised)
print(pca.explained_variance*ratio*)
print(pca.singular*values*)

transformed_data = pca.transform(data_normalised)

plt.scatter( transformed_data[:,0], transformed_data[:,1], c=corporative_job, cmap='jet')
plt.xlabel('PCA 1')
plt.ylabel('PCA 2')
```

Respuesta

Ejercicio 2

Carga el conjunto de datos diabetes incorporado en la librería `sklearn`.

En este conjunto de datos, hay registros de pacientes con diferentes características que se relacionan con la diabetes, así como el diagnóstico de cada uno. La variable objetivo `target` es una medida cuantitativa de la progresión de la enfermedad un año después del inicio.

1.1) Crea un dataframe, incluyendo una columna con el nombre *progresion*, que indique la progressión de la enfermedad del paciente, es decir la variable `target`. El resto de columnas corresponderán con las características o `feature_names`.

1.2) Crea una tabla con los estadísticos descriptivos de todas las características de los pacientes de la base de datos `diabetes` agrupados por tipo: deterioro severo (progressión de la diabetes por encima de la mediana del conjunto), deterioro moderado (progressión de la diabetes por debajo o igual a la mediana del conjunto). Los estadísticos descriptivos deben incluir la media, la desviación típica, el mínimo, el máximo, la mediana, el percentil 25 y el percentil 75.

1.3) Una vez hecho esto, tendréis que representar gráficamente los datos de 'edad', y 'bmi', por tipo de deterioro (severo o moderado) utilizando un diagrama de barras que represente la media y una barra de error correspondiente con el error estándar.

1.4.) Explora la documentación del dataset Diabetes y explica por qué crees que los valores de edad son tan pequeños.

(2 puntos) NM

In [ ]:

```
# Respuesta
```

Ejercicio 3

Ahora que tenéis el conjunto de datos organizados, trazad la distribución de las características "edad", "sexo" y "bmi". Para ello podéis utilizar histogramas como vimos en el Notebook de teoría o podéis elegir utilizar un diagrama de violín de la librería Matplotlib. Aseguraos de mostrar una sola figura con múltiples paneles, dedicando un panel a cada característica de los pacientes (ej. edad). **(1 punto)** NM

In [ ]:

```
# Respuesta
```

Ejercicio 4

Dado este conjunto de datos, en un hospital nos piden explorar qué tres características podrían ser los factores más relevantes para la mala recuperación de pacientes con diabetes.

1) La correlación nos permite estudiar la relación entre dos variables y Pandas nos ofrece una función para calcular correlaciones de forma sencilla, se trata del método corr(). Muestra el resultado en una matriz de correlaciones o correlograma que represente gráficamente la correlación de las características de los pacientes de diabetes entre ellas, así como su correlación con la variable target. Indica cuáles son las tres características más relevantes para la progresión de la enfermedad y justifica tu respuesta. **(1 punto)**

Nota: Te puede ser útil la función Pcolor de Matplotlib. Si representas el correlograma con una matriz de colores, no olvides mostrar la barra de color.

2) ¿Sería posible utilizar una técnica de análisis de datos **no supervisada** para resolver este problema? ¿Por qué? **(0.5 puntos)**

EG

In [ ]:

```
# Respuesta
```

Respuesta

Ejercicio 5

En los ejercicios resueltos hemos visto como usar la regresión logística, y hemos hablado de **la regresión lineal múltiple**.

En este ejercicio, tendréis que aplicar la regressión lineal múltiple para construir un modelo predictivo de deterioro en pacientes de

En este ejercicio, tendréis que aplicar la regresión lineal multiple para construir un modelo predictivo de deterioro en pacientes de diabetes según las características del conjunto de datos.

- En primer lugar, tendréis que dividir los datos en conjuntos de entrenamiento (*training*) y evaluación (*test*), con un tamaño de la muestra para test del 33%.
- A continuación, entrenad el modelo con la porción de datos para el entrenamiento y calculad la precisión del modelo. Escribid una breve explicación que indique lo que significa obtener tal precisión.

(2 puntos) `EG`

In [ ]:

```
# Respuesta
```

Respuesta

Ejercicio 6

Realizad un test estadístico ttest para responder a la siguiente pregunta: ¿Los hombres diagnosticados con diabetes tienen más tendencia al deterioro que las mujeres?
Hipótesis Nula: Los hombres diagnosticados con diabetes no tienen una tendencia al deterioro diferente de la que muestran las mujeres.
Hipótesis Alternativa: Los hombres diagnosticados con diabetes tienen una tendencia diferente al deterioro que las mujeres.
Una vez realizado el test, interpreta los valores resultantes y elabora una respuesta. **(1 punto)** `EG`

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*

In [ ]:

```
# Respuesta
```

Respuesta

Ejercicio 7

Aplicad un clasificador basado en un árbol de decicisión de un máximo de 3 niveles de profundidad para predecir la progression de la diabetes (severa o moderada segun la categorías realizadas en el ejercicio 5) utilizando el sexo, la edad y el bmi 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.

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

(2 puntos) `EI`

In [ ]:

```
# Respuesta
```

Ejercicio Opcional

En este ejercicio, asumiremos que sólo podemos usar 2 de las seguientes variables para predecir el deterioro del paciente: 'age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6'.

Aplica un classificador *K-Nearest Neighbors* para predecir el tipo de paciente según su deterioro (pregresión severa o moderada) dadas solo 2 variables. Para encontrar que par de variables es más relevante para conseguir un mejor rendimiento del modelo, tendrás que evaluar tu clasificador para cada par de varibles.

`NM`

In [ ]:

```
# Respuesta
```

```
# Respuesta
```