Práctica/Reto 1

1. Enunciado

En esta práctica se proporciona un conjunto de datos etiquetados y se quiere diseñar una máquina que estime las etiquetas de casos nuevos que lleguen en el futuro.

Los datos provienen de hospitales donde se han tratado pacientes con una enfermedad grave y muy difícil de diagnosticar. Se pretender acelerar el diagnóstico aprendiendo a clasificar los pacientes según la información recogida sobre ellos.

Los datos proporcionados son:

`retoML1_X_train.csv`	Datos para entrenar el sistema
`retoML1_Y_train.csv`	Etiquetas de los datos de entrenamiento
`retoML1_X_test.csv`	Datos para la competición (casos nuevos en el futuro)

Al abrir las tablas X en una hoja de datos o un visor CSV observamos que hay numerosos valores perdidos, marcados por el símbolo ?

Por otro lado, al abrir la tabla Y podemos ver que hay dos clases: `nockd` (paciente sin afección) y `ckd` (paciente con afección). A la vista de esto nos planteamos la siguiente línea de trabajo:

Tareas

1. Escribir un informe con una descripción de cada atributo de la columna que sea informativa y útil.
2. Procesar los datos para tratar los valores perdidos. Para ello se deben considerar las siguientes cuestiones:
   - ¿Eliminar ese ejemplo porque tiene valores perdidos?
   - ¿Eliminar ese atributo porque tiene valores perdidos?
   - ¿Imputar los valores perdidos?

   A la salida de este paso consideramos que los atributos ya son características.
3. Procesar las características para decidir si se elimina alguna de ellas.
4. A partir de este punto, se sugiere probar diferentes procesados seguidos del clasificador lineal.
   Por ejemplo diferentes aumentados de dimensión, o aumentados seguidos de reducción mediante PCA.
5. Comparar el rendimiento de todas las máquinas obtenidas con curvas ROC, matrices de confusión y métricas F1-Score.
6. Almacenar todos los objetos que componen la máquina.
   Se puede usar el paquete `Pickle` u otro similar.
7. En el segundo fichero se deben cargar todos los objetos guardados que componen nuestra máquina final y ejecutarlos con los datos X de test proporcionados. Las etiquetas predichas deben guardarse en un fichero CSV con el mismo formato que `retoML1_Y_train.csv`

Es obligatorio:

- Realizar cada una de las tareas marcadas en este guión.
- Que el modelo de estimación de etiquetas sea lineal.
- Escribir comentarios en el código.
- Entregar 2 ficheros, uno para el entrenamiento y otro para el test. El segundo debe cargar los objetos necesarios para transformar las X y pasarlas al clasificador que devolverá la estimación de Y.

Está prohibido:

- Utilizar técnicas de reducción de la dimensionalidad como LLE, Isomap o t-SNE que NO se han explicado.
- Entrenar un clasificador que no sea lineal como árboles de decisión, SVM con kernels, Redes Neuronales, etc.
- Evidentemente, presentar una práctica idéntica a la de otro grupo.

2. Condiciones de entrega

- El equipo debe estar formado por 2 alumnos.
- La entrega debe ser un archivo comprimido ZIP que contenga:
  1. Un fichero `nombres.TXT`, con el nombre de los alumnos del grupo
  2. El código de Python que utiliza el conjunto de entrenamiento y sus etiquetas y crea la máquina pedida. Puede ser un script de python (extensión .py) o un cuaderno de Jupyter Notebook (estensión .ipynb)
  3. El código de Python que utiliza el conjunto de test la máquina pedida y genera las etiquetas de dicho conjunto. También puede ser un script de Python o un cuaderno de Jupyter Notebook pero diferente.

     Este segundo código debe funcionar bien en un ordenador diferente que sólo tenga, en una carpeta con un nombre distinto, el fichero `retoML1_X_test.csv` y el fichero con el código. Para comprobar que todo está correcto se sugiere copiar ambos en un ordenador diferente con Python y los paquetes necesarios instalados y ejecutarlo.
     Este proceso lo repetirá el profesor en su ordenador. Si no funciona la práctica tiene 0. Es más importante que funcione a que acierte el 100 %
  4. Un fichero con las etiquetas generadas que debe cumplir obligatoriamente lo siguiente:
     - **Nombre.** Es obligatorio que se llame `retoML1_Y_test.csv`
     - **Formato.** Es obligatorio que sea el mismo formato de `retoML1_Y_train.csv`; es decir una etiqueta por línea.
  5. **Memoria.** Un documento en PDF que tenga:

     Portada con el nombre de la universidad, grado, asignatura y alumnos.

     Secciones una por cada apartado de este guión, respondiendo a las preguntas que se hacen.
- La **fecha límite** para subir el fichero ZIP es la que indica en la actividad de entrega en el **aula virtual**.

3. Valoración de la práctica

Se valorarán los siguientes puntos:

- Cumplir todos los requisitos de entrega (esto no da puntos pero sí los quita). Comprueba todo con el siguiente checklist:
  - [x] Nombre de los alumnos del grupo en un fichero TXT
  - [x] Nombre correcto del fichero de etiquetas predichas
  - [x] Formato correcto del fichero de etiquetas predichas
  - [x] Memoria en PDF
  - [x] Código fuente
  - [x] Todo empaquetado en un fichero ZIP
- Los comentarios añadidos en el código.
- El proceso seguido con los datos para entrenar.
- Las características construidas.
- Un código (extra) capaz de cargar de un fichero un sistema completo listo para funcionar con datos nuevos de características similares a los dados.
- La calidad de la programación; es decir si se han creado clases, se han definido funciones o es todo un único script. También se tendrá en cuenta la silimitud con el código de clase.