Práctica 4

Clasificación con redes bayesianas

Febrero de 2017

Modelos Gráficos Probabilísticos

Curso 2016/2017

Contents

1 Introducción al guión

Para aprender y clasificar redes bayesianas, utilizaremos el software Weka. Se trata de un conjunto de librerías JAVA para la extracción de conocimientos desde conjuntos de datos. Es un software desarrollado en la universidad de Waikato (Nueva Zelanda) bajo licencia GPL lo que ha permitido que sea una de las suites más utilizadas en el área en los últimos años. R dispone de una librería para trabajar con algoritmos implementados en Weka, pero por el momento su uso está limitado a líneas de comandos. Por razones didácticas usaremos el entorno gráfico.

El software lo puedes encontrar en `http://www.cs.waikato.ac.nz/ml/weka`
En *swad* se puede encontrar un manual del software completo, para la versión 3.7.1. aunque en la sesión se van a seguir los capítulos 4 y 8, dedicado al Explorer. Este módulo de Weka, permite visualizar, tratar y aplicar distintos algoritmos de aprendizaje de **un** conjunto de datos.

Las tareas de minería de datos que vamos a considerar son:

- Preprocess: visualización y preprocesamiento de los datos (mal llamado filter)
- Select Attributes: Selección de atributos
- Classify: Aplicación de algoritmos de clasificación
- Visualize: Visualización de los datos por parejas de atributos

Un libro de referencia donde se explican las tareas de minería de datos, los modelos y los algoritmos principales es: *Data Mining: Practical Machine Learning Tools and Techniques* de `Witten y Frank`.

2 Un problema de clasificación: los datos de LED

Se trata de predecir el dígito mostrado en una pantalla de 7 segmentos. El problema fue planteado en el libro de CART [Breiman, L., Friedman, J., Stone, C. J., Olshen, R. A.(1984) y posteriormente incorporado al repositorio de UCI.
Se han implementado 2 tipos de generadores. Una versión pequeña con 7 atributos descriptivos (binarios) y la clase y, una versión extendida, con 24 atributos binarios, por tanto 17 son irrelevantes.
Se pueden también encontrar con ruido, esto es, donde cada atributo tiene la probabilidad de ser invertido entre 0.1 y 0.3.
Los datos proporcionados para la sesión de prácticas, son datos con extensión **(.arff)**. Podemos encontrar varios conjuntos, provenientes de cada una de las versiones comentadas con y sin ruido. Ejemplos de los ficheros de datos disponibles:

- `ledSM.arff` se corresponde a LED (S)mall de 1000 (M) muestras.
- `ledLXM.arff` se corresponde a LED (L)arge de 10000 (XM) muestras.
- `ledLXMn10.arff` se corresponde a LED Large (n)oise 10% de 10000 (XM) muestras.

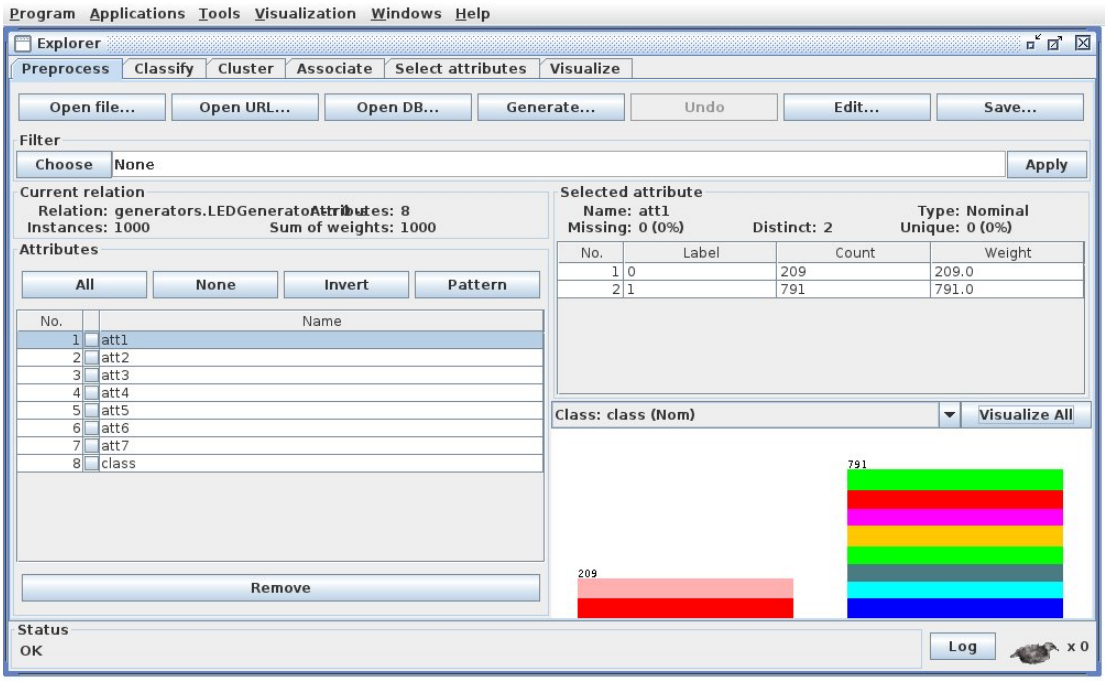
Son fichero de texto, y en la primera línea del fichero (.arff), línea de comentario, se identifica el procedimento de generación de los datos. Así, @relation 'generators.LEDGenerator -n 0' para `ledLXM.arff` o `'generators.LEDGenerator -n 30'` en el fichero `ledLXMn30.arff`.

3 Inspeccionando los datos

Usaremos el fichero `ledSM.arff`, en formato nativo de Weka, para explorar el entorno. Para ello, una vez lanzado el interfaz de usuario de Weka, será necesario seleccionar sucesivamente

```
Applications
Explorer
Open File
```

Cargamos el fichero `ledSM.arff`



En la figura se observan los atributos denominados `atr1`.. `atr7`, y `class`. En el caso concreto el atributo que se muestra con más detalle es `atr1` se muestra un gráfico de la distribución de la clase, por cada uno de los atributos seleccionando alternativamente el atributo. Los distintos valores para la clase en diferentes colores, y hay 209 muestras con etiqueta 0 *(apagado)* para el atributo `atr1` y 791 para con la etiqueta 1 *(encendido)*.
Se puede explorar visualmente la frecuencia absoluta acumulada de cada caso de la clase por cada uno de los atributos, seleccionando:

```
Visualize all
```

