Tarea 3 INF-398

CART

1. [Clasificación] Utilizando el dataset iris de sklearn, entrene un modelo decision tree de clasificación y use la función plot_tree de sklearn para graficar la lógica del árbol. Imprima también las feature_importances_ del arbol y explique como se llegó a esos resultados.
2. [Regresión] Cree un dataset simple usando:

```
X, y = make_regression(n_samples=1000, n_features=4, noise=10,
random_state=42)
```

de la biblioteca sklearn.datasets. Al igual que en el punto 1, entrene un decision tree de regresion, grafique el plot_tree e imprima las feature_importances_ explicando como el modelo llegó a esos resultados.

3. ¿Qué variables del dataset están influyendo más en las decisiones del árbol y por qué el algoritmo determinó dichas variables como las mas importantes?
4. ¿En qué se diferencian los árboles de clasificación y los de regresión al momento de crearse? Explica los principales cambios en la estructura del árbol y el tipo de decisiones que toma.

KNN

1. Para KNN utilice el dataser iris de sklearn, realice un modelo visual con interact (from ipywidgets import interact) creando un slider para cambiar el valor de K. Encontrar el mejor valor de K con validación cruzada.
2. ¿Cómo cambia la precisión del modelo a medida que varía el valor de **K**?
3. ¿Cómo afecta el valor de **K** en el rendimiento del modelo? ¿Por qué ocurre esto?
4. Comentar las diferencias entre un modelo default y uno optimo. Hágalo tanto visualmente como con los resultados de las métricas que estime conveniente.

SVM

1. realice un modelo visual con interact (from ipywidgets import interact) para variar los hiperparámetros: kernel, C y gamma. Crear datos con:

```
X, y = make_classification(
    n_samples=300,
    n_features=2,
    n_informative=2,
    n_redundant=0,
    n_classes=2,
    n_clusters_per_class=1,
    random_state=42
)
```

2. Implemente una búsqueda manual de hiperparámetros con los siguientes valores (Utiliza dos ciclos anidados para probar todas las combinaciones posibles):

```
C_values = [
0.001, 0.003, 0.010, 0.032, 0.100, 0.316, 1.000, 3.162, 10.000, 31.622
]

gamma_values = [
0.001, 0.003, 0.010, 0.032, 0.100, 0.316, 1.000, 3.162, 10.000, 31.622
]
```

Entrena un modelo SVM con kernel por default para cada combinación de C y gamma usando un conjunto de entrenamiento y mide su rendimiento en un conjunto de validación. Usa la métrica de F1-score y Registra el rendimiento en una matriz (donde los ejes representan C y gamma).

3. Use un heatmap para visualizar los resultados de rendimiento del modelo para cada combinación de hiperparámetros. El color de cada celda debe representar el rendimiento del modelo (F1-score) y debe estar anotado el valor correspondiente. TIP: utilice espacio de color='magma' para visualizar los datos
4. ¿Existen "zonas óptimas" en lugar de valores específicos de hiperparámetros que producen un buen rendimiento? Si la respues es positiva indique dicha zona óptima
5. ¿Cómo afecta el valor de C y gamma al rendimiento del modelo SVM? Explica qué pasa cuando C o gamma son demasiado grandes o pequeños.

ENSEMBLE

1. ¿Cuál es la principal diferencia entre los modelos de bagging y boosting? ¿Que ventajas y desventajas presentan?

2. Usando el dataset https://www.kaggle.com/datasets/nikhil7280/weather-type-classification sin limpieza de datos, entrene un modelo de Random Forest, adaBoost y gradientBoost. Imprima las metricas correspondientes y mida el tiempo de ejecución en cada caso.
3. Ahora realice una limpieza de outliers quitando los percentiles inferiores al 5% y superiores al 95% de los datos. y reentrena con estos nuevos datos. Imprima las metricas correspondientes y mida el tiempo de ejecución en cada caso.
4. ¿Cómo influye la eliminación de los outliers en el rendimiento de los modelos? Explica por qué algunos modelos pueden beneficiarse más que otros de la limpieza de outliers.
5. ¿Cuál cree que es el mejor modelo para estos datos ? ¿Por qué?

KAGGLE

Enlace a competencia: https://www.kaggle.com/t/30d42b5d9eb14ec3a65a107ad29c5ad6

DATASET OVERVIEW: https://archive.ics.uci.edu/dataset/2/adult

Descripción:
El conjunto de datos "Adult" (también conocido como "Census Income") es un conjunto de datos de clasificación supervisada que se utiliza comúnmente en problemas de predicción de ingresos. El objetivo del modelo es predecir si un individuo gana más o menos de 50,000 USD al año, en función de varias características demográficas y laborales.

Características:
Este conjunto de datos contiene 14 atributos sobre individuos y su nivel de ingresos. Las características incluyen información personal como la edad, el estado civil, la ocupación, la educación y la nacionalidad, entre otras. El atributo objetivo es una variable binaria que indica si el individuo gana más de 50,000 USD al año (`>50K`) o menos de 50,000 USD al año (`<=50K`).

Formato del Conjunto de Datos:

- **Entradas**: Cada fila representa a un individuo con diversas características demográficas y laborales.
- **Objetivo**: Predecir la clase en función de las características:
  - `>50K`: Ingreso mayor a 50,000 USD
  - `<=50K`: Ingreso menor o igual a 50,000 USD

==========

Feedback opcional (3 puntos extra)

De su opinión sobre las tareas realizadas, por ejemplo, si les gustó y se entretuvieron con el formato, datasets y las competencias, si encontró que fueron de ayuda y aprendieron mientras las hacían, o si encontró que podrían mejorar y en qué podrían mejorar.

Saludos y éxito en su cierre de semestre!🙌