# Práctica 4: Weka – Clasificación con redes bayesianas

## 1. Introducción

En este guión vamos a aprender distintos clasificadores para el conjunto de datos “ledLXMn30.arff” con Weka. Como bien dice el nombre del archivo, tenemos un conjunto de datos con 10000 muestras y un 30% de ruido, por lo que vamos a tener que aplicar preprocesamiento a los datos.

## 2. Inspeccionando los datos

Lo primero, será visualizar el conjunto de datos que tenemos:

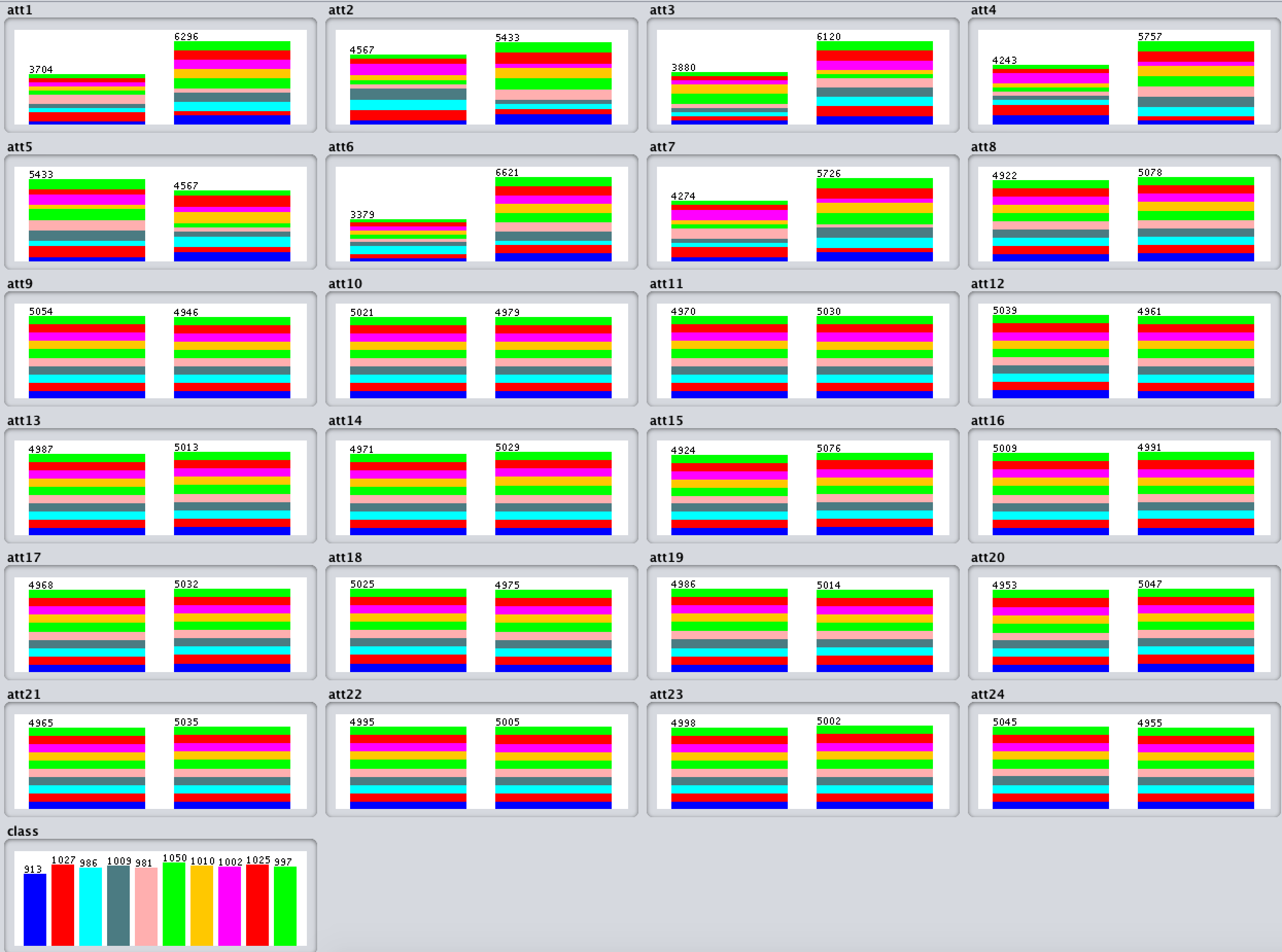


Figura 1: Distribución de frecuencias absolutas para cada atributo

Viendo todas las variables y el conjunto de datos que tiene cada uno. Podemos observar como los atributos att1 al att24 son atributos binarios en los que su valor será 0 o 1. La variable clase por el contrario tiene 10 clases, siendo del 0 al 9 sus etiquetas.

