Actividad: Resolver un problema de clasificación

**Objetivos de la actividad**

Con esta actividad vas a conseguir resolver un problema de clasificación aplicando los modelos basados en árboles de decisión y mediante técnicas de ensamble. Además, conseguirás evaluar la importancia de las variables para distintos modelos y realizar comparaciones entre distintas técnicas. Asimismo, aumentarás tu práctica en Python al crear un conjunto de datos mediante simulación.

**Descripción de la actividad y pautas de elaboración**

El primer paso consiste en crear un conjunto de datos ficticio. Para garantizar que cada alumno obtiene uno distinto se va a emplear el documento de identidad de cada uno para crear el conjunto de datos. Para que sean comparables entre todos, si el número de identidad tiene menos de 8 cifras replicaremos las primeras hasta obtener exactamente 8. Además, para evitar los dígitos 0 y 1, si alguna de las cifras es menor que 2 la sustituiremos por ese número. Aplicando estos cambios tendremos el número del documento de identidad preparado para la resolución de la actividad. Veamos algunos ejemplos:

* Ejemplo 1:

12345678 -> 22345678

* Ejemplo 2:

304156 -> 304145630 -> 32425632

Una vez que tengamos el número del documento de identidad preparado crearemos el conjunto de datos para el problema de clasificación con la función *sklearn.datasets.make\_classification* de la librería *scikit-learn* (ver <https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html>).

* Crea el conjunto de datos empleando los siguientes argumentos:
  + n\_muestra = 200 + 10 · primer dígito dni
  + n\_predictores = 10 + segundo dígito dni + tercer dígito dni
  + n\_informativas = 10 + segundo dígito dni
  + semilla = numero\_dni
  + shuffle = False

Este conjunto de datos tendrá al menos 220 observaciones y 14 variables predictoras, de las cuales al menos 2 no estarán relacionadas con la variable respuesta. Se añade el sesgo y el ruido para que la variable respuesta no sea una combinación lineal exacta de las variables informativas. Al definir shuffle = False estamos forzando a que las variables no informativas del conjunto de datos aparezcan al final.

* Divide el conjunto de datos en 200 observaciones para el entrenamiento y el resto para realizar la validación de los distintos métodos de regresión aplicados.
* Describe tu conjunto de datos (transfórmalo en un data.frame, aplica los métodos .info(), .describe() y obtén el histograma o diagrama de barras de todas las variables (predictoras y la variable respuesta).
* Obtén un modelo de clasificación mediante un árbol de decisión. Utiliza la función *DecisionTreeClassifier* de la librería *scikit-learn* utilizando los argumentos por defecto.
* Ahora indica que el nivel de profundidad máximo permitido es de 3. ¿Qué diferencias observas con el árbol anterior?
* Realiza los siguientes modelos de ensamble:
  + Con reemplazamiento (bagging).
  + Sin reemplazamiento (pasting).
  + Realizando Random Forest (fijando el número de nodos hoja máximo a 4).
  + GBM.
* ¿Cuál es mejor?
* Analiza la importancia de las variables de cada uno de estos métodos.
* No olvides añadir las referencias utilizadas para la elaboración del trabajo al final de la memoria.

**Extensión** **máxima**: 20 páginas, con fuente Calibri 12 e interlineado 1,5.

**Rúbrica:**

|  |  |  |  |
| --- | --- | --- | --- |
| Resolver un problema de clasificación | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Criterio 1 | El conjunto de datos y su descriptivo son correctos. | 2 | 20% |
| Criterio 2 | El modelo de clasificación mediante árboles de decisión es correcto. | 2 | 20% |
| Criterio 3 | Se aplican correctamente las técnicas de ensamble. | 3 | 30% |
| Criterio 4 | La importancia de las variables se muestra correctamente. | 2 | 20% |
| Criterio 5 | Presentación y formato de la memoria de la práctica. Bibliografía. | 1 | 10% |
|  |  | **10** | **100 %** |