prog_datasci_6_preproc_entrega

December 6, 2019

1 Programación para *Data Science*

1.1 Unidad 6: Preprocesamiento de datos en Python

2 Introducció

En este Notebook encontraréis dos conjuntos de ejercicios: un primer conjunto de **ejercicios para practicar** y un segundo conjunto de **actividades evaluables** como PEC de la asignatura.

En cuanto al conjunto de ejercicios para practicar, éstos no puntúan para la PEC, pero os recomendamos que los intentéis resolver como parte del proceso de aprendizaje. Encontraréis ejemplos de posibles soluciones a los ejercicios al propio notebook, pero es importante que intentéis resolverlos vosotros antes de consultar las soluciones. Las soluciones os permitirán validar vuestras respuestas, así como ver alternativas de resolución de las actividades. También os animamos a preguntar cualquier duda que surja sobre la resolución de los **ejercicios para practicar** en el foro del aula.

En relación a las actividades evaluables, veréis que cada una de ellas tiene asociada una puntuación que indica el peso que tiene la actividad sobre la nota 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. Podéis 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.

Además, veréis que todas las actividades 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 independente**: 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.

2.1 Ejercicios para practicar

Los siguientes 3 ejercicios no puntúan para la PEC, pero os recomendamos que los intentéis resolver antes de pasar a los ejercicios propios de la PEC. También encontraréis las soluciones a estos ejercicios al final del Notebook.

2.2 Ejercicio 1

Cargue los datos del fichero `bank_edited.csv` en un dataframe. Este conjunto de datos recoge información respecto a una campaña de marketing de un banco portugués. El conjunto original se puede encontrar en el [repositorio de datos de Machine Learning de la UC Irvine] (http://archive.ics.uci.edu/ml/datasets/Bank+Marketing), pero el conjunto que utilizaremos tiene alguna modificación .

Observación: revise la documentación de la función `read_csv` para ver qué parámetro disponemos para ajustar el proceso de cargar de datos.

Los valores del estado civil (atributo `marital`) contienen errores tipográficos y incluyen el uso de diferentes nomenclaturas. En este ejercicio unificaremos la nomenclatura de los valores de esta variables.NM

a) £Cuantos valors diferentes tiene el atributo `marital` en el conjunto de datos? Mostrad estos valores.

[10]:
```
# Respuesta
```

b) Unificad los atributos `marital` en los valores: “single”, “married” o “divorced”.

[11]:
```
# Respuesta
```

c) £Qué columnas contienen valores perdidos?

[12]:
```
# Respuesta
```

d) Calculad el primer y el tercer cuartil del atributo “balance”.

[13]:
```
# Respuesta
```

2.3 Ejercicio 2

El atributo `poutcome` contiene información sobre si el cliente del banco contractó un deposito. Calula la correlacióm entre el atributo `poutcome` y el resto de atributos (usa la función 'corr'). £Qué variable presenta mayor correlación con `poutcome`?NM

[14]:
```
# Respuesta
```

2.4 Ejercicio 3

El módulo sklearn incluye varios datasets de ejemplo, dentro del módulo `sklearn.datasets`. Estos datasets se almacenan en formato `Bunch`, propio de sklearn. Un `Bunch` es un objeto tipo diccionario, los atributos interesantes son: `data`, con los datos en crudo, `target`, con generalmente las etiquetas de clasificación o etiquetas objetivo, `target_names`, el significado de las etiquetas, `feature_names`, el significado de las características o atributos, `DESCR` , la descripción completa del conjunto de datos.

Importa el dataset `iris` de `sklearn`. Almacena los datos este dataset como un objeto `pandas`, con los correpondientres nombres de variables. Añade la variable `target` en el dataframe con el nombre de atributo `Species` y los valors con el tipo de especie de cada muestra.EG

[15]:
```
# Respuesta
```

3 Ejercicios para la PEC

A continuación, los **ejercicios y preguntas teóricas que debe completar en esta PEC** y que forman parte de la evaluación de esta unidad.

