



Grupo nº8

ENSAMBLE SECUENCIAL DE MODELOS

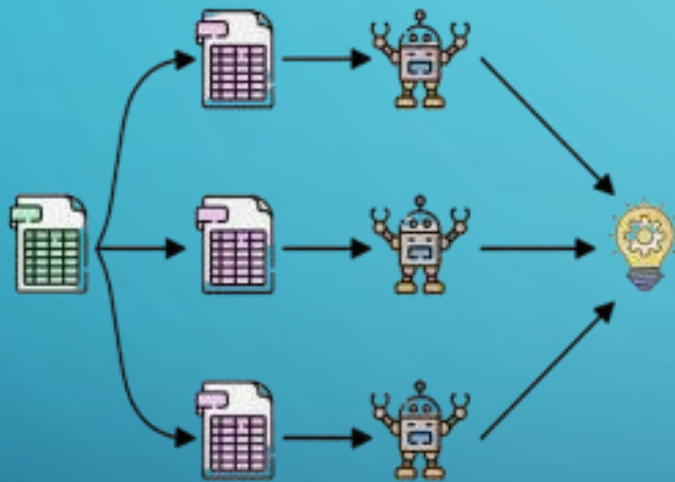
(GRADIENT BOOSTING)

Hecho por:
Ignacio Gutiérrez Serrera
Emilio Manuel Vázquez Cruz

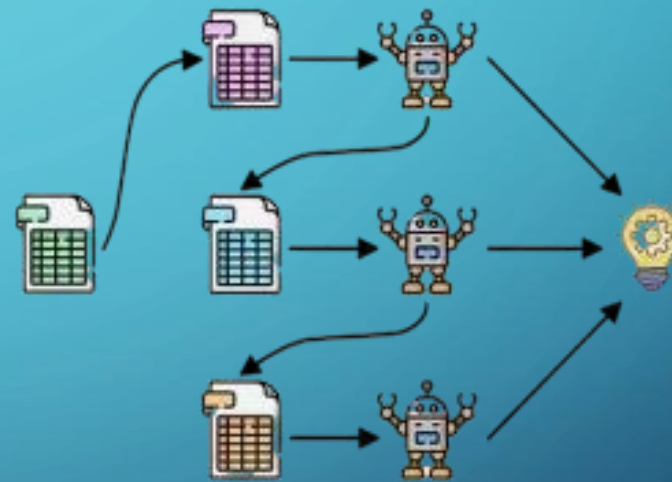
ÍNDICE

1. Introducción
2. Metodología
3. Estructura y decisiones de diseño
4. Experimentos y resultados
5. Conclusión

INTRODUCCIÓN



Parallel



Sequential

METODOLOGÍA



TRABAJO EN PAREJA

Mayormente se trabajó en pareja para asegurar un conocimiento global del trabajo y agilizar el trabajo.



USO DE GITHUB

Sistema de control de versiones distribuido utilizado para registrar los cambios y compartirlo entre ambos miembros.



DOCUMENTACIÓN

Lectura de documentación, tanto de la propuesta, el temario de la asignatura y sobre las librerías utilizadas.

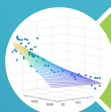
ESTRUCTURA Y DECISIONES DE DISEÑO



Uso de semillas para garantizar reproducibilidad



CSV almacenados en DataFrames



Tareas de Regresión y clasificación



Normalización



Entrenamiento con datos aleatorios y uso de R^2



Heredamos de BaseEstimator y RegressorMixin



Se inicializa la predicción actual como la media de la variable objetivo



Uso de umbral épsilon para la parada temprana

Procedimiento Ensemble secuencial (GradientBoosting)

Entrada:

- Array numpy de datos
- Array numpy de variable objetivo
- Algoritmo base de entrenamiento
- Número de estimadores
- Tasa de aprendizaje (lr)
- Tamaño de la muestra (`sample_size`)
- Paciencia (para la parada temprana)
- Épsilon (para la parada temprana)
- Tipo de la tarea (regresión o clasificación)

Salidas:

- Predicción sobre los datos de prueba del ensemble de modelos entrenado con los datos de entrenamiento.

