APRENDIZAJE AUTOMÁTICO

PROYECTO FINAL:

Fecha límite de entrega: 14 de junio 2021

Valoración: 25 puntos

NORMAS DE DESARROLLO Y ENTREGA DE TRABAJOS

Para este trabajo como para los demás es obligatorio presentar un informe escrito con sus valoraciones y decisiones adoptadas en el desarrollo de cada uno de los apartados. Incluir en el informe los gráficos generados. También deberá incluirse una valoración sobre la calidad de los resultados encontrados (obligatorio en pdf). **Sin este informe se considera que el trabajo NO ha sido presentado.**

Normas para el desarrollo de los Trabajos: EL INCUMPLIMIENTO DE ESTAS NORMAS SIGNIFICA PERDIDA DE 2 PUNTOS POR CADA INCUMPLIMIENTO.

- El código se debe estructurar en un único script python con distintas funciones o apartados, uno por cada ejercicio/apartado de la práctica.
- Todos los resultados numéricos o gráficas serán mostrados por pantalla, parando la ejecución después de cada apartado. No escribir nada en el disco.
- El path que se use en la lectura de cualquier fichero auxiliar de datos debe ser siempre "datos/nombre_fichero". Es decir, crear un directorio llamado "datos"dentro del directorio donde se desarrolla y se ejecuta la práctica.
- Un código es apto para ser corregido si se puede ejecutar de principio a fin sin errores.
- NO ES VÁLIDO usar opciones en las entradas. Para ello fijar al comienzo los parámetros por defecto que considere que son los óptimos.
- El código debe estar obligatoriamente comentado explicando lo que realizan los distintos apartados y/o bloques.
- Poner puntos de parada para mostrar imágenes o datos por consola.
- ENTREGAR EL CODIGO FUENTE Y COPIA DE LOS FICHERO DE DATOS USADOS SI LOS DATOS ORIGINALES SE HAN MANIPULADO A MANO Y EL CÓDIGO NO RECOGE ESTOS CAMBIOS (Registros anulados, variables perdidas, etc)
- **Forma de entrega**: Subir todos los ficheros .py y .pdf a PRADO. SE INDICARÁ EL PROCEDIMIENTO.

1. Ajuste del mejor modelo

Este proyecto se focaliza en el ajuste y selección del mejor predictor para una BBDD (UCI o Kaggle) con el apoyo de la libreria Scikit-Learn. Esta librería contiene funciones de muy alto nivel que pueden ser muy útiles si se comprende bien su funcionamiento. Por tanto, las funciones que se usen de Scikit-Learn deben de justificarse y explicar con que objetivo se usan y el significado de todos sus parámetros. Los valores fijados por defecto en la librería no se consideran elecciones justificadas a priori. Decisiones sin justificación y resultados sin interpretación no serán considerados válidos.

Es obligatorio comparar hipótesis de al menos DOS de los siguientes modelos y un modelo lineal para poder optar a la puntuación total.

- **Perceptron Multicapa**. Considerar arquitecturas de 3 capas y un número de unidades por capa en el rango 50-100. Considerar el número de neuronas por capa como un hiperparámetro.
- **Máquina de Soporte de Vectores (SVM)**: se recomienda el núcleo RBF-Gaussiano o el polinomial. Encontrar el mejor valor para los parámetros libres hasta una precisión de 2 cifras (enteras o decimales) (hacer una búsqueda dicotómica)
- **Boosting**: Se recomienda que para clasificación se usen funciones "stump". Para regresión se recomiendan los árboles como regresores simples, justificando el valor del parámetro de aprendizaje.
- **Random Forest**: Usar como hiperparámetros los valores que por defecto se dan en teoría y experimentar para obtener el número de árboles adecuado.
- **Red de Funciones de base Radial**: Fijar el valor del número de núcleos a usar, K, y usar el algoritmo de k-medias para estimar sus localizaciones. Evaluar distintos valores de K como criterio para la elección del K final.

