APRENDIZAJE AUTOMÁTICO

Trabajo.3: Programación

Fecha límite de entrega: 30 de Mayo

Valoración: 12 puntos

NORMAS DE DESARROLLO Y ENTREGA DE TRABAJOS

Para este trabajo como para los demás es obligatorio presentar un informe escrito con lass 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 de cada ejercicio/apartado de la práctica se debe estructurar en Python incluyendo las funciones que se hayan definido.
- Todos los resultados numéricos o gráficas serán mostrados por pantalla, parando la ejecución después de cada apartado. El código NO DEBE escribir nada a disco.
- El path que se use en la lectura de cualquier fichero auxiliar de datos debe ser siempre "datos/nombre_fichero". Es decir, se espera que el código lea de un directorio llamado "datos", situado dentro del directorio donde 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.
- Todos los ficheros (*.py, *.pdf) se entregan juntos dentro de un único fichero zip, sin ningún directorio que los contenga.
- ENTREGAR SOLO EL CÓDIGO FUENTE, NUNCA LOS DATOS.
- **Forma de entrega**: Subir a PRADO

1. Ajuste de Modelos Lineales

Este proyecto se focaliza en el ajuste y selección del mejor predictor lineal para un conjunto de datos dado 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, para cada función de Scikit-Learn, que use, debe de explicar por qué es necesaria su uso además de explicar el significado de todos sus parámetros. Los valores fijados por defecto en la libreria no se consideran elecciones justificadas a priori. Decisiones sin justificación y resultados sin interpretación no serán considerados válidos. En todos los casos los pasos a desarrollar serán aquellos que nos conduzcan al ajuste y selección del mejor modelo y a la mejor estimación del error E_{out} de dicho modelo. Como mínimo se habrán de analizar y comentar los siguientes pasos **sobre un problema de clasificación y otro de regresión**:

1. Comprender el problema a resolver. Identificar los elementos X, Y and f del problema y describirlos en detalle.
2. Selección de la clase/s de funciones a usar. Justificar cúales y porqué.
3. Identificar las hipótesis finales que usará.
4. Si la base de datos define conjuntos de training y test, únalos en un solo conjunto y genere sus propios conjuntos de training y test. Justifique el mecanismo de partición.
5. Justifique todos los detalles del preprocesado de los datos: codificación, normalización, proyección, etc. Es decir, todas las manipulaciones sobre los datos iniciales hasta fijar el conjunto de vectores de caraterísticas que se usarán en el entrenamiento.
6. Justifique la métrica de error a usar. Discutir su idoneidad para el problema.
7. Justifique todos los parámetros y el tipo de regularización usada en el ajuste de los modelos seleccionados. Justificar la idoneidad de la regularización elegida.
8. Selección de la mejor hipótesis para el problema. Discuta el enfoque seguido y el criterio de selección usado. ¿Cúal es su error E_{out}?
9. Suponga ahora que Ud. que desea afinar la mejor hipótesis encontrada en el punto anterior usando todos los datos para entrenar su modelo final. Calcule la nueva hipótesis y el error E_{out} de la misma?. Justifique las decisiones y los criterios que aplique.

Las transformaciones no-lineales de las variables pueden definirse a partir de las potencias y productos de potencias de las variables originales, conjuntos de polinomios ortogonales, etc. Si se usan transformaciones no polinómicas de las variable, como $\log, \sqrt{()}$, sin, etc, debe justificar el interés de las mismas. **Bases de datos a usar**

- **https://archive.ics.uci.edu/ml/datasets/Superconductivty+Data**
- **https://archive.ics.uci.edu/ml/datasets/Dataset+for+Sensorless+Drive+Diagnosis**

Alternativamente estos ficheros de datos estarán disponibles también en PRADO.

Recomendación: desarrollar un código en Python lo suficientemente general que permita ser reusado, en su mayor parte, en el desarrollo del Proyecto Final. Se recomienda escribir funciones que permitan ser reusadas.