Algoritmo:

Entrenamiento:

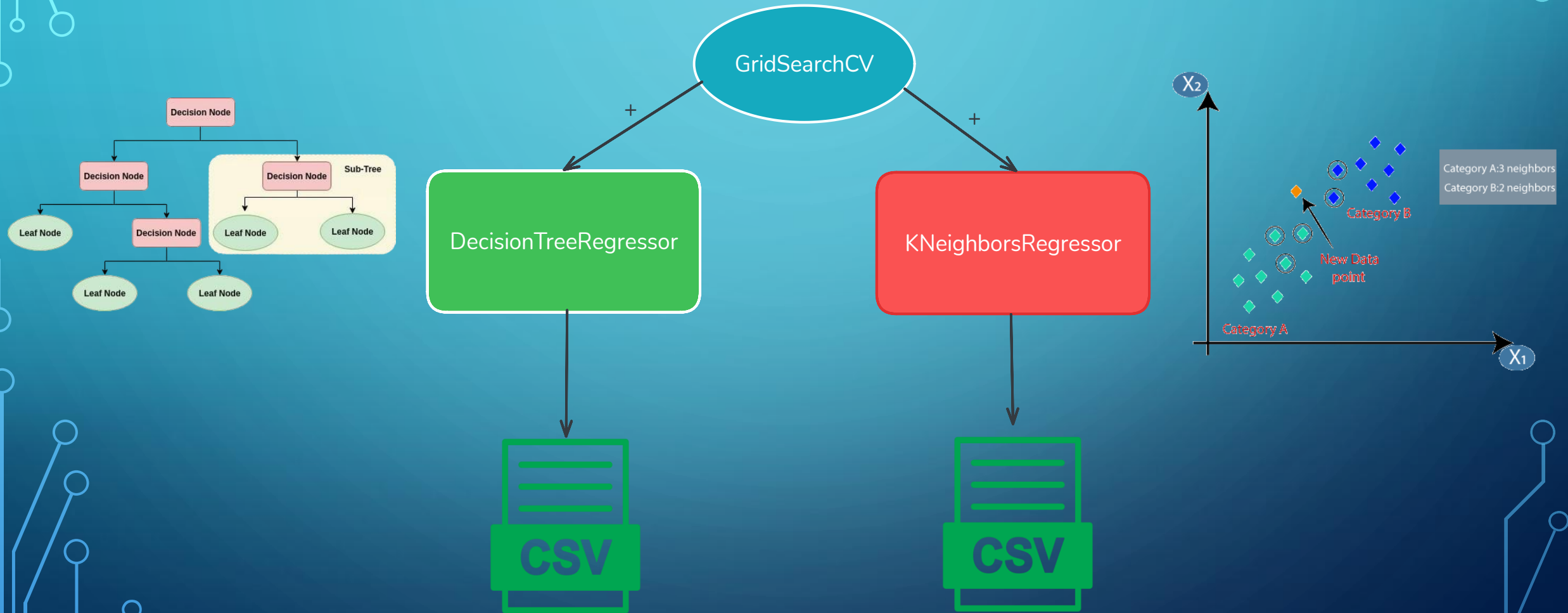
1. Inicializar la primera predicción con `pred0`
2. `pred_actual = pred0`
3. Por cada i en `n_estimators`:
 1. `residuoi = y - pred_actual`
 2. entrenar `estimadori` con un subconjunto aleatorio de los datos marcado por `sample_size` y usando `residuoi` como variable objetivo.
 3. obtener las predicciones, `predi`, de `estimadori`
 4. `pred_actual = pred_actual + predi * lr`
 5. Calcular el R^2 de la predicción del meta-modelo hasta la iteración actual y comprobar si la diferencia con el R^2 de la predicción anterior es menor que `epsilon`:
 1. Si no: no hacer nada
 2. Si: decrementar en 1 la paciencia
 6. si `paciencia = 0` entonces finalizar el bucle de cálculo de estimadores.
4. Devolver el conjunto de modelos entrenados

Predicción:

