

UNIDAD TEMÁTICA 6: Ensamblés de Algoritmos

Trabajo de Aplicación 1

ESCENARIO

Ejercicio 1 Votación

Para implementar un clasificador de ensamble, se puede comenzar construyendo un proceso de base de clasificación simple.

Por ejemplo, podemos construir un proceso de árbol de decisión para el dataset Iris.

El proceso estándar con un árbol de decisión tendría un operador “decision tree” y un “retrieve”, y luego se aplicaría el modelo a datos no vistos y se evaluaría el rendimiento con un operador “performance evaluation”.

Para hacer un modelo de ensamble:

- Inserta un operador “retrieve” y asócialo al dataset Iris
- Inserta un proceso “Split Validation”, conecta los puertos
- Dentro del “Split Validation”, en la sección del modelo inserta un operador “Vote”, que es también un subproceso. En la sección derecha debe haber un operador “Apply Model” y un “Performance Evaluation”
- Dentro del “Vote”, agrega tres operadores que serán los que colaboren, los “modelos base”: “Decision Tree”, “k-nn” y “Naive Bayes” (todos ya estudiados anteriormente)
- Discute las configuraciones correspondientes para los modelos base, documenta los valores que decida el equipo para los diferentes parámetros.
- Ejecuta el proceso y analiza los resultados que se despliegan.
- Modifica el proceso principal para entrenar en forma independiente (pero con los mismos datos) modelos “Decision Tree”, “k-nn” y “Naive Bayes”. Configura sus parámetros de manera igual a lo hecho con los operadores análogos en el ensamble
- Ejecuta el proceso y realiza una tabla comparativa con todos los valores de resultados, de los modelos básicos independientes y del ensamble.

Ejercicio 2 Bagging

Similar al modelo "Vote" en RapidMiner, el operador "Bagging" actúa como un modelo con múltiples modelos base dentro de él.

- En un nuevo proceso, incorpora el dataset Iris y un operador "Split Validation"
- Dentro del "Split Validation", en la sección de modelo agrega un operador "Bagging",
- Dentro del Bagging (que es un proceso), inserta un "Decision Tree" y configura sus parámetros por defecto
- El operador "Bagging" toma dos parámetros (deja los valores por defecto de 0.9 y 10):
 - Tasa de muestreo
 - Iteraciones
- Ejecuta el proceso y observa los resultados. Podrás examinar todos los **m** modelos generados (en este caso 10) de acuerdo al parámetro "iteraciones".
- Los resultados de los modelos base se integran utilizando votación sencilla.
- Documenta los resultados. Analiza comparativamente estos resultados con los obtenidos en el Ejercicio 1.

Ejercicio 3 Random Forest

Este algoritmo funciona de manera similar a los otros modelos de ensambles; el usuario tiene que especificar la cantidad de árboles de base a ser construidos.

Como el modelo interno de base es siempre un árbol de decisión, no tiene una especificación de subproceso interna.

Todos los parámetros específicos de los árboles se configuran a nivel del operador RandomForest.

El parámetro clave que especifica el número de árboles de base es *Number of Trees*.

Realización:

- En un nuevo proceso, incorpora el dataset Iris y un operador “Split Validation”
- Dentro del “Split Validation”, en la sección de modelo agrega un operador “Random Forest”,
- Configura los parámetros del Random Forest
 - Documenta los valores que eliges y justifícalos
 - Se sugiere comenzar con 10 árboles para observar los resultados.
- Ejecuta el proceso y observa los resultados. Podrás examinar todos los **m** modelos generados (en este caso 10)
- Los resultados de los modelos base se integran utilizando votación sencilla.
- Documenta los resultados. Analiza comparativamente estos resultados con los obtenidos en el Ejercicio 1.

Remite toda la tabla comparativa, producto de los 3 ejercicios, a la tarea de la webasignatura, junto con todos los modelos desarrollados.