## 3. Selección de características

Lo primero será seleccionar características ya que tenemos 24 atributos de las cuales seguramente podamos desechar algunas. Por lo tanto, vamos a la pestaña “Select attributes” y aplicamos un evaluador “CfsSubsetEval” y como método de búsqueda el “GreedyStepwise”, con una cross-validation de 10 Folds, quedando como resultado:

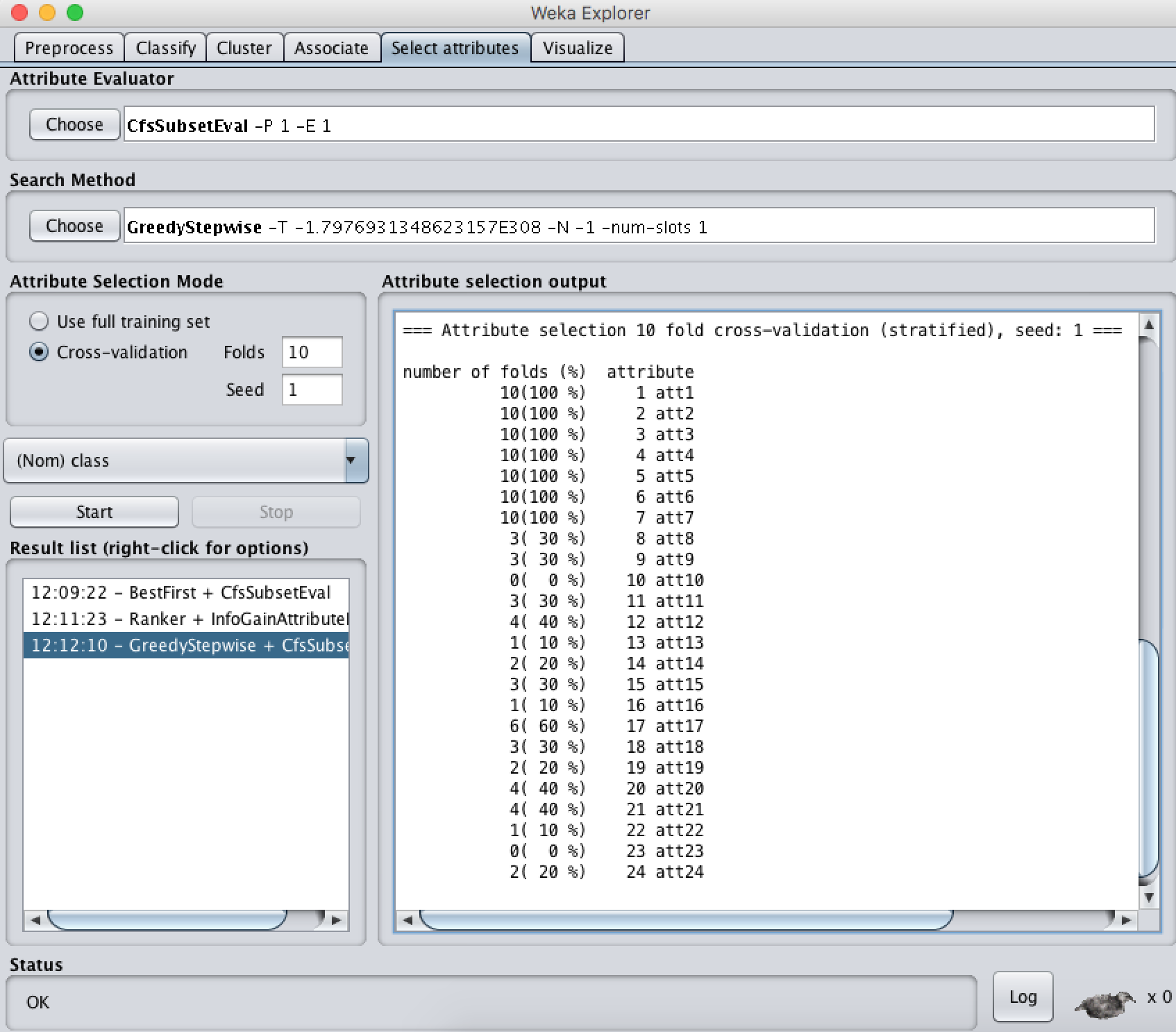


Figura 1: Distribución de frecuencias absolutas para cada atributo

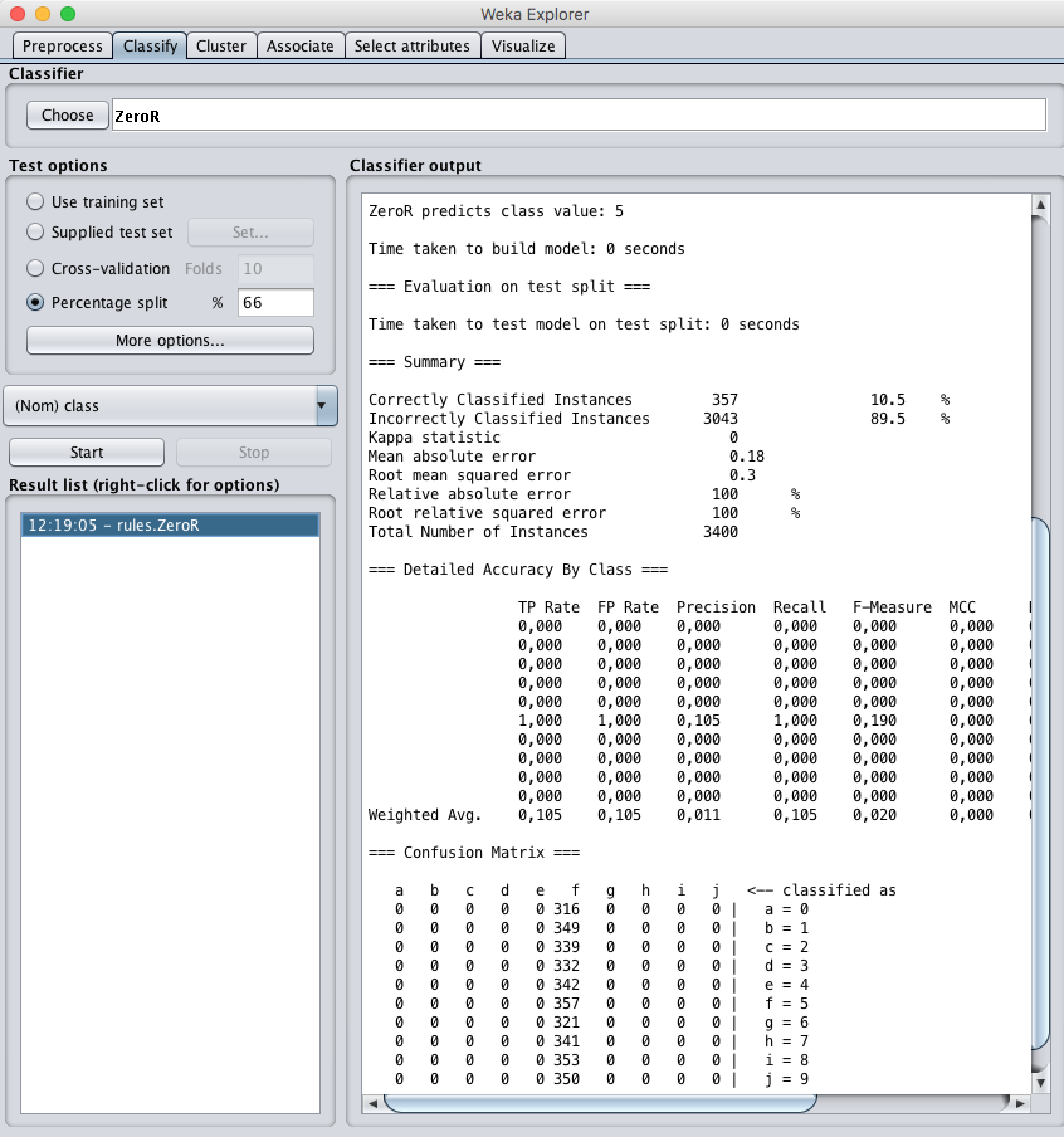
Por lo que nos quedaremos con todos los atributos excepto el 10, 23, 13, 16, 22, 14, 19 y 24. Quedando los datos:



