prog_datasci_6_preproc_entrega

April 9, 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

En esta PEC trabajaremos con un conjunto de datos (***dataset***) propio de una película de **ciencia ficción**. Se trata de una base de datos simplificada que recoge características de avistamientos de **OVNIs** reportados en todo el mundo desde finales de la segunda guerra mundial: platillos volantes en Estados Unidos, luces fugaces en el cielo austral, puntos rojos en lo alto de una cumbre, etc. Esperamos que disfrutéis de esta PEC.

1.2.1 Ejercicio 1

Carga el dataset del archivo `ufo.csv` que encontraréis en la carpeta `data` y responde a las siguientes preguntas:

(a) Importa el archivo `ufo.csv` de la carpeta de datos en un dataframe. Examina el conjunto de datos y muestra por pantalla sus dimensiones y el nombre asignado a cada una de las columnas. Luego imprime las 5 primeras entradas y las 10 últimas. **(0.5 puntos)** NM

```
[ ]: # Respuesta
```

(b) Fíjate que los nombres de las columnas no son óptimos para la manipulación del dataset. Modifícalos siguiendo las instrucciones siguientes: sustituye los espacios en blanco y caracteres especiales (paréntesis, asteriscos, etc.) del interior de cada palabra por el caracter `_` (*barra baja* o *underscore*), y elimina los espacios en blanco al principio y final de palabra. Para la realización de este apartado, te puedes inspirar con este link. **(0.5 puntos)** EG

```
[ ]: # Respuesta
```

(c) Continuamos explorando las características del dataset. Imprime por pantalla el tipo de las variables del dataset. Teniendo en cuenta la información de cada variable, ¿te parecen los tipos más adecuados? ¿Qué tipos asignarías a cada variable? **(0.5 puntos)** NM

```
[ ]: # Respuesta
```

(d) Asigna el tipo `float` a las (nuevas) variables `duration_seconds` y `latitude` y verifica que los tipos se hayan cambiado correctamente en el dataframe. Para simplificar el ejercicio, puedes eliminar del dataframe todas las entradas *problemáticas* y para el control de excepciones, puedes utilizar las sintaxis `try` / `except`. **(1 punto)** EG

```
[ ]: # Respuesta
```

(e) ¿Hay duplicados en el dataset? **(0.5 puntos)** NM

```
[ ]: # Respuesta
```

1.2.2 Ejercicio 2

En este ejercicio trabajaremos las variables `city` y `country` para mejorar el conjunto de datos existente y poder realizar análisis más precisos que involucren características geográficas.

(a) Imprime por pantalla una muestra aleatoria de 30 entradas de la variable `city`. Ejecuta la instrucción varias veces para comprobar que los valores van cambiando. **(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

(b) Supongamos que el conjunto de datos se carga en la variable `df_ufo`, explica la utilidad de la siguiente instrucción:

```
df_ufo['city'].str.extract(r"\(([A-Za-z]+)\)", expand=False)
```

(0.5 puntos) EI

[ ]:
```
# Respuesta
```

(c) Si analizas muestras aleatorias diferentes observarás que la variable `city` contiene, en algunos casos, los países en los que se producieron los avistamientos. Sustituye los valores nulos posibles de la variable `country` por los valores correspondientes indicados en la variable `city`. Imprime por pantalla los nuevos valores de la variable `country`.

NOTA1: Aunque no todo lo que aparece en `city` son países, para simplificar el ejercicio, supondremos que sí lo son, sin preocuparnos de unificar nombres.

NOTA2: Recuerda que el valor `NaN` corresponde a un valor perdido y existen diversas maneras de tratarlo, una de ellas se explica en el Notebook de teoría.

(1 punto) EI

[ ]:
```
# Respuesta
```

1.2.3 Ejercicio 3

Como ya hemos visto en el notebook de teoría, las técnicas de preprocesamiento de datos buscan convertir un conjunto de datos bruto en un conjunto de datos idóneo para ser *digerido* por un algoritmo de minería de datos. En este caso, vamos a suponer que nuestro objetivo es predecir qué forma tendrá un OVNI (UFO) avistado en Estados Unidos dado el nombre del estado (variable `state`) donde se ha realizado el avistamiento. Para ello, tendremos que adaptar y mejorar el dataset inicial.

(a) A partir del dataframe inicial, crea un nuevo dataframe con solamente aquellos registros que tengan el campo `country` igual a 'us'. Elimina todas las columnas a excepción de las correspondientes a `state` y `shape`. Del conjunto de datos resultante, ¿qué porcentaje de sus filas contiene al menos un valor nulo? **(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

(b) Siguiendo con este nuevo dataset, elimina todas las filas correspondientes a valores nulos y unifica los valores 'unknown', 'changed' y 'changing' bajo el nombre 'other'. Imprime por pantalla los valores que toma ahora la variable `shape`. **(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

(c) Utiliza `LabelEncoder` de `sklearn` para codificar númericamente la variable `shape` en una nueva columna del mismo dataframe. **(1 punto)** EG

[ ]:
```
# Respuesta
```

(d) Fíjandose en la variable `shape`, ¿crees que este nuevo dataset está balanceado? **(0.5 puntos)** EI

[ ]:
```
# Respuesta
```

1.2.4 Ejercicio 4

En este ejercicio volveremos a trabajar con el conjunto de datos resultante del ejercicio 1, pero esta vez nos focalizaremos en el tratamiento de variables numéricas.

(a) ¿Cuáles son los valores estadísticos (e.g. media, desviación estándar, etc.) de las variables `duration_seconds`, `latitude` y `longitude`? **(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

(b) Añade una columna al dataset que permita clasificar de manera equilibrada entre tres clases de duración: 'long', 'medium', 'short'. Verifica que las clases estan más o menos balanceadas y comprueba que la media aritmética de cada clase es coherente con el tipo de duración asignada (la duración media 'long' debería ser mayor que la 'medium', y ésta que la 'short'). Puedes inspirarte de este post para la realización del ejercicio. **(1 punto)** EG

[ ]:
```
# Respuesta
```

(c) Transforma el conjunto de datos inicial para que contenga columnas con la codificación binaria de cada una de las tres clases creadas en el apartado anterior. **(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

(d) Imagina que queremos crear un modelo para clasificar zonas geográficas según son más o menos propensas a avistamientos de OVNIs y creemos que los datos de latitud y longitud de este dataset nos podrían ser útiles. Normaliza dichas dos variables utilizando el módulo `StandardScaler` e indica cómo han cambiado sus estadísticas básicas. **(0.5 puntos)** NM

[ ]:
```
# Respuesta
```

1.2.5 Ejercicio opcional

Ejecuta la siguiente instrucción para la instalación del package `wordcloud`:

```
!pip install wordcloud
```

Utiliza este nuevo package para construir una nube con las palabras más usadas en la variable `comments` del dataframe inicial. Muestrala por pantalla mediante la función `imshow` de `matplotlib.pyplot`. **(Opcional)** EG

[ ]:
```
# Respuesta
```