Buscar el mejor modelo posible para la base de datos seleccionada y justificar cada uno de los pasos dados para conseguirlo. En regresión, se recomienda usar penalización LASSO en el proceso de selección de variables. También pueden usarse técnicas de reducción de dimensionalidad, ej. PCA o Random Projection si se justifica que su uso mejora los resultados obtenidos. Todos los proyectos deben justificar los siguientes apartados:

1. Definición del problema a resolver y enfoque elegido. Identifique el uso de los datos en el ajuste y valoración de resultados
2. Codificación de los datos de entrada para hacerlos útiles a los algoritmos.
3. Valoración del interés de la variables medidas para el problema y selección de un subconjunto (en su caso).
4. Necesidad de la normalización de los datos e interés de la técnica usada (en su caso)
5. Interés y justificación de la función/es de pérdida usada.
6. Argumentos a favor de la idoneidad de los modelos seleccionados para la BBDD.
7. Argumentar sobre la idoneidad de la regularización usada (en su caso)
8. Algortimo de aprendizaje usado en cada modelo, especificando y justificando los valores de todos los parámetros e hiperparámetros usados.
9. Selección de la mejor hipótesis. Justifique la técnica usada y calcule el error E_{out} de dicha hipótesis.

10. Valoración de los resultados (gráficas, métricas de error, análisis de residuos, etc)
11. Argumente que se ha obtenido la mejor de las posibles soluciones para la muestra dada. Argumentar en términos de los errores de ajuste y generalización.

Puntuaciones

1. **Hasta 15 puntos**. Aquellos proyectos que solo comparen una técnica frente a un modelo lineal.
2. **Hasta 25 puntos**. Aquellos proyectos que comparen dos técnicas y un modelo lineal.
3. **BONUS**: A quienes habiendo alcanzado más de 20 puntos en la parte obligatoria, 5 puntos adicionales por cada técnica adicional incluida en la comparación y correctamente ajustada.

CRITERIOS PARA DATOS PERDIDOS: No todos los algoritmos de aprendizaje necesitan criterios para sustituir los datos perdidos. En caso de necesitarse usar los siguientes:

- Cuando una muestra de datos tengan más del 10 % de sus datos perdidos puede eliminarse del conjunto de datos si no afecta al tamaño del conjunto de datos. Si es más del 20 % debe eliminarse.
- Los datos perdidos que de variables reales se sustituiran por la suma del valor medio de la variable más un valor aleatorio en el intervalo $[-1{,}5\sigma, 1{,}5\sigma]$ siendo σ la desviación típica de la dicha variable.
- Los datos perdidos que sean de variables categóricas aplicar lo siguiente:
  - Calcular la distribución de probabilidad de las categorías en la columna, se obtiene una multinomial sobre las categorias.
  - Calcular la distribución acumulada, sumando iterativamente la probabilidad de cada categoría. Tendreis un vector que toma valores crecientes de 0 a 1, y que además define intervalos de valores asignados a cada categoria.
  - Sortear un número aleatorio uniforme en [0,1]
  - Asignar la categoria que corresponda a ese valor dentro de [0,1].

  Aplicar el criterio de que el extremo inferior de cada intervalo es un valor de dicho intervalo y el extremo superior del siguiente intervalo.

CRITERIOS PARA CLASES DESBALANCEADAS

- Si el tamaño de la clase más pequeña es suficiente para aprender. Puede aplicar ponderación de clases en el función de error. Otras técnicas disponibles en Scikit-Learn pueden usarse si se justifica su mayor idoneidad para el problema.
- También puede usarse un enfoque de partición del conjunto datos de entrenamiento en múltiples conjuntos homógeneos para su uso con un clasificador basado en un "ensemble" de clasificadores.
- Si el tamaño de la clase no pemite aprender, entonces eliminar la clase. En este caso se debe pedir la aprobación del profesor.

El uso de resultados y enfoques existentes en la literatura sobre las bases de datos está permitido y de hecho se alienta, siempre y cuando se deja manifiestamente claro que uso se hace de dicha información/resultado y cual es la aportación del proyecto sobre la misma. En caso contrario se entenderá plagio. Incluir las referencias de la bibliografía usada.