Figura 1: Distribución de frecuencias absolutas para cada atributo

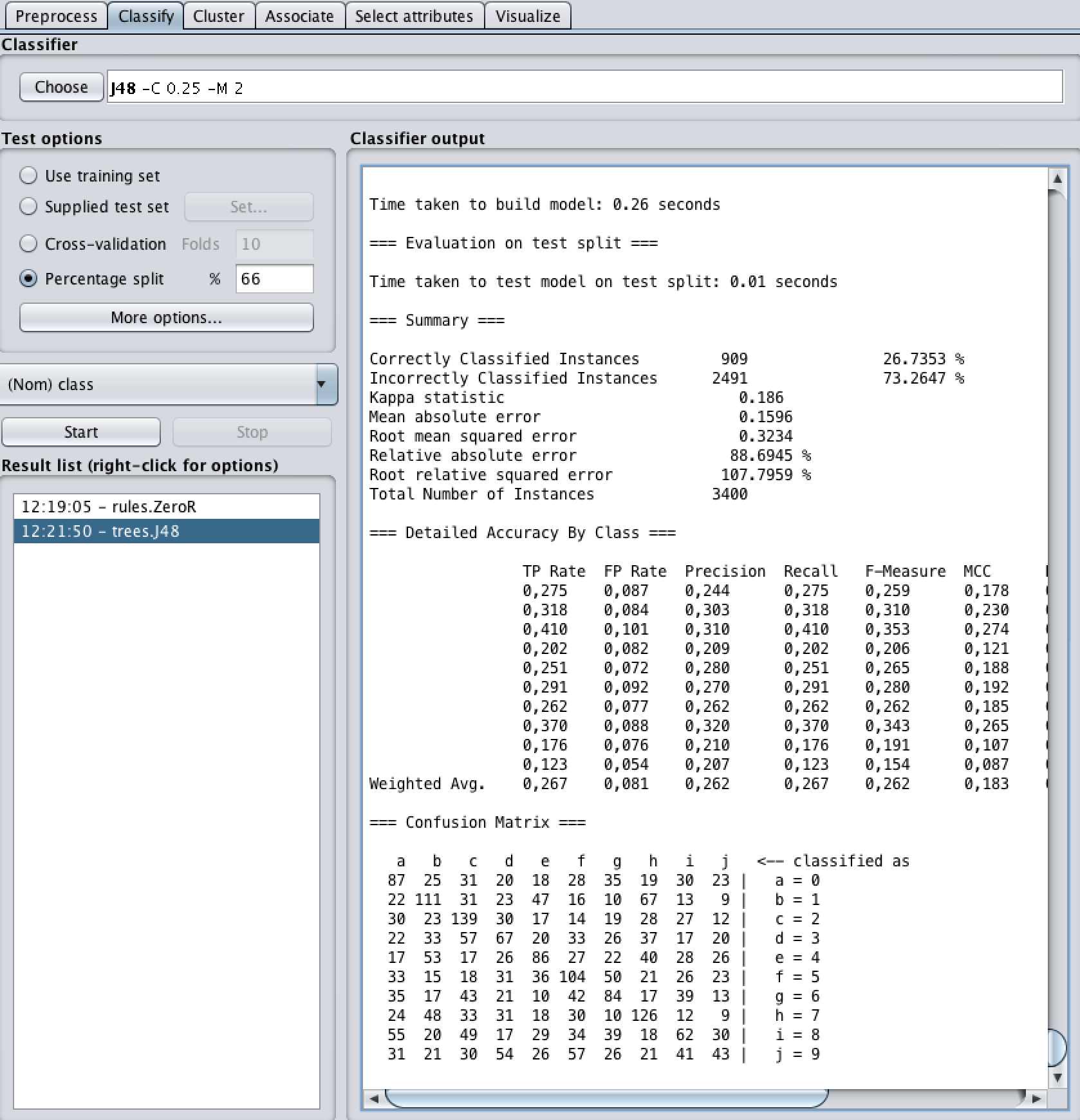
## 4. Clasificación con selección de características

Si primeramente aplicamos un clasificador ZeroR para obtener una visión preliminar de cómo funciona el clasificador para este conjunto de datos obtenemos:



