

Trabajo.3: Programación

Fecha límite de entrega: 30 de Mayo

Valoración: 20 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 de cada ejercicio/apartado de la práctica se debe estructurar en un script Python incluyendo las funciones que se hayan definido. Cada script debe incluirse en un fichero distinto.
- 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 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.
- Todos los ficheros (*.py, *.pdf) se entregan juntos dentro de un único fichero zip, sin ningún directorio que los contenga.
- ENTREGAR SOLO EL CODIGO FUENTE, NUNCA LOS DATOS.
- **Forma de entrega:** Subir el zip al Tablón docente PRADO.

1. AJUSTE DE MODELOS LINEALES

Este ejercicio se centra en el ajuste de un modelo lineal a dos conjuntos de datos dados con el objetivo de obtener el mejor predictor posible. En todos los casos los pasos a desarrollar serán aquellos que nos conduzcan al ajuste y selección del mejor modelo y a la estimación del error E_{out} del modelo final. Cómo 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.
2. Selección de las clase/s de funciones a usar. Identificar cuáles y porqué.
3. Fijar conjuntos de training y test que sean coherentes.
4. Preprocesado los datos: codificación, normalización, proyección, etc. Es decir, todas las manipulaciones sobre los datos iniciales hasta fijar el conjunto de vectores de características que se usarán en el entrenamiento.
5. Fijar la métrica de error a usar. Discutir su idoneidad para el problema.
6. Discutir la técnica de ajuste elegida.
7. Discutir la necesidad de regularización y en su caso la justificar la función usada para ello.
8. Identificar los modelos a usar.
9. Estimación de hiperparámetros y selección del mejor modelo.
10. Estimación por validación cruzada del error E_{out} del modelo. Compárela con E_{test} , ¿qué conclusiones obtiene?
11. Suponga que Ud ha sido encargado de realizar este ajuste para una empresa. ¿Qué modelo les propondría y que error E_{out} les diría que tiene?. Justifique las decisiones.

Las clases de funciones con variables no lineales pueden definirse a partir de las potencias y productos de potencias de las variables originales. Si se usan otras transformaciones de las variables iniciales, como \log , $\sqrt{}$, \sin , etc debe justificarse el interés de las mismas. **Bases de datos a usar**

- <https://archive.ics.uci.edu/ml/datasets/optical+recognition+of+handwritten+digits>
- <http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime>

Alternativamente estos ficheros de datos estarán disponibles también en la web de decsai.

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.