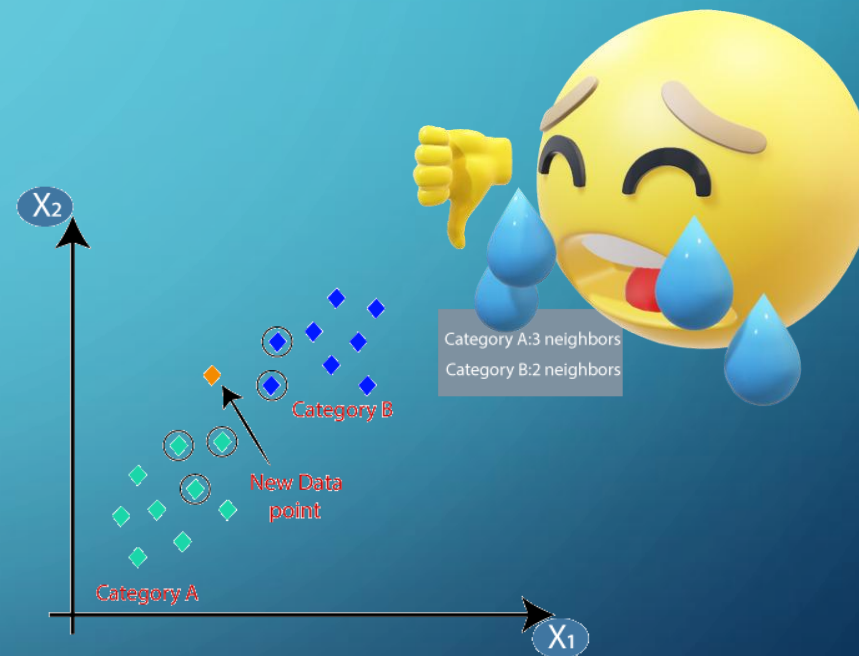
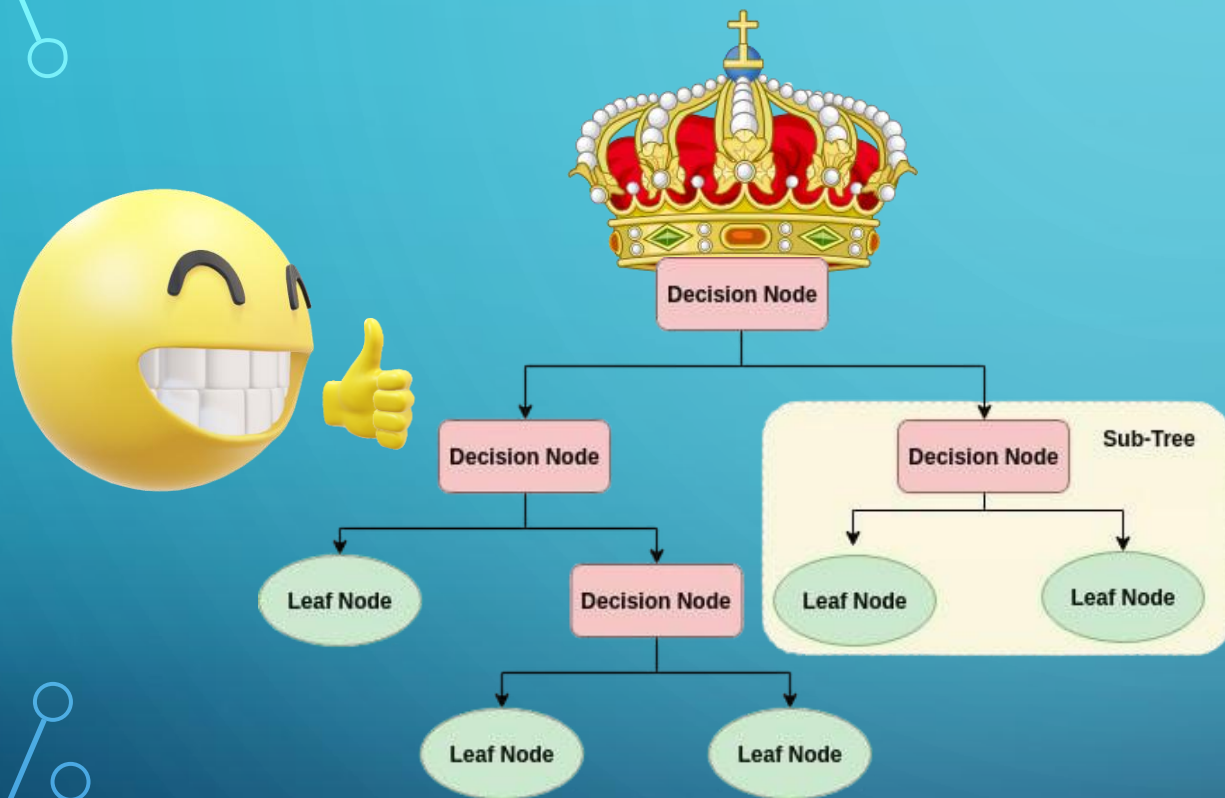
Inicializar la primera predicción como la media de la variable objetivo.

6. `pred_actual = media(array variable objetivo)`
7. Por cada `modeloj` en `modelos entrenados`:
 1. `pred_actual += lr * modelo.predict(datos)`
8. Devolver el conjunto con la predicción para cada elemento.

EXPERIMENTOS Y RESULTADOS



CONCLUSIÓN



GRACIAS POR SU ATENCIÓN