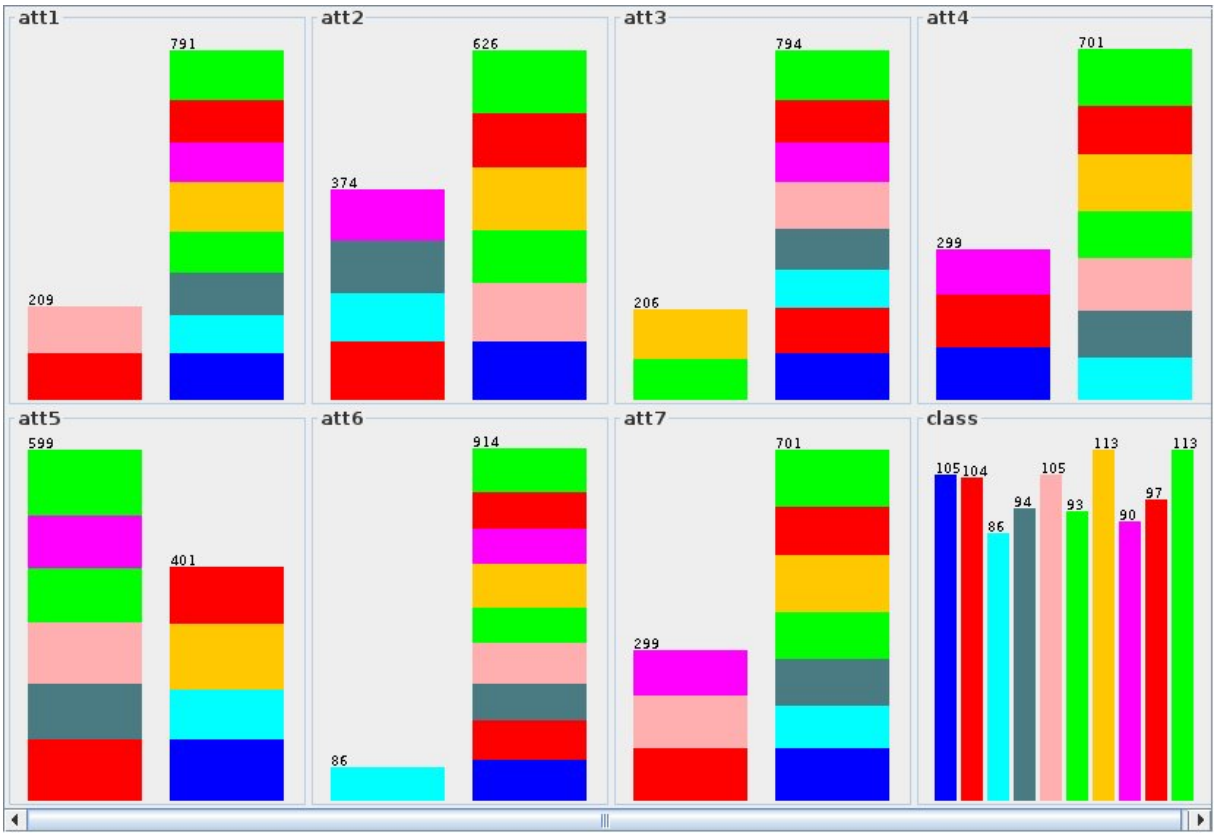


Figure 1: Distribución de frecuencias absolutas para cada atributo

En la esquina inferior derecha de la figura 1 la frecuencia absoluta acumulada de la variable clase, vemos que la clase más frecuentes con 103 muestras es la **6** y la **9** (empate).

4 Salida estándar de un clasificador

El clasificador por defecto denominado `ZeroR`, predecirá la clase **6**, primera encontrada (orden numérico de casos), para toda muestra a clasificar.

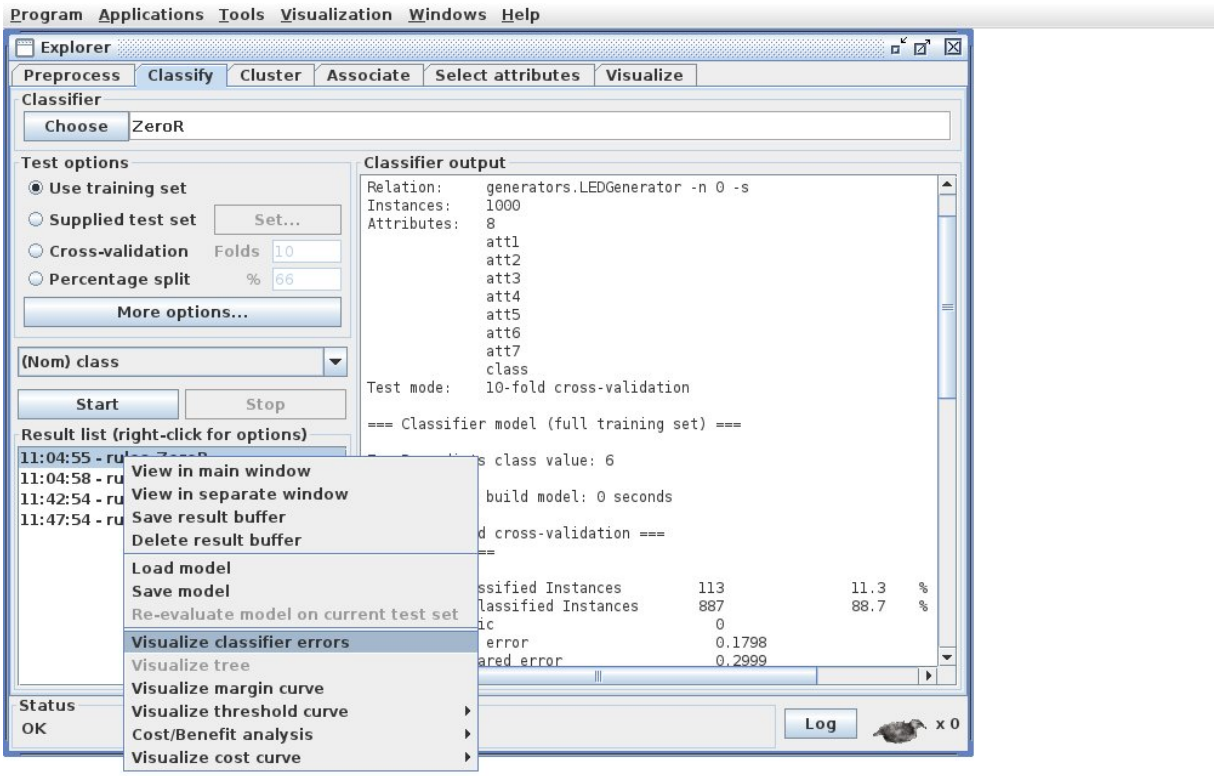
```
Classify
ZeroR
Start
```

Para estimar las tasas de éxito del clasificador (con los ojos cerrados, ZeroR), se ha usado la configuración por defecto que es Validación cruza de 10 folds. La salida en la pantalla central está convenientemente etiquetada:
Se identifica el algoritmo de clasificación seleccionado, la relación o datos que se usaron, el método de estimación de tasas de éxito, el número de muestras clasificadas, se muestra además, una lista de métricas estándares comunes a todos los clasificadores, como son Precisión, Recall, F ... que se obtienen apartir de la matriz de confusión.