4 Ejercicio 1

Importa el fichero `siri.csv` de la carpeta de datos. En ella encontrarás notas (en un rango de 0 a 10) correspondientes a estudiantes de tres escuelas diferentes para cuatro asignaturas. Una vez importados los datos, realiza las siguientes operaciones:

a) Muestra por pantalla una lista de las diferentes escuelas en los datos. En caso no observar tres escuelas por errores tipográficos, mayúsculas y otros, corrige la nomenclatura para homogeneizar una columna con las tres escuelas.

(1 punto) NM

[16]:
```
# Respuesta
```

b) Buscar notas claramente anómalas. En caso de encontrar notas anómalas sustitúyelas por NaN.

(1 punto) EI

[17]:
```
# Respuesta
```

c) Encuentra los valores perdidos. Imputa los valores perdidos por la media de los datos por columna

(0.5 puntos) NM

[18]:
```
# Respuesta
```

d) Comprueba que ya no hay valores perdidos en el objeto pandas.

(0.5 puntos) NM

[19]:
```
# Respuesta
```

4.1 Ejercicio 2

En este ejercicio trabajaremos un conjunto de datos que se derivó del censo estadounidense de 1990, utilizando una fila por censo dentro de un grupo de viviendas o bloques. Un grupo de bloques es la unidad geográfica más pequeña para la que La Oficina del Censo de EE.UU. publica datos de muestra (un grupo de bloques generalmente tiene una población de 600 a 3,000 personas).

- Pace, R. Kelley and Ronald Barry, Sparse Spatial Autoregressions, Statistics and Probability Letters, 33 (1997) 291-297

a) Generar un objeto `pandas` de nombre `housings` con los datos que genera la función `datasets.fetch_california_housing()` dentro de `sklearn`. El objeto debe contener todas las variables del conjunto de datos incluida una columna con los valores de `Prices` (target).

(1 punto) EG

[20]:
```
# Respuesta
```

b) Obtener, los nombres de las variables, el número total de filas y columnas, y comprobar la posible existencia de valores perdidos en los datasets. Obtener la media, desviación típica, valores máximos y mínimos de todas las variables del data frame.

(0.5 puntos) NM

[21]:
```
# Respuesta
```

c) Calcula la correlación de la variable `Prices` con el resto de variables. Ordena el vector de correlaciones de mayor a menor en valor absoluto. £Qué variable aparece con más correlación con los precios? Interpreta los resultados.

(0.5 puntos) NM

[22]:
```
# Respuesta
```

4.2 Ejercicio 3

Con la mismo dataset anterior:

a) Con cada una de las variables `Latitude` y `Longitude`, crea dos variables discretizadas con 10 segmentos, de forma que:

- en las primeras cada intervalo contenga el mísmo número de muestras (equidensidad por segmento) en (`qLat`, `qLon`)
- en las segundas cada intervalo tengan la misma longitud de intervalo (equidistancia por segmento) en (`sLat`,`sLon`)

(1 punto) EI

[23]:
```
# Respuesta
```

b) Observa la distribución en las variables discretizadas calculando el número de muestras dentro de cada intérvalo. Podéis mirar el método de pandas .value_counts() para contar valores de cada grupo.

(0.5 puntos) EG

[24]:
```
# Respuesta
```

c) Muestra

- los precios promedio por cada segmento de Latitud (sLat)
- los precios promedio por cada segmento de Longitud (sLon)

(1 punto) NM

[25]:
```
# Respuesta
```

d) Calcula y muestra la media y la desviación típica para todas las variables para cada combinación de Latitud y Longitud en ambos casos (equidensidad por segmento, equidistancia por segmento). Observación: quita las filas que contengan algún NaN del dataframe resultante.

(1 punto) NM

[26]:
```
# Respuesta
```

4.3 Ejercicio 4

Estandarización de variables.

