Práctica 9.

Aprendizaje Automático

Fecha de entrega: 5 de mayo de 2019

Esta práctica tiene como objetivo aplicar algunos de los algoritmos de aprendizaje automático disponibles en el entorno Scikit-Learn de Python a distintos conjuntos de datos. Se entregarán los notebooks utilizados que deben incluir las respuestas a las preguntas planteadas, los resultados obtenidos y la interpretación de dichos resultados.

Los conjuntos de datos

Por cada conjunto de datos que utilices deberás incluir una breve descripción del mismo.

- Nombre del conjunto de datos
- Breve descripción del problema que describe
- Nombre y tipo de las variables
- Estadísticos descriptivos de cada variable

Parte 1: Agrupamiento o clustering

Usa el conjunto de datos de las viviendas en California que puedes descargar con la función: `sklearn.datasets.fetch_california_housing`

En el conjunto de datos cada fila es un "bloque" que es la mínima unidad geográfica para la cual el censo estadounidense hace públicos los datos (suele tener desde varios cientos a unos pocos miles de habitantes).

El objetivo es realizar agrupamientos de los bloques que presenten características similares. No incluyas la latitud y longitud para no agrupar los bloques por su cercanía geográfica, sino por sus características. Tampoco incluyas la variable target del dataset que es la mediana del precio de las viviendas en el bloque en cuestión.

1) Describe el conjunto de datos tal y como se indica más arriba y extrae algunas conclusiones de las variables, su distribución y su correlación.
2) Considera si debes re-escalar las variables antes y el tipo de escalado que usas. Razona tu elección.
3) Aplica el algoritmo de clustering k-medias y determina el número de clusters que consideras adecuado para el conjunto de datos, justificando tu elección.
4) Trata de averiguar qué representa cada uno de los clusters que has obtenido. Si en el apartado anterior has obtenido más de 5 clusters, basta con que comentes los dos más numerosos y los dos menos numerosos. ¿Qué valores toman las variables en cada cluster? Puedes usar estadísticos descriptivos
   Te recomendamos que uses las variables en su escala original y no en la transformada (ya que se interpretará mucho peor).

5) Pinta los clusters en un gráfico de dispersión en función de dos de las variables de entrada que consideres interesantes. ¿Ves que considerando solamente esas dos variables se diferencien bien algunos de los clusters? ¿Cuáles? Por el contrario, ¿cuáles se confuden más? Te recomendamos que en este caso uses las variables tal y como las usaste para hacer el clustering (es decir, re-escaladas, si las re-escalastes) para así diferenciar los clusters mejor

Documenta todo el proceso en un notebook de jupyter con comentarios, texto explicando las soluciones y toda la información que consideres necesaria.

Parte 2: Clasificación

Usa el conjunto de datos de los vinos que puedes cargar con la función:
`sklearn.datasets.load_wine`

En este caso, realizaremos una tarea de clasificación donde cada elemento a clasificar es un vino que viene descrito por una serie de propiedades numéricas y que puede pertenecer a una clase de vino entre tres posibles.

1) Describe el conjunto de datos tal y como se indica más arriba y extrae algunas conclusiones de las variables y su distribución.
2) Considera si debes normalizar o estandarizar las variables antes para usar un árbol de decisión. Razona tu elección.
3) Configura una partición de los datos con un 30% para el conjunto de test y estratificando la muestra.
   Analiza los resultados de entrenamiento y test que obtiene un árbol de decisión en función de la profundidad máxima del árbol. Pinta la evolución de la curva de aprendizaje.
   Determina el valor óptimo de dicho parámetro de manera razonada.
4) Pinta el árbol de decisión óptimo que has encontrado y analiza lo siguiente:
   a) Interpreta someramente la pregunta que se realiza en el nodo raíz y los nodos hijos resultantes. Hazlo tanto en el contexto de un problema de clasificación (¿qué clases ha clasificado mejor?), como en el del problema representado en el conjunto de datos (¿qué sentido tiene esa pregunta y la clasificación que infiere dentro del problema?).
   b) Analiza si hay variables que sirven para discriminar entre algunas clases
   c) Analiza si hay variables del conjunto de datos que no se han usado.
   d) Identifica los nodos en los que existe mayor confusión.
5) Pinta un árbol de decisión sub-óptimo que sobreaprenda. Por ejemplo, el que se obtiene para un nivel más de profundidad máxima. Identifica los nodos nuevos.
6) Crea la matriz de confusión de los datos de test. Analiza también los valores de “precision” y “recall” (exhaustividad) para cada una de las clases (usa para ello **sklearn.metrics.classification_report**).
7) Configura un clasificador k-NN para la misma partición de datos.
   a) Determina si tiene sentido o no escalar los datos.
   b) Encuentra el valor óptimo de k que no sobreaprenda.
   c) Compara los resultados de precisión y exhaustividad de ese k-NN óptimo con los que obtiene el árbol de decisión.

Documenta todo el proceso en un notebook de jupyter con comentarios, texto explicando las soluciones y toda la información que consideres necesaria.

Parte 3: Regresión

Usa el conjunto de datos de la diabetes que puedes cargar con la función **`sklearn.datasets.load_diabetes`**

En este caso, realizaremos una tarea de regresión donde cada elemento del conjunto de datos es un paciente descrito por una serie de características y un conjunto medidas sobre el suero sanguíneo. La variable a predecir es una medida cuantitativa del progreso de la enfermedad un año más tarde.

1) Describe el conjunto de datos tal y como se indica más arriba y extrae algunas conclusiones de las variables, su distribución.
   Presta especial atención a la matriz de gráficos de dispersión y en especial a la fila de la variable a predecir, ya que nos interesa saber qué variables que están relacionadas con ella.
2) Considera si debes normalizar o estandarizar las variables antes para usar un perceptrón multicapa. Razona tu elección.
3) Vamos a entrenar dos tipos de redes:
   - MLP1 con una capa oculta de 200 neuronas
   - MLP2 con dos capas ocultas de 10 neuronas cada una

   Realiza una validación cruzada de cada una de ellas con k=10 variando el parámetro alpha que controla el aprendizaje del perceptrón y determina el valor óptimo (es decir aquel que maximiza el Mean Square Error en negativo).
   Asegúrate de que no salen warnings indicando que no se ha alcanzado la convergencia durante el entrenamiento (basta con poner un número de max_iter suficientemente grande).
   Pinta la curva de aprendizaje de cada perceptrón. ¿Alguno de los dos perceptrones domina al otro? ¿Por qué crees que se producen las diferencias?

Entrega

La entrega se realizará a través del campus virtual subiendo un fichero comprimido con los notebooks de jupyter (uno por cada apartado). En la primera celda de cada notebook debe aparecer el número de grupo y los nombres completos de sus integrantes.

Además, el nombre del archivo comprimido será P9GXX, siendo XX el número de grupo.