Cambiar de método de estimación a particionamiento en training y test (2/3 y 1/3 respectivamente), seleccionar el botón de

```
Percentage split
Start
```

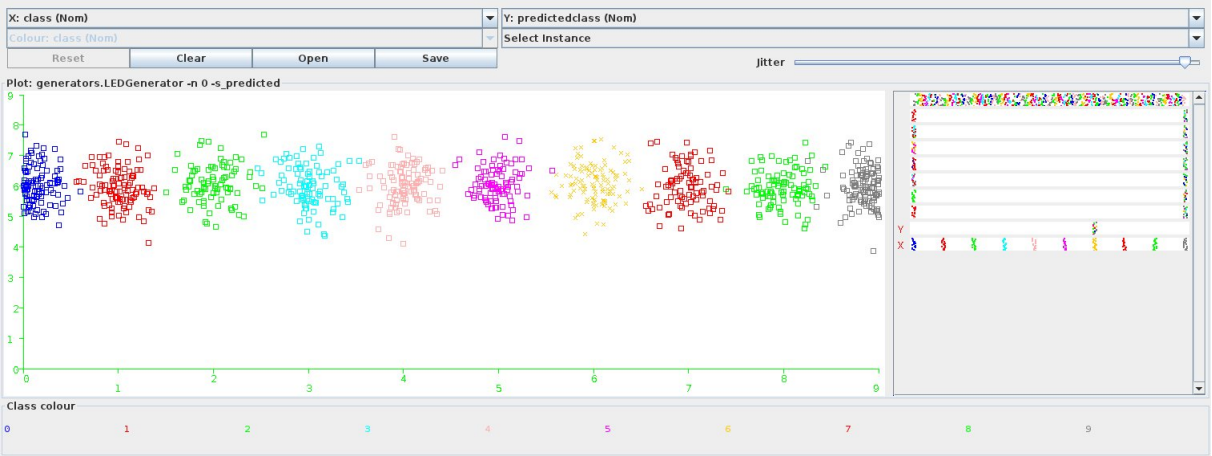
Se lleva a cabo una nueva clasificación los resultados se mantienen en el buffer, así podemos revisitar cada salidas de cada ejecución y comparar resultados. Con ValidaciónCruzada se obtiene unas tasas de error del 88.7, clasificando 1000 muestras mientras que, con partición 2/3 y 1/3 del conjunto de datos, se clasifican 340 muestras se obtiene un 89.70%.



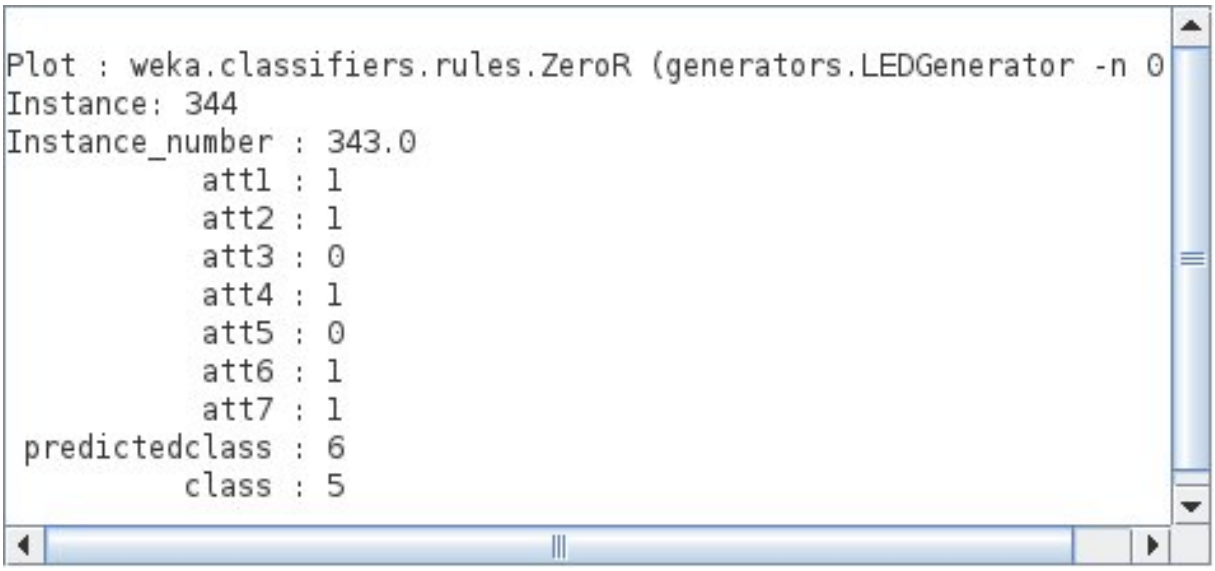
Se pueden analizar los errores de clasificación de forma visual y/o en detalle, seleccionando sobre uno de los clasificadores, con el botón derecho del ratón en la lista de resultados:

```
Visualize classifier errors
```

En el gráfico puede verse que cada una de las muestras han sido clasificadas como **6** y vemos distintas nubes de puntos de colores por cada valor de clase real, la clase predicha. No es así, con el otro clasificador, ¿porqué?