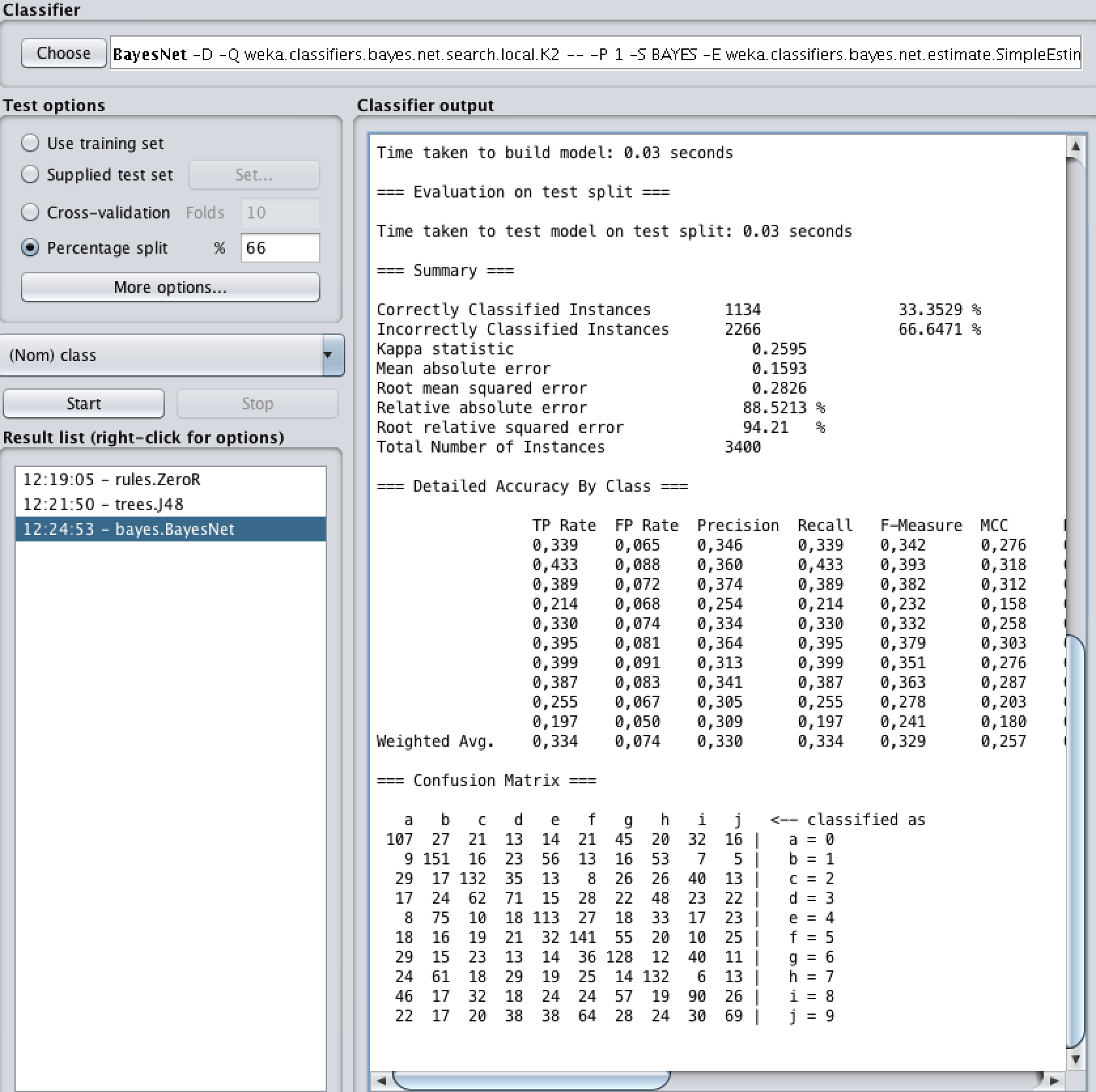
Donde tenemos un accuracy del 89.5%.

Si por ejemplo aplicamos un árbol C4.5 obtenemos un árbol con un número de hojas de 1409 y un tamaño del árbol de 2817.



Donde el accuracy es del 73.26%, mucho peor que el ZeroR.

Si pasamos a usar técnicas basadas en bayes, podemos tener los siguientes resultados:



Donde el accuracy ha sido del 66.64%, mucho menor que los anteriores. Si por ejemplo cambiamos el algoritmo de búsqueda por un HillClimber, obtenemos un accuracy del 66.5%.