Práctica 1. Aprendizaje Automático

Fecha de entrega: 8 de marzo de 2020

Esta práctica tiene como objetivo aplicar a distintos conjuntos de datos algunos de los algoritmos de aprendizaje automático disponibles en el entorno Scikit-Learn de Python. Se elaborará una memoria mediante notebooks de jupyter que incluirán los apartados debidamente señalizados, con su código, la salida del mismo cuando corresponda, y las respuestas a las preguntas planteadas.

IMPORTANTE: Usa la opción **random_state** en las funciones que generan las particiones de datos y que implementan los algoritmos de aprendizaje automático para que los resultados del notebook sean reproducibles en todo momento.

Los conjunto 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
- Tabla con el nombre y tipo de las variables
- Tabla de estadísticos descriptivos de cada variable

Parte 1: Agrupamiento o clustering

Usa el conjunto de datos de causas de arrestos en los diferentes estados de Estados Unidos en 1973 que puedes descargar del campus virtual. El conjunto contiene las cifras de arrestos por asalto, asesinato y violación por 100.000 residentes. Además, contiene la variable con el porcentaje de población que vive en áreas urbanas, la cual no usaremos para el clustering.

El objetivo es realizar agrupamientos de los estados que presenten un perfil de arrestos similar e identificar cuál ese ese perfil de arrestos y qué estados pertenecen a él.

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 un algoritmo de clustering de los que hemos visto en clase con una parametrización (el valor de *k* en el algoritmo de *k*-medias, o la forma en la que se agrupan clusters en el caso jerárquico).
   Determina el número de clusters que consideras adecuado para el conjunto de datos y justifica tu elección.
4) Da un sentido a cada uno de los clusters que has obtenido en el contexto del problema que representa el conjunto de datos.
   Si obtienes un número mayor de 4 clusters, comenta solamente los dos los dos más numerosos y los dos menos numerosos.
   Para analizar los clusters:
   - Usa estadísticos descriptivos (número de individuos, media, desviación típica, mediana, cuartiles) para describir los clusters.

- Usa una matriz de gráficos de dispersión que pinte los clusters usando un color diferente para ver la separación de los clusters en función de cada par de variables de entrada. ¿Qué clusters se separan mejor y en función de qué variables? ¿y cuáles se cofunden más?
    - Para ello, usa la función **`seaborn.pair_plot`** de la librería de representación gráfica seaborn, como puedes ver en este ejemplo

Te recomendamos que uses las variables en su escala original y no en la transformada, ya que en la escala original puedes relacionar los valores de las variables con lo que representan en la vida real.

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_breast_cancer`
En este caso, realizaremos una tarea de clasificación donde cada elemento a clasificar es un tumor que viene descrito por una serie de variables numéricas y que puede ser benigno o maligno.

De cada tumor se han tomado ciertas medidas que describen diez propiedades (textura, perímetro, etc). Por cada propiedad hay tres variables, mean, error y worst, que describen el valor medio, la desviación típica y los valores extremos, para cada una de esas propiedades. En total hay, por tanto, 30 variables.

1) Describe el conjunto de datos tal y como se indica más arriba y extrae algunas conclusiones de las variables y su distribución entre las dos clases. Como únicamente son dos clases, prueba a representar los datos con la función **`seaborn.pair_plot`** usa la opción **`corner=True`** para evitar pintar excesivas variables.
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 uno de los parámetros que regulan la capacidad de aprendizaje del árbol (elige el que consideres oportuno). 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 qué variables tienen mayor poder discriminante.

c) Analiza si hay variables del conjunto de datos que no se han usado (en este caso, mejor indica qué tipo de propiedades no ha usado el árbol, y qué tipo de medidas no ha usado).
d) Identifica los nodos hoja en los que existe mayor confusión, o que creas que han podido generar particiones poco generalizables.

5) Pinta un árbol de decisión sub-óptimo que sobreaprenda. Por ejemplo, el que se obtiene para el siguiente valor al óptimo en la curva de aprendizaje que realizaste.
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) Dado que el conjunto de datos tiene muchas variables, elige según tu criterio entre tres y cinco variables para medir distancias entre las observaciones. Puedes usar aquellas variables que mejor separen las clases según la inspección visual del punto 1 o según el árbol óptimo del punto 4.
b) Determina si tiene sentido o no escalar los datos.
c) Encuentra el valor óptimo de k que no sobreaprenda.
d) 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 del fichero countries.csv disponible en el campus virtual y que se ha obtenido del repositorio Kaggle.

https://www.kaggle.com/noxmoon/world-countries-predicting-gdp

En este caso, realizaremos una tarea de regresión donde cada elemento del conjunto de datos es un páis descrito por una serie de características sociales, económicas, geográficas y demográficas. La variable a predecir es el PIB del país (GDP en inglés de Gross Domestic Product).

0) Antes de empezar vamos a eliminar la variable "Region" del conjunto de datos, ya que no la usaremos para predecir.
Además, utiliza la función **dropna** que elimina de un data frame todas las filas (es decir, observaciones) que tienen un valor perdido (*na* quiere decir *not available*) para alguna variable. De esta forma, no tendrás que preocuparte por imputar un valor a dichas observaciones.
Asegúrate de que el dataframe resultante no tiene la variable Region, ni valores perdidos.
1) Describe el conjunto de datos tal y como se indica más arriba y extrae algunas conclusiones de las variables, especialmente a la matriz de gráficos de dispersión y al coeficiente de correlación de la variable objetivo (GDP) con el resto de variables. ¿Hay variables que tengan una relación clara con ella?

2) Establece un criterio en base al coeficiente de correlación para filtrar aquellas variables que tengan poca relación (ya sea directa o inversa) con la variable objetivo. Di qué variables pasan tu filtro y qué coeficiente de correlación tienen con ella.
3) Considera si debes normalizar o estandarizar las variables antes para usar un perceptrón multicapa de decisión. Razona tu elección.
4) Configura una validación cruzada con k=5 y dos perceptrones multicapa
    - MLP1 con una capa oculta de 200 neuronas
    - MLP2 con dos capas ocultas de 50 neuronas cada una

    Pinta la curva de aprendizaje para cada perceptrón 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).

    ¿Alguno de los dos perceptrones domina al otro? ¿Por qué crees que se producen las diferencias?
5) Entrena el perceptrón elegido con todo el conjunto de datos y genera las predicciones del GDP que hace el perceptrón para todo los países.

    A continuación, píntalas en un diagrama de dispersión frente a los valores observados para el GDP (en el eje Y) y pinta la recta que se genera con la predicción perfecta de todos los valores.

    Detecta países para los cuales el perceptrón a infraestimado más su GDP (aproximadamente). Idem con países donde se ha sobreestimado mucho su GDP. ¿Puedes aventurar alguna razón por la que esos países tienen más (o menos) GDP del que el perceptrón les asigna? ¿Es realmente un problema del perceptrón o eres capaz de ver alguna razón económica, política o social?

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 P1GXX, siendo XX el número de grupo.