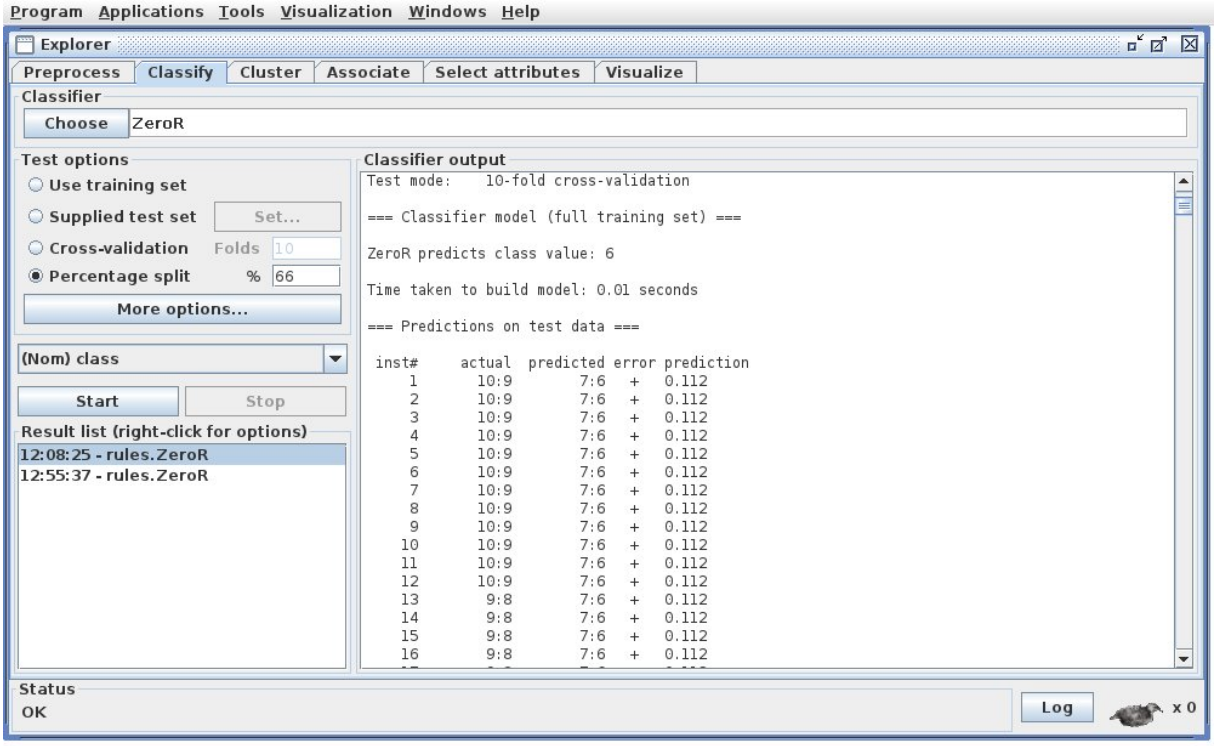


Se puede ver analíticamente cada muestra erróneamente clasificadas, pinchando en cada punto del gráfico, que se identifica por una o más muestras. Para cada uno se muestra los valores de cada atributo, clase real y clase predicha.



Puede verse una salida más completa de la clasificación efectuada por el modelo, por ejemplo para cada muestra ver clase real y la clase predicha. Selecionando sucesivamente:

```
More options
Store prediction for visualization
Plain text
```

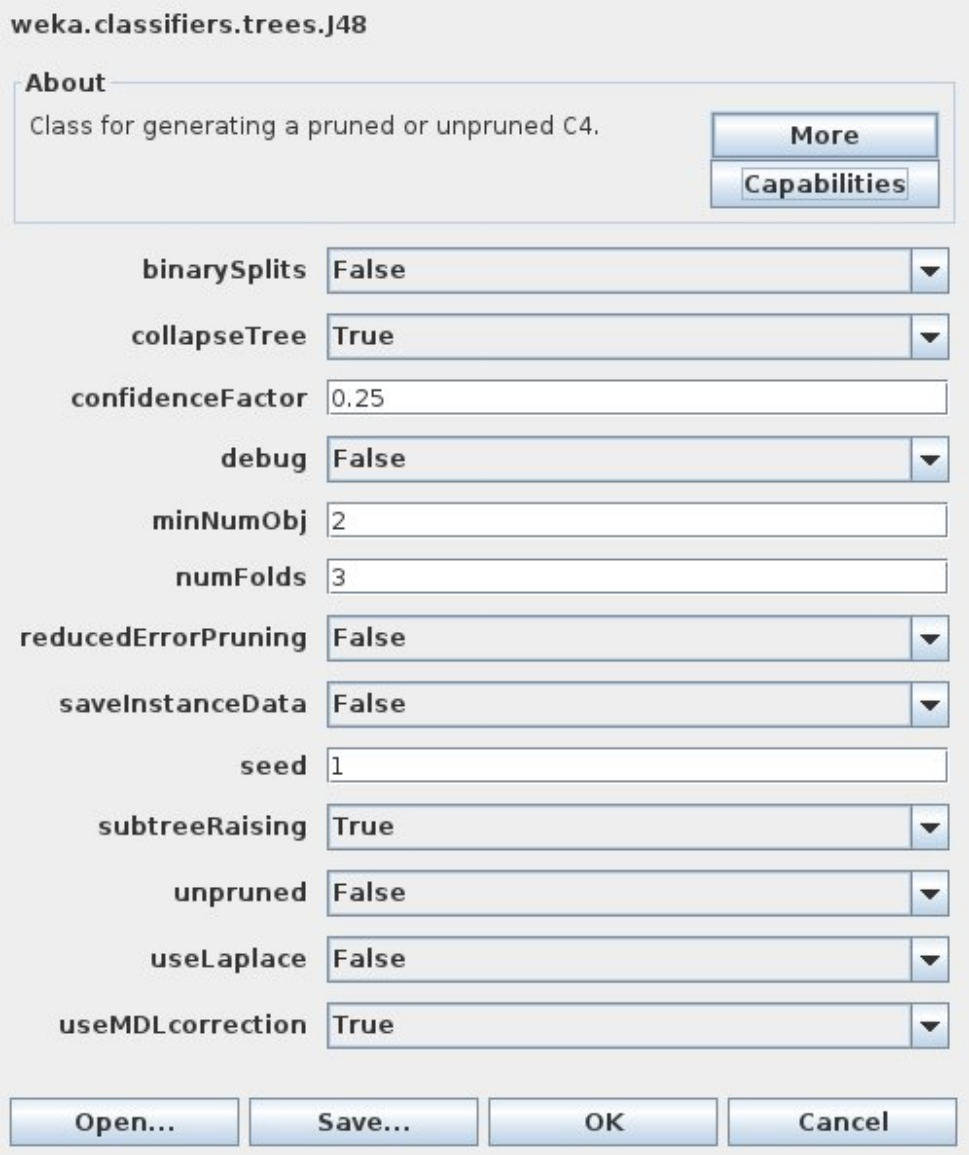


Seleccionando nuevos clasificadores.

En Weka se encuentran un gran número de métodos de clasificación basados en distintos modelos. Un clasificador bien conocido es, C4.5 basado en árboles. En la ventana de clasificación, pestaña `Classify`, hay que seleccionar

```
Choose
trees
J48
Close
```

De esta forma se selecciona el algoritmo indicado cuyo nombre en Weka se corresponde con J48. El método de clasificación está seleccionado con los parámetros por defecto. El número y el valor de los parámetros dependerá de cada método particular, pero todos se consultan y modifican de la misma forma, con una ventana de diálogo como la que se muestra a continuación.



