Comparación de decision trees entre RapidMiner y Weka.

Se utiliza el dataset Iris y las herramientas RapidMiner y Weka para hacer una comparación de la performance de sus modelos de árboles de decisión midiéndola con un Split de datos de un 70% y un 30% para entrenamiento y test respectivamente en ambas.

RapidMiner

Tipo de problema. Clasificación y regresión.

Algoritmo base. C4.5.

Características requeridas de atributos y label. Las variables de entrada pueden ser numéricas o nominales. Se exige una variable objetivo nominal para clasificación y numérica para de regresión.

Parámetros.

- Criterion: gain_ratio. Selecciona el criterio que se usará para seleccionar los atributos sobre los que hacer los splits.
- Maximal Depth: 10. La máxima profundidad del árbol.
- Apply pruning: Activado. Si se aplica pruning o no.
- Confidence: 0.1. Nivel de confianza usado para el error pesimista del cálculo de pruning.
- Apply prepruning: Activado. Si se aplica prepruning o no.
- Minimal gain: 0.01. La ganancia de un nodo se calcula antes del Split. Se hace el Split si la ganancia es mayor al minimal gain.
- Minimal leaf size: 2. Tamaño mínimo de observaciones por hoja.
- Minimal size for split: 4. El tamaño de un nodo es el número de ejemplos en él. Solo se hace el Split para obtener nodos con un tamaño mayor al minimal size for Split.
- Number of alternatives for pruning: 3. Numero de nodos alternativos testeados para un Split cuando el prepruning previene un Split.

Weka

Tipo de problema. Clasificación y regresión.

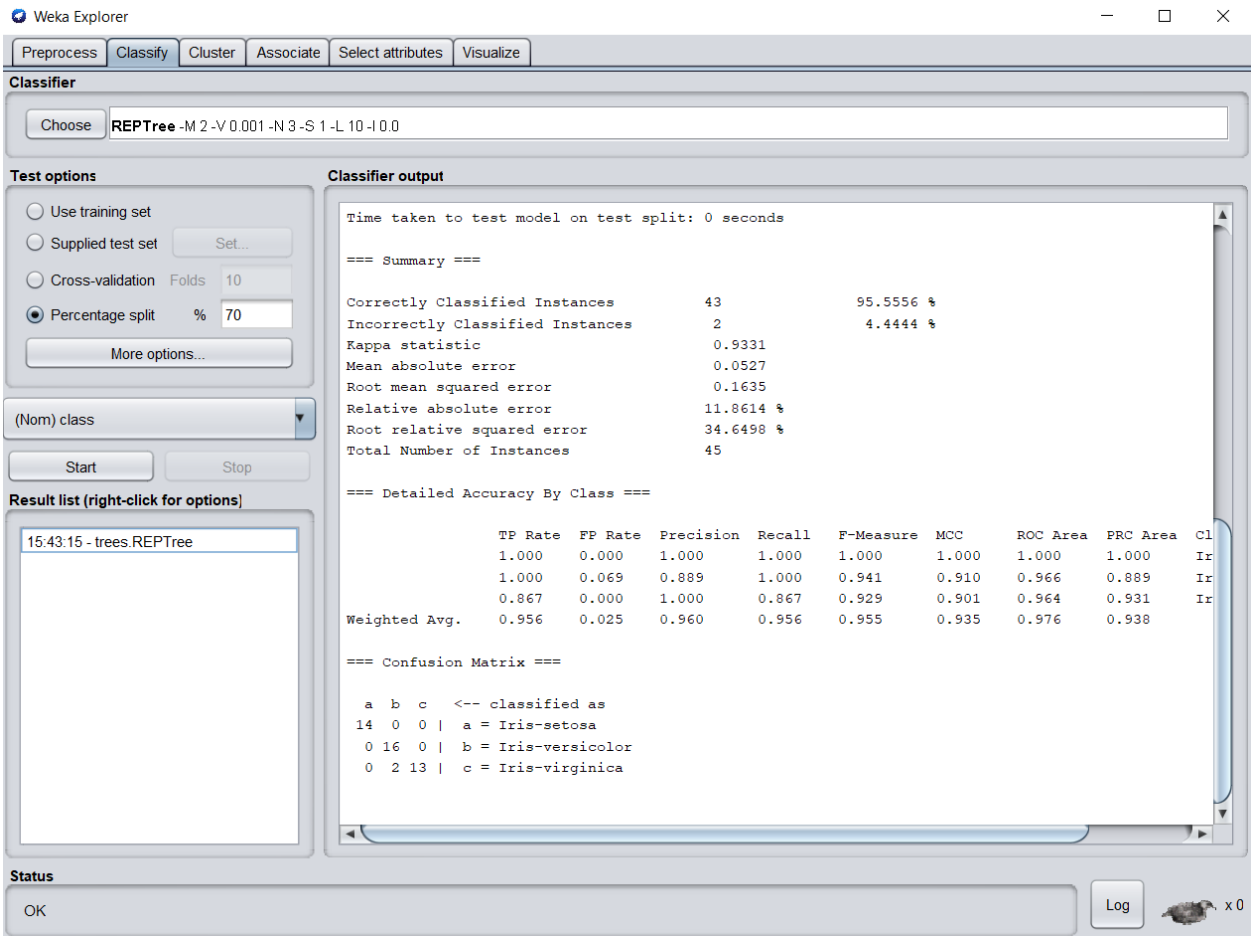
Algoritmo base. C4.5.

