Trabajo Práctico Integrador

Presentación de modelos predictivos a través del ensamble RandomForest

Grupo 1: Nazareno Magallanes – Lautaro Terreno – Hernán De Lorenzo – Marcelo Valeiras
14 Noviembre 2024

Indice de Temas

Presentación del caso y objetivo

- Se desea construir una librería que permita construir modelos predictivos a través de una versión simplificada del ensamble RandomForest.

- El objetivo es entrenar un modelo (árboles de decisión) con algún conjunto de datos etiquetados (aprendizaje supervisado) de forma que permita predecir la variable objetivo.

- Nos enfocaremos en problemas de clasificación

- Datasets utilizados: cancer_patients (Kaggle)
  PlayTennis (Kaggle)
  iris (sklearn datasets)

Organización del grupo y herramientas de trabajo

TP Final - Algo 2

View 1 | New view

Filter by keyword or by field

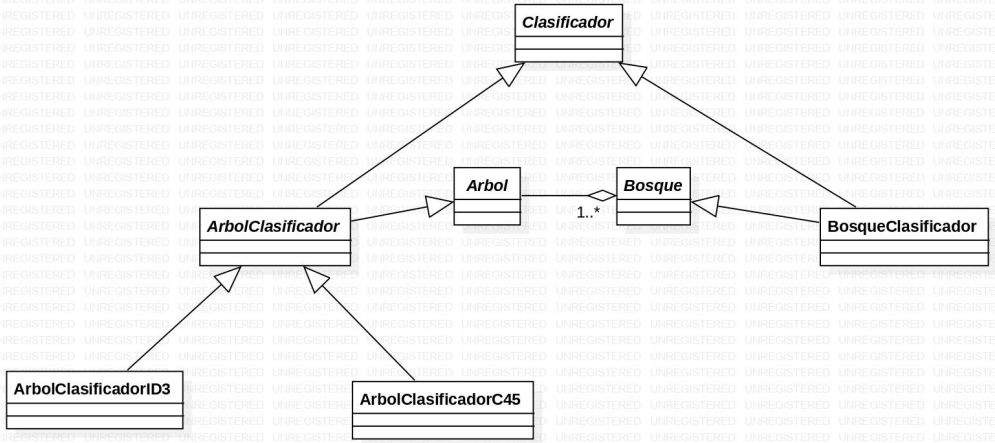
	Title	Assignees	Status	Prioridad
1	Separar clase ArbolDecision en Arbol y AlgoritmoDecision #2	lauterre	Done	Alta
2	Crear esqueleto C4.5 #5	hdelorenzo and naz...	Done	Alta
3	Crear funcion copy #6	lauterre	Done	Media
4	Arreglar funcion predict #7	nazarenomm	Done	Alta
5	Agregar funciones de arboles #8	hdelorenzo	Done	Media
6	Sacar metodos abstractos para darle funcionalidad #9	lauterre	Done	Media
7	Crear ClasificadorBosque #10	lauterre	Done	Alta
8	Ajustar clases, metodos, archivos y demas #11	Chelo78 and nazare...	Done	Baja
9	Dejar de apoyarnos en pandas y usar arrays de numpy #12	nazarenomm	Todo	Baja
10	Agregar Hiperparámetros #13	Chelo78	Done	Alta
11	Arreglar condiciones de parada #16	nazarenomm	Done	Alta
12	Agregar Excepciones #18	hdelorenzo	In Progress	Baja
13	Agregar clase Graficador #22	lauterre and nazare...	Done	Baja
14	Poda - Reduced Error Pruning #23	Chelo78	Done	Media
15	Poda - Rule Post Pruning #24	nazarenomm	In Progress	Media
16	Agregar metricas #26	nazarenomm	Done	Media
17	Hacer tests #28		Todo	Baja

Proyecto de github

Comunicación constante

Peer coding

Decisiones de diseño (diagrama de clases)



Clases auxiliares:

- Graficador
- Herramientas
- Impureza
- Metricas
- Hiperparámetros

Presentación del árbol de decisión modelo ID3

El modelo ID3 es una estructura de árbol donde cada nodo interno representa una *pregunta* sobre una característica (atributo o feature), cada rama representa el resultado de la pregunta, y cada nodo hoja representa la clasificación.