Para ello, es necesario pulsar el botón derecho del ratón sobre la línea de parámetros: `J48 -C 0.25 -B -M 5`. Se pueden modificar los valores de los parámetros introduciendo nuevos valores o seleccionando valores entre listas desplegables, para saber más acerca del algoritmo (referencia bibliográfica) y los parámetros, sus dominios e interpretación se pulsa:

```
More
```

En la figura 2, puede comprobarse que el algoritmo usado se corresponde al C4.5

5 Las redes bayesianas como clasificadores

Algunos de los algoritmos descritos en la clase de teoría sobre clasificación con redes bayesianas, se encuentran implementados en Weka. Para ello, será necesario seleccionar el modelo de Redes bayesianas que se encuentra bajo el identificador

```
bayes
```

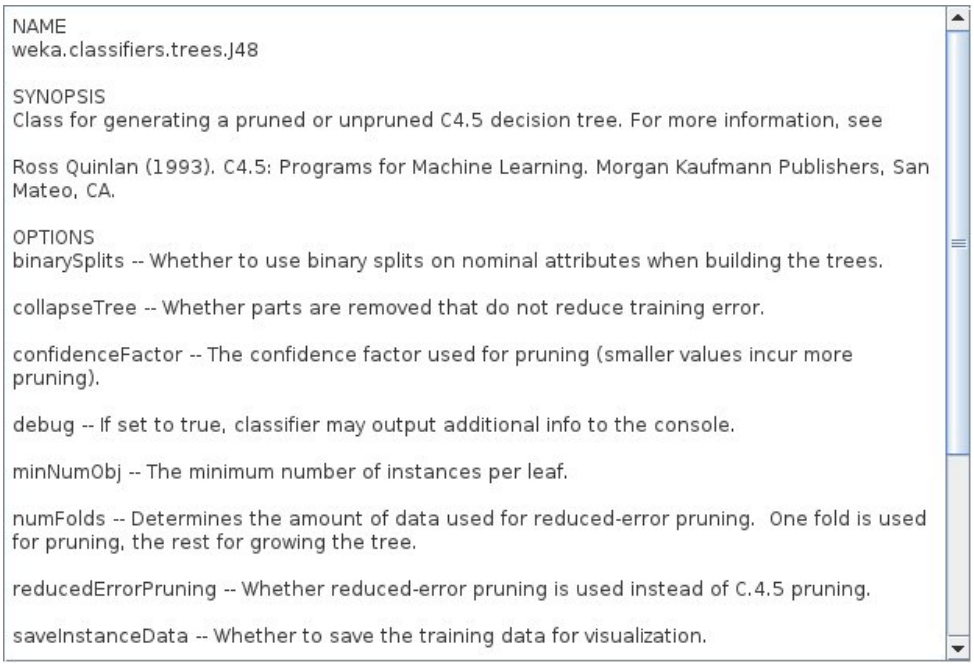


Figure 2: Información acerca del algoritmo con sus detalles técnicos, referencia obligada en caso de utilización del mismo en cualquier trabajo de investigación.

La mayoría presuponen datos discretos. Por ello, solo estarán disponibles aquellos algoritmos que pueden operar con el fichero de datos cargado. Esto puede consultarse, dado un algoritmo seleccionado con

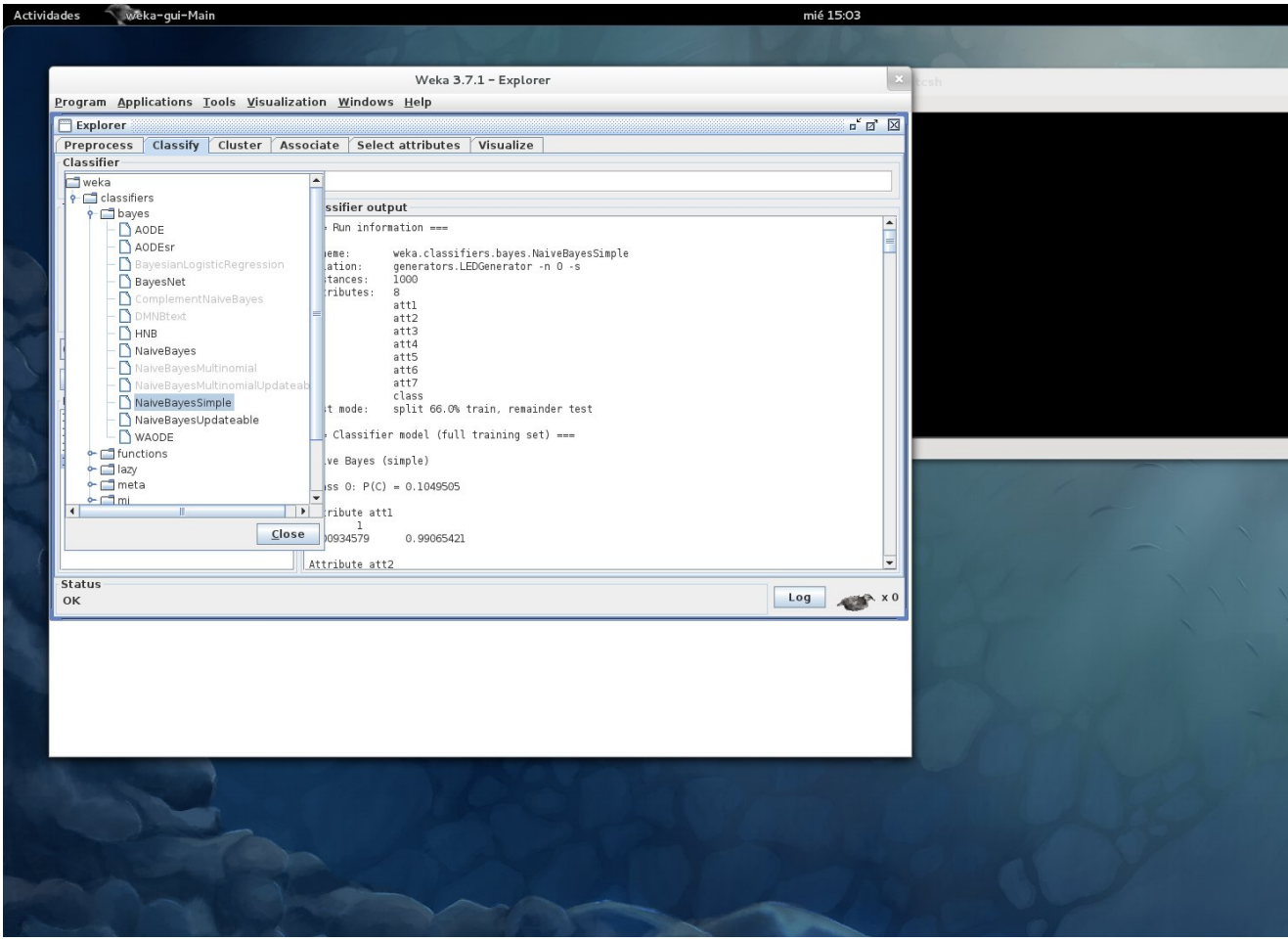
```
Capabilities
```

5.1 Naive Bayes

El primer clasificador bayesiano que vamos a utilizar es el Naive Bayes

```
NaiveBayesSimple
```

Ya conocemos la estructura del mismo, y los datos numérico aprendidos del modelo se muestran una vez realizada la clasificación. $P(C)$ y $\forall A_i, P(A_i|C)$.
Este algoritmo, a diferencia de otros que veremos a continuación, no tiene un grafo como resultado que se pueda explorar.