Características requeridas de atributos y label. Las variables de entrada pueden ser numéricas o nominales. Se exige una variable objetivo nominal para clasificación y numérica para de regresión.

Parámetros.

- batchSize: 100. Numero de instancias a procesar si la predicción batch está siendo realiza.
- Debug: False. Si es verdadero el clasificador podría poner info adicional como salida en consola.
- doNotCheckCapabilities: False. Si es verdadero las capacidades del clasificador no son checkeadas antes de la compilación.
- initialCount: 0.0. Valor inicial del contador de la clase.
- MaxDepth: 10. Máxima profundidad del árbol, con -1 no hay restricción.

- MinNum: 2.0. Mínimo peso total para las instancias de una hoja.
- minVariancePrep: 0.001. Mínima proporción de la varianza de todos los datos que necesita estar en un nodo para que se haga un Split.
- noPruning. False. Si se realiza pruning.
- numDecimalPlaces. 2. Número de posiciones decimales para usar en la salida del modelo.
- numFolds: 3. Determina el tamaño de los datos usados para pruning.
- Seed. 1. La semilla usada para la aleatoriedad de los datos.
- spreadIntialCount. False. Distribuir el recuento inicial en todos los valores en lugar de utilizar el recuento por valor.
-



Resultados

Se obtuvo una performance de un 100% en el caso de RapidMiner y un 96% en Weka utilizando los parámetros que se muestran en la sección anterior para cada uno.

Otras herramientas.

Azure Machine Learning Studio

Tipo de problema. Clasificación y regresión.

Algoritmo base. Se utiliza una versión mejorada de los árboles de decisión con una gradient boosting machine.

Características requeridas de atributos y label. Las variables de entrada pueden ser numéricas o nominales. Se exige una variable objetivo nominal para clasificación y numérica para de regresión.

Parámetros.

- Maximum number of leaves per tree. Número máximo de hojas del árbol.
- Minimum number of samples per leaf node. Mínimo número de ejemplos en un nodo hoja.
- Learning rate. Ritmo de aprendizaje.
- Total number of trees constructed. Número de árboles construidos al entrenar el algoritmo.
- Random number seed. Semilla de entrenamiento.
- Allow unknown categorical levels. Seleccionado crea un nuevo nivel para cada atributo categórico.

KNIME

Tipo de problema. Clasificación.

Algoritmo base. C4.5.

Características requeridas de atributos y label. Las variables de entrada pueden solo ser numéricas o nominales. La variable objetivo solo puede ser nominal.

Parámetros.

- Class columna. Selecciona la variable objetivo.
- Quality measure. Para seleccionar la medida de calidad para la cual se calcularan los Splits. Las opciones son Gini index y Gain Ratio.
- Pruning method. Pruning reduce el tamaño del árbol y evita el overfitting.
- Reduced error pruning. Si se checkea se usa un simple método de pruning.
- Min number records per node. Mínimo numero de registros requeridos por nodo.
- Number records tos toe for view. Selecciona el número de registros guardados en un tree para la view.
- Average Split point. Al checkearla el valor para Split con atributos numéricos se determina según la media de los valores que separan a las dos particiones.
- Number threads. Permite multiprocesamiento.
- Skip nominal columns without domain information. Seleccionada las columnas nominales que no información de valores de dominio se saltean.
- Force root split column. Seleccionada, el primer split es calculado en la columna elegida sin evaluar ninguna otra para posibles splits.
- Binary nominal splits. Al seleccionarla a los atributos nominales se les hacen splits binarios.
- Max nominal. Número máximo de valores nominales.

- Filter invalid attribute values in child nodes. Habilitando esta opción se hace un post procesamiento del tree y se filtran checkeos inválidos.
- No true child strategy: Opciones para cuando el valor de los atributos de un nodo es desconocido.
- Missing value strategy. Opciones para los valores faltantes.

Python Sci kit learn

Tipo de problema. Clasificación y regresión.

Algoritmo base. CART

Características requeridas de atributos y label. Las variables de entrada pueden solo ser numéricas. La variable objetivo puede ser nominal o numérica.

Parámetros.

- Criterio. Función para medir la calidad del Split. Puede ser Gini o entropy.
- Splitter. Estrategia utilizada para elegir el Split en cada nodo. Las opciones son mejor o random.
- Max Depth. Máxima profundidad del árbol.
- Min samples. Mínimo de ejemplos requeridos para hacer un Split generando un nodo interno.
- Min samples leaf. Mínimo de ejemplos requeridos para hacer un Split generando un nodo hoja.
- Min weight fraction. Fracción de peso mínimo del total de peso requerido en un nodo hoja.
- Max features. Máximo número de variables de entrada considerado para hacer el mejor Split.
- Random_state. Controla la aleatoriedad del estimador.
- Max_leaf_node. Máximo número de nodos hoja.
- Min_impurity_decrease: Se le hará un Split a un nodo si el Split da un decenso de la impureza mayor o igual a este valor.
- Class_weight: Pesos asociados a las clases.
- Ccp_alpha: Parámetro de complehidad usado para el pruning de mínimo costo-complejidad.