- Sólo puede manejar atributos categóricos.

El entrenamiento de este modelo consiste en construir la estructura del árbol a través de estos pasos:

1. Selección del atributo.
2. División del conjunto de datos.
3. Construcción del árbol.
4. Podado del árbol.

Presentación del árbol de decisión modelo ID3

- Predicciones
  Una vez construido el árbol de decisión, se recorre el árbol desde la raíz hasta la hoja.
  1. Iniciar en la raíz.
  2. Evaluar la pregunta del nodo.
  3. Seguir la rama apropiada.
  4. Repetir hasta llegar a una hoja.

Entropía y Ganancia de información

La entropía de Shannon nos indica un nivel de impureza de los datos en dicho conjunto respecto a la variable objetivo que toma el valor de una clase.

La ganancia de información para un atributo se calcula como la diferencia entre la entropía del conjunto de datos antes de la división y la entropía ponderada del conjunto de datos después de la división:

Presentación del árbol de decisión modelo C4.5

- El modelo de árbol de decisión C4.5 es una extensión del algoritmo ID3.

C4.5 puede manejar tanto atributos categóricos como continuos.
Para los atributos continuos, se busca el umbral que maximiza la ganancia de información, dividiendo así el atributo en un conjunto de valores discretos.

Mejoras respecto al ID3:

- Gain Ratio - Split info
- Imputación de NA
- Costos asimétricos
- Datos ponderados

Presentación del árbol de decisión modelo C4.5

Criterio de división: En lugar de usar solo la ganancia de información como criterio de división, C4.5 utiliza una métrica llamada **Gain Ratio.**

- **Gain Ratio** corrige la tendencia de la ganancia de información a favorecer atributos con muchos valores posibles. Penaliza la división de atributos con muchos valores distintos, lo que ayuda a evitar el sobreajuste.

- Se introduce un nuevo concepto llamado **split info** para calcular el gain ratio, que mide la dispersión de los valores del atributo A en el conjunto de datos.

- **Gain ratio** se calcula utilizando el mismo **information gain** que utiliza ID3.

- El atributo con el **mayor Gain ratio** se selecciona como el mejor atributo para dividir el conjunto de datos.

Presentación del ensamble Random Forest

- Bagging (Bootstrap Aggregating)
- Combinación de modelos para producir una predicción.

Random Forest construye un conjunto de árboles de decisión utilizando un subconjunto aleatorio de muestras y un subconjunto aleatorio de atributos.
Se incluye la utilización de hiperparámetros:

- Selección de algoritmo de árbol: ID3, C4.5.
- Criterios de parada: profundidad máxima del árbol, número mínimo de observaciones por nodo, ganancia de información mínima, número mínimo de observaciones por hoja.
- Cantidad de estimadores del ensamble (árboles del bosque).

Pasamos a la Demo

Dificultades encontradas y soluciones implementadas

- Decisiones de diseño:
  - C4.5 heredaba de ID3
  - Clases
- Apoyarnos en Pandas en lugar de Numpy
- Uso del graficador
- Funcionalidad complicada de implementar

Conclusiones y mejoras

- Logramos construir una librería que construye modelos predictivos del ensamble RandomForest.
- Aseguramos la extensibilidad para poder implementar diferentes algoritmos de árboles de decisión.
- Aplicamos los conocimientos adquiridos durante la cursada en el proyecto.

Mejoras:

- Dejar de apoyarnos en Pandas.
- Implementar funcionalidades faltantes (Post pruning, datos ponderados, costos asimétricos).
- Incluir manejo de excepciones.
- Realizar testeo exhaustivo con el uso de pytest.
- Refactorización de código para lograr una mejor eficiencia.
- Agregar métricas.
- Implementar el algoritmo CART.

“GRACIAS POR SU ATENCION”