5.2 Modelos de redes bayesianas generales

Aprendiendo redes bayesianas generales o n-dependientes, n= 1, 2 .. M, donde primero se aprende la estructura y luego la componente paramétrica. Se encuentran bajo el algoritmo genérico denominado:

```
BayesNet
Close
```

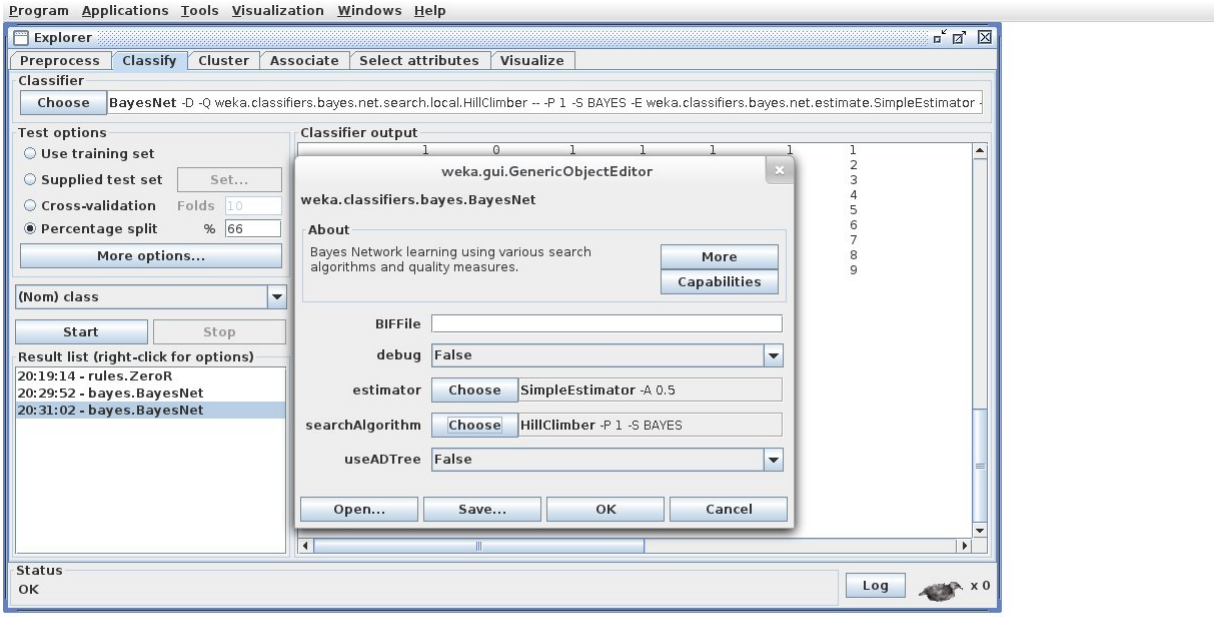
Ya hay disponible un método por defecto con sus parámetros por defecto pero, *¿Qué algoritmo concreto es?*

Pinchando en la línea de parámetros:
`BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator -- -A 0.5`

se despliega la ventana de diálogo donde, podemos consultar y seleccionar los parámetros del método seleccionado. El parámetro del método

```
estimator
```

permite seleccionar la forma de estimar los componentes numéricos, una vez la estructura de la red ya ha sido aprendida. Se puede cambiar, mediante la secuencia de selección:




```
Choose
BayesNetEstimador
Close
```

Para conocer con más detalles acerca del estimador seleccionado, pulsar en la línea de parámetro botón derecho. Para solicitar más información,

`More...`

Pero aún no sabemos cómo se aprende la estructura del clasificador...

Haciendo repaso sobre aprendizaje... Los enfoques vistos para el aprendizaje de la estructura de una RB

1. enfoque basado en métrica más búsqueda, *(search & score)*
2. enfoque basado en independencias

El enfoque *(search & score)*, se corresponde a seleccionar en el criterio de búsqueda, en `searchAlgorithm`

```
Choose
local
K2
Close
```

Esto es, estamos aprendiendo utilizando K2, métrica definida por `Cooper and Herskovits (1992)`, ver apuntes, utiliza un hillclimbing para explorar el espacio limitado a un orden preestablecido entre los atributos, tal como están definidos en la base de datos. La salida de un clasificador basado en RRBB tiene más información además de la estándar.

