

prog_datasci_7_analisis_entrega

December 15, 2021

1 Fundamentos de Programación

1.1 PEC 7 - Solución

A continuación, encontraréis la solución a los ejercicios de esta PEC.

Recordad, que como en PECs anteriores, pueden existir otras maneras de resolver los ejercicios, también válidas (y que también permiten obtener la máxima puntuación de los mismos).

1.1.1 Ejercicio 1

En las etiquetas y en los envases de los alimentos que consumimos habitualmente es muy común encontrar su composición. Algunos alimentos son mas ricos en grasas, azucares,... En la web https://www.matportalen.no/verktoy/the_norwegian_food_composition_table/ podemos obtener una tabla muy completa de la composición de diferentes alimentos. Os proporcionamos un extracto en el fichero `Composition3.csv`.

Estos alimentos están categorizados en diferentes grupos que podemos encontrar en la columna `Category`. En este primer ejercicio os proponemos hacer un primer análisis de esta tabla. **(3.5 puntos)**

- a) Carga la tabla en un dataframe. Recuerda de tener en cuenta los separadores decimales, y la codificación del fichero. Muestra los primeros 5 elementos y comprueba que las columnas generados son correctas. Si hay alguna que no lo es correcta, borrarla. NM **(0.5 puntos)**

[17]: `# Respuesta`

- b) Muestra el número de alimentos (filas), elimina las filas que tengan un elemento vacío en alguna de sus columnas, y vuelve a contarlas. NM **(1 punto)**

[18]: `# Respuesta`

- c) Hay filas con el valor `M` en alguna de sus columnas. Substituye estas `M` por el valor especial de numpy `NaN`. Una vez realizado este paso, comprueba el tipo de cada columna. Todas deberían ser de tipo numérico a excepción del nombre del alimento y su categoría. Si no es así, convierte lo que haga falta, estando atento a los separadores decimales. Después de realizar este paso, haz una gráfica que permita ver el rango de todos los atributos. NM **(1 punto)**

[19]: *# Respuesta*

- d) Haz una PCA de 2 componentes de la composición, y muestra los componentes y la varianza explicada de cada uno de ellos. ¿Que quieren decir los valores obtenidos? NM **(1 punto)**

[20]: *# Respuesta*

- e) Muestra una gráfica donde podamos ver como evoluciona la variabilidad explicada, a medida que vamos incrementando hasta 10 componentes. ¿Tiene sentido hacer una reducción a tantos componentes? **(Opcional)**

[21]: *# Respuesta*

1.1.2 Ejercicio 2

En este ejercicio nos gustaría ver si podemos agrupar los alimentos en grupos diferentes según sus características de una manera **no supervisada**. Con esa finalidad, utilizaremos el método **KMeans** que hemos visto en la teoría. **(3 puntos)**

- a) Antes de aplicar el KMeans, normaliza los datos utilizando la función **MinMaxScaler**. ¿Por que razón es importante hacer esta transformación? NM **(1 punto)**

[22]: *# Respuesta*

- b) Uno de los problemas de los métodos no supervisados es identificar el número óptimo de clusters. Para poder estimar esta número óptimo, se utiliza frecuentemente **el método de Elbow**. Busca información sobre este método y utilízalo. ¿Cual es el número óptimo de clusters? EI **(1 punto)**

[23]: *# Respuesta*

- c) Aplica el método KMeans especificando el número óptimo de clusters obtenido en el apartado anterior. Compara las agrupaciones obtenidas con la categoría de la columna **Category**. Podemos encontrar alguna relación entre el clustering del KMeans y las categorías (Cereals, Vegetables, Meat, etc) de la base? NM **(1 punto)**

[24]: *# Respuesta*

1.1.3 Ejercicio 3

En los siguientes ejercicios trabajaremos con el dataset **titanic_edited.csv**. Este conjunto de datos contiene información sobre los pasajeros del barco Titánic (edad, sexo, clase, tipo cabina, etc), y si sobrevivieron o no. El conjunto original se puede encontrar en la web de [Kaggle](#), pero el conjunto que vamos a utilizar tiene alguna modificación.**(3.5 puntos)**

- (a) Si nuestro objetivo es realizar un modelo que nos permita determinar si, dadas unas características, el pasajero sobrevivirá o no, ¿nos encontramos ante un problema de clasificación o de regresión? NM**(0.5 puntos)**

[25]: # Respuesta

- (b) Explora el dataset: ¿Qué variables crees que pueden ser relevantes en la supervivencia de los pasajeros? Razona la respuesta. Haz una matriz de [correlación](#) de las variables para encontrar relaciones relevantes entre sí. NM(0.5 puntos)

[26]: # Respuesta

- (c) Basándonos en la teoría de la anterior unidad de preprocesamiento (unidad 6), sigue los siguientes pasos:
- (1) Comprueba si existen variables con valores perdidos y, en caso de existir, elimínalas.
 - (2) Algunas variables son categóricas y otras numéricas. ¿Tienes que hacer alguna transformación para poder trabajar con ellas? NM(1 punto)

[27]: # Respuesta

- (d) ¿Se puede hacer un modelo preciso con la información proporcionada para predecir si los pasajeros sobrevivirán o no? Explora la función [RandomForestClassifier](#). Recuerda separar el conjunto de datos en una parte de entrenamiento (training, 80%) y una parte de evaluación (test, 20%). EG (0.5 puntos)

[28]: # Respuesta

- (e) ¿Qué variables son más importantes para el modelo? Explora el método `feature_importances_` de la función `RandomForestClassifier`. EG (1 punto)

[29]: # Respuesta

- (f) **(Opcional)** En los apartados anteriores hemos visto la importancia de separar nuestro dataset en el subconjunto de entrenamiento y evaluación. Sin embargo, no podemos asegurar que nuestro modelo sea muy generalizable ya que sólo lo hemos probado por una partición de train/test concreta. Para mejorar la validación de nuestro modelo, podemos utilizar lo que se conoce como **validación cruzada**.
- (1) Busca información sobre el método k-Fold y explica en qué consiste. (2) A continuación aplica un k-Fold con k=10. ¿Podemos concluir que nuestro modelo es bueno para predecir la supervivencia de los pasajeros? EG

[30]: # Respuesta