prog_datasci_6_preproc_entrega

May 10, 2023

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

1.2.1 Ejercicio 1

El dataset 'Customers.csv' es un conjunto de datos que recoge información de los clientes ideales en una tienda imaginaria. El conjunto original se puede encontrar en el repositorio de datos kaggle, pero el conjunto que utilizaremos tiene alguna modificación.

(a) Importa el archivo `Customers.csv` que encontrarás en la carpeta 'data'. Muestra el tipo de variable de cada columna y su nombre, el número total de filas y columnas del dataframe y las primeras 20 filas. **(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

b) Hay una variable que tendría que ser de tipo 'object' pero es una variable numérica. ¿Cuál es y porque? Transfórmala para que sea de tipo 'object'. **(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

c) Eliminar los valores anómalos es un punto clave para el procesamiento de los datos para obtener un dataset de mayor calidad. Observad las variables numéricas y sustituid por `NaNs` los valores anómalos que podáis identificar. Razonad porque los consideráis anómalos. Comprobad que las variables numéricas ya no contienen estos valores anómalos después de la modificación. **(1 punto)** EI

[ ]:
```
# Respuesta
```

1.2.2 Ejercicio 2

a) Muestra los diferentes valores de la columna "Profession" y el número de clientes para cada profesión. ¿Qué profesión proporciona más clientes? Muestra el resultado mediante un `pie chart` **(1 punto)** EG

Nota Tenéis que añadir la frecuencia de cada profesión dentro del `pie chart` con dos decimales. Podéis consultar la documentación de matplotlib para realizar este ejercicio.

[ ]:
```
# Respuesta
```

b) Para aumentar la rentabilidad del negocio, se quiere realizar una campaña publicitaria para aumentar el consumo medio por profesión. Por eso, se quiere orientar la campaña publicitaria al sector con menos consumo medio. ¿Cuál tendría que ser este sector? Ordena el consumo mediano por profesión de mayor a menor. **(1 punto)** NM

[ ]:
```
# Respuesta
```

1.2.3 Ejercicio 3

Queremos respresentar la variable `Age` como una variable discreta en lugar de contínua, porque nos interesa separar esta información en tres categorías: young, adult, old. Separa la variable en tres grupos de forma uniforme, considerando que el primer grupo corresponde a la categoría "young", el segundo corresponde a "adult" y finalmente, la última categoría a "old". Crea una nueva columna con el nombre de "AgeCategory" donde clasificaremos las personas como "young", "adult" y "old", muestra el nombre de filas para cada una de las categorías, así como la mediana de ingresos por categoría. ¿En qué categoría el ingreso medio es más alto? **(1 punto)** EG

Nota Para realizar este ejercicio podéis utilizar las funciones de pandas.

[ ]:
```
# Respuesta
```

1.2.4 Ejercicio 4

El conjunto de datos 'github.csv' contiene información de algunos repositorios de la paltaforma de desarrollo colaborativo GitHub. El que utilizaremos en esta parte tiene algunas modificaciones comparado con el original que se puede encontrar en kaggle.

a) Importa el fichero 'github.csv' que encontrarás en la carpeta 'data'. Muestra el nombre de las columnas y el número total de filas utilizando los atributos de los *dataframes*. Selecciona una muestra del 35% de los datos de forma aleatória. Seguidamente, muestra por pantalla las últimas 10 filas de esta muestra. **(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

b) Para un correcto tratamiento de los datos queremos trabajar con filas que no tengan valores nulos en las columnas. Borra las que tengan alguno y, después, vuelve a mostrar por pantalla la cantidad de filas restantes. **(1 punto)** NM

[ ]:
```
# Respuesta
```

c) Queremos ver cuáles son los 5 usuarios (y solo estos) que tienen más repositorios al conjunto de datos, ordenados de **menos a más**. Para ello, se pide crear dos columnas nuevas a partir de 'repositories': user y repository_name. **(1.5 puntos)** EG

NOTA: La columna 'repositories' tiene el siguiente formato: usuario/nombre_repositorio. Puedes consultar el apartado de *series* de la documentación de Pandas para elegir la función más adecuada.

[ ]:
```
# Respuesta
```

1.2.5 Ejercicio 5

Nos han pedido filtrar la información del conjunto de datos para realizar estadísticas sobre los repositorios de GitHub. A continuación se encuentran los puntos solicitados: **(2 puntos)**

1) Lista los repositorios que tengan más de 900 estrellas o más de 650 forks. **(0.25 puntos)** NM

2) Indica cuantos y qué repositorios no han tenido contribuidores ni estrellas. **(0.25 puntos)** NM

3) ¿Cuál es el lenguaje de programación más utilizado por el usuario openfoodfacts? ¿Cuántas veces lo ha utilizado? **(0.5 puntos)** NM

4) ¿Qué repositorio ha tenido más interacciones? Las columnas que se consideran interacciones son: stars_count, forks_count, issues_count i pull_requests. **(0.5 puntos)** NM

5) Muestra los tres lenguajes con más contribuidores. **(0.5 puntos)** EG

NOTA: Para este último punto utiliza la función *nlargest()* que puedes consultar aquí.

[ ]:
```
# 1) Respuesta
```

[ ]:
```
# 2) Respuesta
```

[ ]:
```
# 3) Respuesta
```

[ ]:
```
# 4) Respuesta
```

[ ]:
```
# 5) Respuesta
```

1.2.6 Ejercicio opcional

Comenta el siguiente código y explica para qué sirve.

[ ]:
```
from sklearn import preprocessing
import warnings
warnings.filterwarnings("ignore")

qt = preprocessing.QuantileTransformer(random_state=0)

df[["AnnualIncome", "SpendingScore"]] = qt.fit_transform(
    df[["AnnualIncome", "SpendingScore"]])

df.head()
```

[ ]:
```
# Respuesta
```