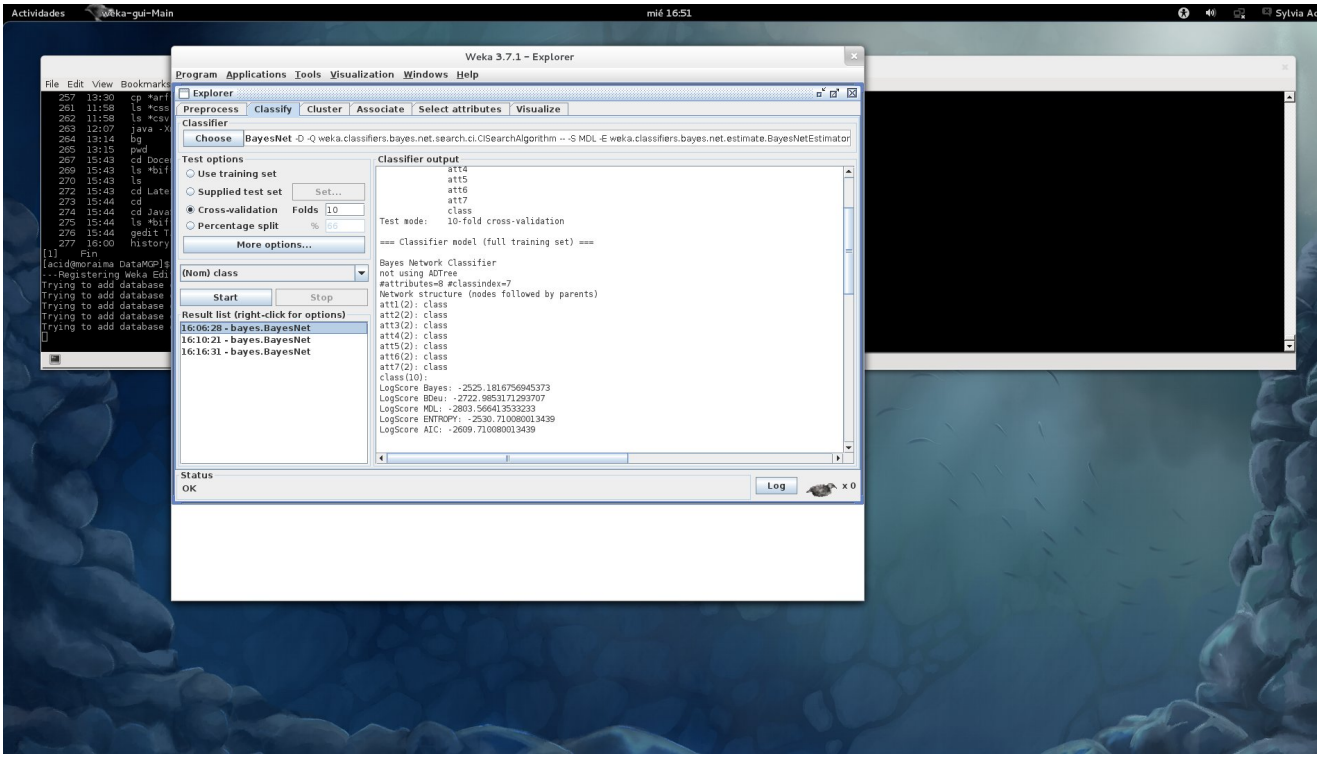


Figure 3: Salida específica de un clasificador basado en RRBB.

En la figura 3, puede verse la lista de padres de cada nodo, así como métricas adicionales que miden el ajuste del modelo a los datos. Así, a igualdad de tasas de éxito de 2 clasificadores, se puede discriminar usando criterios adicionales.
Pero esto no es lo único específico de los RRBB. Una vez obtenido el clasificador se puede ver la estructura aprendida, ver figura 4.
Para ello, en el método seleccionado para construir el clasificador, `result list`, botón derecho del ratón

```
Visualize graph
```

Pero hay mucho más, siguiendo con el método BayesNet, y con la estrategia *(search & score)* ... En figura 5 pueden verse algunos de los algoritmos vistos para la clasificación, que van desde estrategias de búsqueda aleatorias como Genéticos o Simulated annealing pasando por espacios de búsqueda restringidos como TAN.

Todas ellas se combinan con distintas métricas *score* de las estudiadas como son MDL, BDe, AIC etc, para evaluar cada red explorada en el espacio. Se puede además partir de una red vacía inconexa o apartir del Naive Bayes...

El enfoque basado en*(independencias)*, se corresponde a seleccionar en el criterio de búsqueda, en `searchAlgorithm`

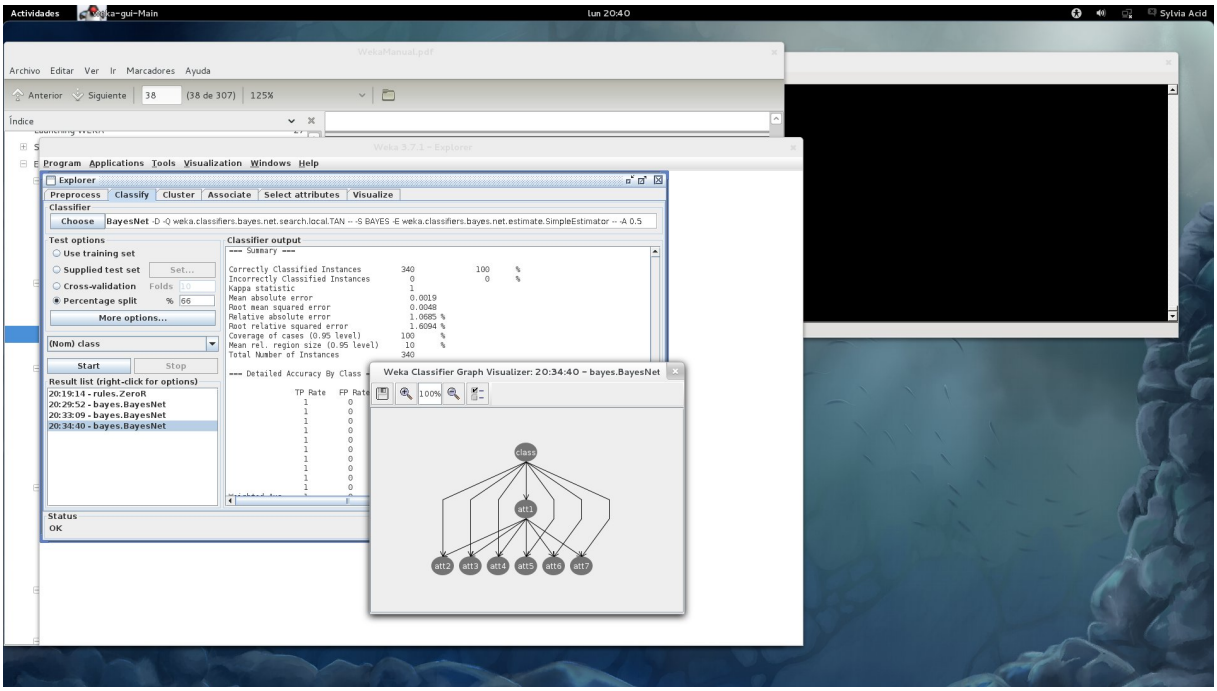


Figure 4: Visualización de la estructura del clasificador aprendido.

```
Choose
ci
ICSSearchAlgoritm
Close
```

Por último nos queda comentar otra forma de aprender un clasificador basado en RRBB. La estrategia consiste en cambiar la métrica y usar las tasas de éxito para dirigir la búsqueda, lo que se conoce como Wrapper. Esta estrategia se selecciona en el criterio de búsqueda, en `searchAlgorithm`

```
Choose
global
HillClimber
Close
```

Se usa una estrategia de búsqueda HillClimber, que añade y borra arcos, que usa un método de estimación del clasificador Leaving-one-out, con restricción en el número de padres con lo que reduce mucho el espacio de búsqueda y por tanto simplifica mucho el modelo.

