prop_prog_datasci_6_preproc_entrega

December 11, 2022

1 Fundamentos de Programación

1.1 PEC 6: Preprocesamiento 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

Tenemos un compañero que se quiere comprar un coche eléctrico, y queremos proporcionarle una lista fiable de estaciones dónde puede reservar un punto donde cargar su vehículo en caso de necesidad, así como unas sugerencias de compra basadas en criterios objetivos.

1.2.1 Ejercicio 1

En el primer ejercicio trabajaremos con un conjunto de datos (***dataset***) de los puntos de carga para vehículos eléctricos de la ciudad de Barcelona. En el archivo "EE_points.json" tenemos información sobre dichos puntos, pero queremos revisar previamente la información del mismo, y generar un pequeño resumen con la información que nos interesa más.

(a) Importa el archivo `EE_points.json` de la carpeta de datos en un dataframe. Examina el conjunto de datos y muestra por pantalla el nombre de las columnas, el contenido de las 7 primeras filas y el número de filas. **(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

(b) No todos los puntos de carga son validos para todos los tipos de vehículos. Muestra los diferentes valores de la columna "Vehicle_type"? **(0.5 puntos)** EG

[ ]:
```
# Respuesta
```

(c) Vistos los valores previos, parece que los datos se introdujeron en el fichero con algunos "fallos", asegúrate de que corregimos correctamente aquellos que no tienen un valor númerico en el campo "Vehicle_type" (sin modificar el fichero de entrada). Vuelve a comprovar los valores únicos de dicha columna tras la modificación **(1.0 puntos)** EG

[ ]:
```
# Respuesta
```

(d) Muestra el tipo de variable de las columnas del dataset y modifica el tipo de la columna "Vehicle_type" para que aparezca como integer... comprueba tras el cambio que se ha modificado como esperamos**(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

(e) Crea un nuevo dataframe partiendo del corregido anteriormente que contenga solo la "Station_address","Station_lat" y "Station_lng", donde "Reservable_station" sea "True", y donde el "Vehicle Type" sea 1. Asegurate de que no repetimos la misma dirección dos veces eliminando los duplicados, y que reseteamos la columna índice del dataframe. Para acabar muestra el número de estaciones, o filas del dataframe resultante, así como las primeras 5 filas. **(1 punto)** NM

[ ]:
```
# Respuesta
```

f) Exporta el dataframe resultante al fichero "output.csv" utilizando ";" como separador. **(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

1.2.2 Ejercicio 2

Tras proporcionar a nuestro compañero la lista de puntos de carga, parece que ha quedado convencido en comprar el vehículo pero duda que modelo es el que le interesa más. En la web "Kaggle.com" hemos encontrado una lista que creemos que le puede ser útil.

(a) Como en el ejercicio anterior, importa el archivo `ElectricCar.csv` de la carpeta de datos, en un dataframe. Examina el conjunto de datos y muestra por pantalla el nombre de las columnas, el contenido de las 7 primeras filas y el número de filas. **(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

(b) No estamos interesados en los vehículos que no tienen disponible el precio (valores perdidos). Elimínalos del dataframe y vuelve a contar el número de filas**(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

(c) Para hacer un análisi rápido de las diferentes marcas, y precios de sus vehículos, queremos obtener los nombres de los diferentes fabricantes "Brands" y el precio medio de sus vehículos. Utilizando "groupby" crea un nuevo dataframe que tenga esta información y muéstrala. Que fabricante es el que nos ofrece, en media, modelos más económicos? **(1 punto)** NM

[ ]:
```
# Respuesta
```

(d) Muestra los modelos y precio de los vehículos de la marca "SEAT". **(1 punto)** NM

NOTA1: si te encuentras con algun problema... quizás es que hay un espacio detrás del nombre del fabricante, elimina estos espacios de la columna "Brand" antes de nada.

[ ]:
```
# Respuesta
```

e) En este apartado vamos a generar 3 grupos con un número de muestras similares, utilizando el metodo "qcut". Etiquetaremos estos tres grupos en la columna "Price_Category", con las etiquetas "low","medium" y "high". Crea esta columna, y muestra cuántas filas hay para cada uno de sus valores, así como el precio medio de cada uno de estos grupos. **(1 punto)** EI

[ ]:
```
# Respuesta
```

(f) Tras generar esta segmentación, los siguientes puntos que parecen importar a nuestro compañero son la aceleración y la autonomía. Crea un dataframe que contenga solo los vehículos de la "price_Category" "low" y normaliza las columnas "AccelSec" y "Range_Km". Normaliza dichas dos variables utilizando el módulo `MinMaxScaler` de sklearn. Como la Acceleración es mejor como más pequeña es, deberemos invertir los valores obteniendo un 0 para el valor de "AccelSec" mas alto, y un 1 para el "AccelSec" menor. Crea las dos columnas nuevas para los valores normalizados, y una tercera con el producto de los mismos, que llamaremos "Product". **(1 punto)** EI

[ ]:
```
# Respuesta
```

(g) ¿Utilizando los calculos anteriores, que vehículo de la categoria de precios ”low” muestra el valor ”Product” mas alto? ¿Es el coche más caro de los etiquetados en la categoría ”low”? **(1 punto)** NM

[ ]:
```
# Respuesta
```

(h) **(OPCIONAL)** Genera una gráfica donde podamos ver como evoluciona el precio del vehículo en función de el atributo ”Product” que hemos creado.

[ ]:
```
# Respuesta
```

(i) **(OPCIONAL)** ¿Si no hubieramos limitado nuestra decisión al segmento ”low”, a que conclusión hubieramos llegado? ¿Puedes repetir la normalización a partir de los datos originales, y visualizar la tendencia ”PriceEuro” vs ”Product” ?

[ ]:
```
# Respuesta
```