**ACTIVIDAD. Enunciado de la prueba:**

El alumnado deberá realizar las siguientes tareas. En todas ellas, se deberán ir importando todas las librerías y paquetes que se vayan necesitando.

1. **TAREA 1.**

- Abrir el dataset 'ds_salaries.csv' **en la variable 'df'**.
- Quedarse sólo con las 1000 primeras del dataset.
- Eliminar la columna 'salary_in_usd'.
- Codificar la variable 'job_title'.
- La variable objetivo es 'salary'.
- La variable predictora X será la columna 'job_title' codificada.
- Instanciar un nuevo modelo de regresión lineal y entrenarlo con todas las muestras de 'X' y de 'y'.
- Hacer un scatter en rojo con la nube de puntos de X frente a 'salary', y encima un plot en azul de de 'job_title' frente a las predicciones del salario para todas las muestras predictoras de X.
- Hacer lo mismo para la regresión polinómica de orden 2 y de orden 3.
- Según las figuras, indicar si estas dos variables tienen una relación lineal o polinómica (de grado 2 o 3).
- Hacer lo mismo pero en un bucle que haga los tres modelos y grafos de forma automática.

2. **TAREA 2.**

- Cargar el dataset 'iris' del paquete 'seaborn', **en la variable df_iris**.
- Fronteras de decisión para dos clases de la variable 'species': 'setosa' y 'virginica': hay que visualizar cuáles son las fronteras de decisión para un modelo entrenado mediante regresión logística a partir de dos características predictoras: 'sepal_width' y 'petal_length'. La variable objetivo es 'species'.
- Crear una función llamada 'plot_fronteras' que muestre una gráfica con las fronteras de decisión de un modelo de clasificación. Esta función recibe como argumentos:
  - ✓ El modelo utilizado para hacer las predicciones.
  - ✓ Las 2 características predictivas (en formato array de NumPy).
  - ✓ Los ejes para añadir el diagrama de dispersión con datos de entrenamiento y testeo.
- Crear una función 'mostrar_fronteras' que invoque a la función anterior y muestre sobre ella un diagrama de dispersión con los datos de entrenamiento y de validación. Según el gráfico resultante, ¿es el conjunto de datos linealmente separable?
- Calcular ahora las fronteras de decisión para las tres clases de 'species'

## 3. TAREA 3.

- Utilizar el conjunto de datos de dominio público 'vw.csv', obtenido de kaggle.com.
- Utilizar en todos los casos 'random_state=42'.
- Codificar las variables categóricas.
- La variable objetivo es ‘model’, y las predictoras son ‘transmission’ y ‘fuelType‘, **seleccionarlas mediante SQL**. Obtener las variables predictoras y objetivo para entreno y testeo (20%).
- Entrenar los siguientes algoritmos con GridSearchCV y los siguientes rangos de parámetros, quedarnos con el mejor parámetro en cada caso.
    - Árbol de decisión, max_depth = 2, 4, 6
    - K-vecinos: n_neighbors = 5, 10, 15
    - Support Vector Machines (SVC): kernel = ‘linear’, ‘poly’, ‘rbf’.
- Entrenar los dos primeros modelos con los parámetros óptimos resultantes de GridSearchCV, y añadir el modelo ‘stacking’ usando como modelos base los dos primeros modelos de esta lista, y como meta-modelo el SVC (con su parámetro óptimo calculado previamente). Obtener los 'scores' y comparar los resultados.
  **Para ello, crear un diccionario con los modelos y hacer un bucle que vaya recorriendo el diccionario**. Cada vez hay que imprimir en pantalla un mensaje de que ha comenzado el entrenamiento de cada modelo. Ir metiendo en un nuevo diccionario, como claves los nombres de los algoritmos que se van entrenando, y como valores los scores de los modelos resultantes de cada algoritmo. Al final hay que hacer un bar-plot con los acrónimos de los algoritmos en el eje x y los valores de los 'score' en el eje y. Indicar el algoritmo que consigue el mejor 'score'.