Importa el dataset `iris` de `sklearn`. Almacena este dataset como un objeto `pandas`, con los correpondientres nombres de variables.

Llamamos tipificar una variable al proceso de convertirla en una Normal Estándar $N(\mu = 0, \sigma = 1)$. Busca en `sklearn.preprocessing` un método que te permita tipificar cada variable numérica del objeto pandas de `iris` de forma que:

- El valor medio de cada columna sea 0
- La desviación típica de cada columna sea 1

a) Aplica el método para crear un nuevo dataframne `iris_typ` que cumpla esas condiciones. Comprueba que la tipificación es correcta.

(1 punto) EG

[27]:
```
# Respuesta
```

b) Busca y aplica un segundo mecanismo de normalización de forma que:

- el mínimo de la variable sea 0, y
- el máximo de cada variable sea 1

Comprueba que la normalizacion es correcta
(0.5 puntos) EG

[28]:
```
# Respuesta
```

4.3.1 Ejercicio Opcional

La Oficina de Estadísticas de Transporte del Departamento de Transporte de los Estados Unidos (DOT) rastrea el rendimiento en término de puntualidad de los vuelos en EE.UU. operados por grandes compañías aéreas.

La información resumida sobre el número de vuelos puntuales, retrasados, cancelados y desviados se publica en el Informe mensual del consumidor de viajes aéreos del DOT y en este conjunto de datos de retrasos y cancelaciones de vuelos de 2015. Encontrarás una muestra de estos datos en el fichero `vuelos.csv` en la carpeta de datos.

Importa el fichero de vuelos. Imputa los valores perdidos con el método KNNImputer con los valores por defecto.

Calcula y muestra la matriz de correlación de las siguientes variables a partir de la matriz imputada.

- `ARRIVAL_DELAY`
- `DIVERTED`
- `CANCELLED`

- `AIR_SYSTEM_DELAY`
- `SECURITY_DELAY`
- `AIRLINE_DELAY`
- `LATE_AIRCRAFT_DELAY`
- `WEATHER_DELAY`

Interpreta el resultado
EI

[29]:
```
# Respuesta
```

4.4 Soluciones a los ejercicios para practicar

4.5 Ejercicio 1

a) £Cuantos valores diferentes tiene el atributo `marital` en el conjunto de datos? Mostrad estos valores.

[30]:
```
import pandas as pd
import numpy as np

data = pd.read_csv("data/bank_edited.csv", sep=";", dtype={"balance":np.float})

import numpy as np

# Con unique podemos encotrar los valors únicos

v = data.marital.unique()

print("There are {} different values in marital:\n{}".format(len(v), v))
```

```
There are 11 different values in marital:
['married' 'single' 'marrid' 'divorced' 'maried' 'sing' 'Married'
 'MARRIED' 'DIVORCED' 'Single' 'SINGLE']
```

b) Unificad los atributos `marital` en los valores: "single", "married" o "divorced".

[31]:
```
data.loc[(data.marital == "Married") | (data.marital == "maried") | (data.
 ↪marital == "MARRIED") |
         (data.marital == "marrid"), "marital"] = "married"
data.loc[(data.marital == "Single") | (data.marital == "SINGLE") | (data.
 ↪marital == "sing"), "marital"] = "single"
data.loc[(data.marital == "DIVORCED"), "marital"] = "divorced"

# Comprobación

v = data.marital.unique()
print("There are {} different values in marital:\n{}".format(len(v), v))
```

```
There are 3 different values in marital:
['married' 'single' 'divorced']
```

c) Qué columnas contienen valores perdidos?

[32]:
```
# Para esto usamos la función  any_isna a cada columna del dataframe
print(data.isna().any())
```

```
age          False
job          False
marital      False
education    False
default      False
balance       True
housing      False
loan         False
```

```
contact      False
day           True
month        False
duration      True
campaign     False
pdays        False
previous     False
poutcome     False
y            False
dtype: bool
```

d) Calculad el primer y el tercer cuartil del atributo `balance`.

