prog_datasci_7_entrega

December 18, 2022

1 Fundamentos de Programación

1.1 PEC 7: Introducción al análisis de datos en Python

En este Notebook se 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 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.

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.

1.2 Ejercicios para la PEC

En esta PEC trabajaremos con un conjunto de datos (***dataset***) del sistema eléctrico español. Salvando algún ligero ajuste que hemos hecho para facilitaros el análisis, todos los datos estan disponibles en abierto en esta página web de Red Eléctrica de España (REE). Nuestro *dataset* contiene información horaria del precio medio de la electricidad (€/MWh), del consumo peninsular (MWh), del saldo entre interconexiones (MWh), de la generación de distintas tecnologías renovables y no renovables (MWh) así como el índice semanal de llenado de los embalses (MWh). Esperamos que disfrutéis de esta PEC.

1.2.1 Ejercicio 1

(a) Carga el *dataset* e imprime por pantalla sus dimensiones, así como el nombre de las variables, su tipo y el número de valores no perdidos que contienen. NM **(0.5 puntos)**

[ ]:
```
# Respuesta
```

(b) Cambia el nombre de la primera columna a "fecha", asígnala como índice del *dataset* y modifica el tipo de variable a 'datetime64[ns]'. Imprime por pantalla las 2 primeras filas y las 2 últimas. NM **(0.5 puntos)**

[ ]:
```
# Respuesta
```

1.2.2 Ejercicio 2

(a) Calcula los valores estadísticos típicos del dataframe (min, max, media, etc.) e imprime por pantalla el nombre de la columna correspondiente a "Generación programada PBF" cuyo valor mínimo sea más alto. NM **(0.5 puntos)**

[ ]:
```
# Respuesta
```

(b) Vamos a analizar como se comporta la demanda peninsular según la hora del día. Para ello, traza un `boxplot` de la `Demanda programada PBF total` cuyo eje horizontal se corresponda con las 24 horas del día. Te puede ser útil extraer la hora del índice `fecha` creado en el ejercicio 1. EG

Nota: Para más información sobre la función boxplot podéis consultar este enlace.

(1.0 punto)

[ ]:
```
# Respuesta
```

1.2.3 Ejercicio 3

Continuamos con el análisis de la demanda. Ahora haremos uso de técnicas de *machine learning* no supervisadas para buscar patrones de consumo. En concreto, intentaremos buscar **dos clusters**, uno característico del patrón de consumo en días laborables y otro, en días no laborables.

(a) A partir del dataframe resultante del ejercicio 1, extrae la serie correspondiente a la demanda del 2021 y modifica sus dimensiones según la tupla (365, 24). El objetivo es crear un `array` de `numpy` cuyas filas sean los días del año y cuyas columnas, las horas de cada día. NM **(0.5 puntos)**

```
[ ]: # Respuesta
```

(b) Crea un modelo `KMeans`con **dos** clusters usando el algoritmo 'elkan' y traza la gráfica de los centroides de cada *cluster* en función de las horas del día. EG **(1.0 punto)**

Nota: En el enlance siguiente encontarás más información sobre la función KMeans.

```
[ ]: # Respuesta
```

(c) Utiliza la función imshow de matplotlib para visualizar los clusters anteriores sobre una matriz cuyas filas representan las semanas del año y cuyas columnas, los días de la semana. Te puede ser útil eliminar los 3 primeros y 5 últimos días del año para cuadrar semanas de lunes a domingo en cada columna. NM **(opcional)**

```
[ ]: # Respuesta
```

1.2.4 Ejercicio 4

En los siguientes ejercicios nos vamos a focalizar más en la variable "Precio mercado SPOT Diario". Calcula la matriz de correlación del conjunto de datos del ejercicio 1 y visualízala gráficamente. ¿Qué dos variables muestran mayor correlación con el precio? NM **(1.0 punto)**

```
[ ]: # Respuesta
```

1.2.5 Ejercicio 5

Hemos visto que no todas las variables del *dataset* están igual de correlacionadas con la variable precio. Esto nos hace preguntarnos, si queremos predecir el precio del mercado diario a partir de las otras variables, es necesario incluirlas todas?