6 Parte obligatoria

Aprender distintos clasificadores para el conjunto de datos `ledLXMn30.arff`. Apartir de un modelo original de referencia, indicar las 3 redes que mejores tasas de clasificación han logrado. Realizar un pequeño informe con la descripción del problema,

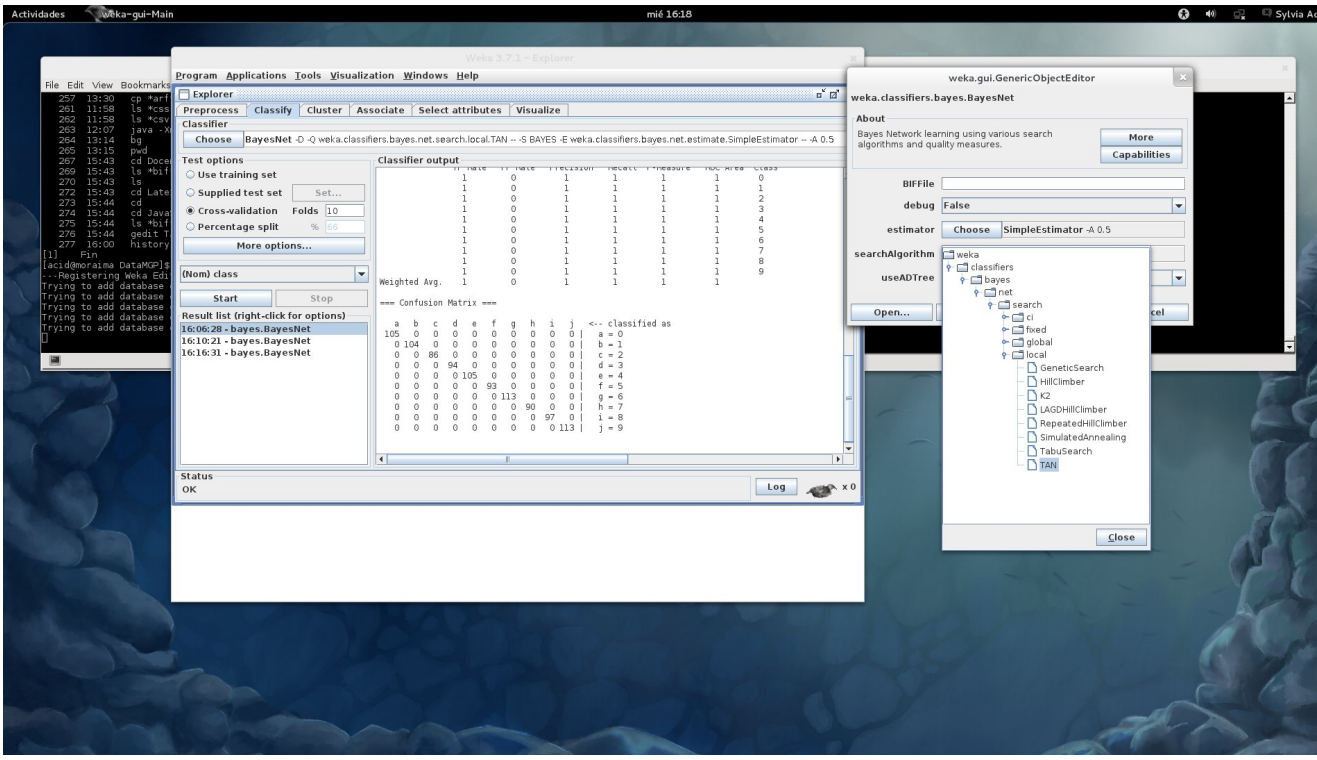


Figure 5: Métodos de búsqueda implementados en Weka que se puede utilizar para explorar el espacio

preprocesamiento de los datos (en caso de haberse utilizado), las redes obtenidas (gráfica) así como la especificación detallada de los parámetros en utilizados en su obtención.

7 Parte para subir nota

Como trabajo opcional de esta parte de la asignatura, se debe entregar un script en R para construir un clasificador bayesiando de cualquiera de los realizados en la etapa anterior.

References

[Breiman, L., Friedman, J., Stone, C. J., Olshen, R. A.(1984)] Classification and regression trees. *CRC press.*

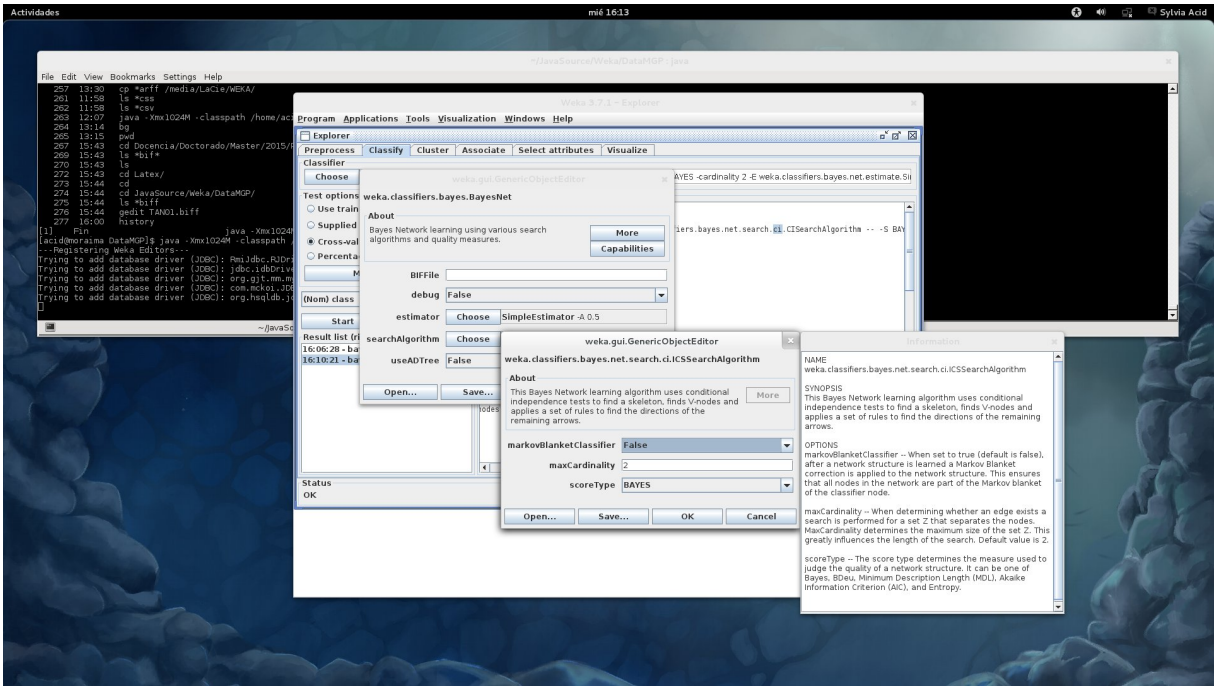


Figure 6: Enfoque basado en independencias