[33]:
```
print((data['balance'].quantile(0.25)))
print(data['balance'].quantile(0.75))
```

```
68.0
1476.0
```

4.6 Ejercicio 2

El atributo `poutcome` contiene información sobre si el cliente del banco contractó un deposito. Calula la correlacióm entre el atributo `poutcome` y el resto de atributos (usa la función 'corr'). Qué variable presenta mayor correlación con 'poutcome'?

[34]:
```
# Visualizamos los valores de la columna poutcome
import pandas as pd
import numpy as np
import warnings

warnings.filterwarnings('ignore')

data.poutcome.unique()
```

[34]:
```
array(['unknown', 'failure', 'other', 'success'], dtype=object)
```

[35]:
```
# Seleccionamos únicamente la muestras que contienen  informacióno precisa␣
 ↪sobre el ciente contrató o no el depósito

data_pout = data[data.poutcome.isin(("failure","success"))]

# Discretizamos la columna poutcome para poder calcular la correlación

data_pout['poutcome_cat'] = data_pout.poutcome.astype("category").cat.codes

# Calculamos la correlación con el resto de columnas

data_pout.corr()["poutcome_cat"]
```

[35]:
```
age             0.090540
balance         0.039791
day             0.009252
duration        0.142385
campaign       -0.059986
pdays          -0.276853
previous        0.023411
poutcome_cat    1.000000
Name: poutcome_cat, dtype: float64
```

Observad que la columna `pdays` muestra tener mayor correlació en valor absoluto con poutcome_cat. Tened en cuenta que hemos asignado un 0/1 a "failure"/"succes" de manera arbitraria, así que el signo no es significtivo en este caso.

4.7 Ejercicio 3

El módulo sklearn incluye varios datasets de ejemplo, dentro del módulo `sklearn.datasets`. Estos datasets se almacenan en formato `Bunch`, propio de sklearn. Un `Bunch` es un objeto tipo diccionario, los atributos interesantes son: `data`, con los datos en crudo, `target`, con generalmente las etiquetas de clasificación o etiquetas objetivo, `target_names`, el significado de las etiquetas, `feature_names`, el significado de las características o atributos, `DESCR` , la descripción completa del conjunto de datos.

Importa el dataset `iris` de `sklearn`. Almacena los datos este dataset como un objeto `pandas`, con los correpondientres nombres de variables. Añade la variable `target` en el dataframe con el nombre de atributo `Species` y los valors con el tipo de especie de cada muestra.

[36]:
```
from sklearn import datasets
import pandas as pd
iris_bunch = datasets.load_iris()
iris_pandas = pd.
 ↪DataFrame(iris_bunch['data'],columns=iris_bunch['feature_names'])
iris_pandas['Species'] = iris_bunch['target_names'][iris_bunch['target']]
```

[37]:
```
iris_pandas.head()
```

[37]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	Species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

[38]:
```
iris_pandas.tail()
```

[38]:
```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
145                6.7               3.0                5.2               2.3
146                6.3               2.5                5.0               1.9
147                6.5               3.0                5.2               2.0
148                6.2               3.4                5.4               2.3
149                5.9               3.0                5.1               1.8

       Species
145  virginica
146  virginica
147  virginica
148  virginica
149  virginica
```

[39]:
```
iris_pandas.describe()
```

[39]:
```
       sepal length (cm)  sepal width (cm)  petal length (cm)  \
count         150.000000        150.000000         150.000000
mean            5.843333          3.057333           3.758000
std             0.828066          0.435866           1.765298
min             4.300000          2.000000           1.000000
25%             5.100000          2.800000           1.600000
50%             5.800000          3.000000           4.350000
75%             6.400000          3.300000           5.100000
max             7.900000          4.400000           6.900000

       petal width (cm)
count        150.000000
mean           1.199333
std            0.762238
min            0.100000
25%            0.300000
50%            1.300000
75%            1.800000
max            2.500000
```