Cuando se construyen modelos, a veces tener muchas variables puede llevar a problemas de dimensionalidad, por lo tanto, es útil comprobar si se puede reducir este número sin perder información relevante.

En este ejercicio realizaremos un **análisis de componentes principales (PCA)** para saber si podemos explicar el precio del mercado diario con un número reducido de variables (*features*).

(a) A partir del conjunto de datos del ejercicio anterior, crea un DataFrame **sin** la variable `Precio mercado SPOT Diario` y normalízalo haciendo uso de `StandardScaler`. NM **(0.5 puntos)**

```
[ ]: # Respuesta
```

(b) Con el *dataset* normalizado, crea un modelo de la clase `PCA` sin especificar el número de componentes a retener y verifica que el número de componentes finales coincide con el de columnas del *dataset* de entrenamiento. NM **(0.5 puntos)**

```
[ ]: # Respuesta
```

(c) ¿Cuántas componentes son necesarias para explicar como mínimo el 80% de la varianza del conjunto de datos de entrenamiento? EI **(1.0 punto)**

```
[ ]: # Respuesta
```

1.2.6 Ejercicio 6

En este ejercicio vamos a trabajar con técnicas supervisadas de *machine learning* para intentar predecir numéricamente el precio del mercado SPOT diario con un modelo de regresión lineal `LinearRegression`.

(a) A partir del *dataset* del ejercicio 1, utiliza la función `train_test_split` para seleccionar el conjunto de datos que servirán para la fase de entrenamiento (***training***) y para la fase de evaluación (***test***). Selecciona el 20% del total de muestras para el *test*. EG **(0.5 puntos)**

```
[ ]: # Respuesta
```

(b) Crea el modelo de `LinearRegression` y entrénalo. Al tratarse de un modelo de regresión, muestra por pantalla el coeficiente de determinación (**R2**) obtenido cuando se aplica el modelo sobre el conjunto de *test*. Recuerda que el mismo modelo dispone de la función `score` para el cálculo de R2. EG **(0.5 puntos)**

```
[ ]: # Respuesta
```

(c) Repite el modelo de regresión linear de los apartados anteriores, pero esta vez utilizando datos normalizados reducidos con el modelo PCA. Se obtienen resultados parecidos? **Opcional**

Nota: Usa como número de componentes el resultado del apartado 5c) (80% de la varianza explicada).

```
[ ]: # Respuesta
```

1.2.7 Ejercicio 7

En este ejercicio seguiremos con el ***dataset* del ejercicio 1** para trabajar con las técnicas de *machine learning* supervisadas pero enfocadas, esta vez, a la clasificación. Para ello, primero de todo, debemos definir al menos dos clases de precios.

(a) Añade una columna al *dataset* que sea 1 si el `Precio mercado SPOT Diario` es superior o igual a su percentil 50% y 0 si es inferior. Verifica por pantalla que, efectivamente, la columna nueva suele contiene dos valores. NM **(0.5 puntos)**

[ ]:
```
# Respuesta
```

(b) Como en el ejercicio anterior, utiliza la función `train_test_split` para seleccionar el conjunto de datos que servirán para la fase de entrenamiento (***training***) y para la fase de evaluación (***test***). Pero esta vez, utiliza los datos normalizados del ejercicio 5a como input y selecciona el 30% del total de muestras para el *test*. A continuación, crea el modelo de `LogisticRegression` y entrénalo. Muestra por pantalla el nivel de exactitud (*accuracy*) obtenida cuando se aplica el modelo sobre el conjunto de *test*. Recuerda que el mismo modelo dispone de la función `score` para el cálculo de la exactitud. EG **(0.5 puntos)**

[ ]:
```
# Respuesta
```

(c) Fíjate que el *score* cambia cada vez que ejecutas el código del apartado anterior. Y es normal porque las muestras escogidas en el conjunto de datos de *training* y de *test* no son siempre las mismas. Haz una **validación cruzada** del modelo con la función `cross_val_score` siguiendo un método *k-Fold* (`KFold`) con k=5 y `shuffle=True`. Muestra por pantalla la media de los scores obtenidos y su desviación típica. EG **(1.0 punto)**

[ ]:
```
# Respuesta
```