Minería de Datos

Práctica 1. Práctica introductoria a Weka

Ángel Ríos San Nicolás

19 de marzo de 2021

1. Responde las siguientes cuestiones:

   a. Abre el fichero de datos "credit-g.arff".

   b. ¿Cuántos atributos tiene? ¿De qué tipo son cada uno de ellos?
   Tiene los siguientes 21 atributos:

Número	Nombre	Tipo
1	*checking_status*	Nominal
2	*duration*	Numérico
3	*credit_history*	Nominal
4	*purpose*	Nominal
5	*credit_amount*	Numérico
6	*savings_status*	Nominal
7	*employment*	Nominal
8	*installment_commitment*	Numérico
9	*personal_status*	Nominal
10	*other_parties*	Nominal
11	*residence_since*	Numérico
12	*property_magnitude*	Nominal
13	*age*	Numérico
14	*other_payment_plans*	Nominal
15	*housing*	Nominal
16	*existing_credits*	Numérico
17	*job*	Nominal
18	*num_dependents*	Numérico
19	*own_telephone*	Nominal
20	*foreign_worker*	Nominal
21	*class*	Nominal

   c. ¿Cuántas instancias?
   El fichero de datos tiene 1000 instancias.

   d. ¿Existen atributos con valores perdidos (missing)? ¿Cuáles?
   No existen atributos con valores perdidos.

   e. ¿Cuáles son los valores máximo, mínimo y medio del campo "age"? ¿Qué distribución siguen? ¿Cuántos valores de edad diferentes hay? ¿Cuántos de esos valores no se repiten? ¿Y en cuánto al campo "credit_amount"?

   - Campo *age*.
   Mínimo: 19
   Máximo: 75
   Media: 35.546
   La distribución de los datos respecto al atributo *age* tiene una clara asimetría positiva, es decir, la densidad de las edades crece rápidamente con la edad y después,

en general, decrece lentamente de manera que la moda es menor que la mediana que a su vez es menor que la media.
Hay 53 valores diferentes de los que solo uno no se repite.

- Campo *credit_amount*.
  Mínimo: 250
  Máximo: 18424
  Media: 3271.258
  La distribución de los datos respecto al atributo *credit_amount* se asemeja a la anterior, tiene una asimetría positiva mucho más pronunciada.
  Hay 921 valores distintos de los que 847 son únicos, no se repiten.

f. ¿Cuál es el valor más frecuente de la clase y en cuánta proporción?
   El valor más frecuente de la clase es *good* con una proporción de $\frac{700}{1000} = 0.7$, es decir, el 70%.

2. Realiza las siguientes tareas de prepocesado:

   a. Abre el fichero de datos "credit-g.arff".

   b. Cambia el nombre del atributo de clase de "Class" a "credit".

   c. Crea una copia del atributo "age". Renombrala a "ageDisc5", y discretiza sus valores en 6 rangos diferentes. Observa la distribución obtenida, ¿te parece bien esta discretización? Realiza nuevas copias del atributo "age", prueba a discretizar con diferente número de "bins" y compara los resultados.

      La discretización con solo 6 rangos diferentes no es adecuada porque no refleja bien la distribución por edades. Hay demasiadas instancias en cada rango con lo que las frecuencias son muy altas. No se aprecia bien, por ejemplo, la asimetría.

      Con 10 y 20 rangos diferentes se aprecia mucho mejor la distribución de los datos. Se aprecia la asimetría positiva. Con 20 rangos podemos observar tendencias que no observamos con 10, por ejemplo, que la distribución no decrece a partir de la moda porque en $(33, 35.8]$ hay 72 instancias y en $(35.8, 38.6]$ hay 92.

   d. Repite la operación, usando PKIDiscretize con la opción "useEqualFrecuency" activada. ¿Qué diferencias ves con respecto a lo obtenido en la pregunta anterior?

      Independientemente del número de "bins" escogido, PKIDiscretize siempre discretiza en 31 rangos porque, según su descripción, fuerza a que el número de "bins" sea la raíz cuadrada del número de valores del atributo y $\sqrt{1000} = 31.622776601683793...$, con lo que se toman 31.

      Observamos que no es cierto que los rangos tengan el mismo tamaño. Los primeros rangos separan edades con diferencias de un año, con lo que prácticamente tenemos la frecuencia de cada edad. Sin embargo, los últimos rangos engloban más valores del atributo, por ejemplo, el rango $(60.5, 64.5]$ que combina los valores $61, 62, 63, 64$. Esto provoca que haya frecuencias más altas por el hecho de permitir más valores, lo que puede llevar a malinterpretar los datos si no se tiene en cuenta.

   e. Combina los valores "blank" y "stores" del atributo "other_payment_plans" en uno sólo, de forma que sólo tenga dos valores.

   f. Elimina los atributos "employment" y "purpose".

   g. Normaliza los atributos numéricos en una escala de 0 a 1.

   h. Guarda el fichero .arff con todos los cambios.

   i. Abre el fichero de datos "breast-cancer.arff".

   j. Observa las estadísticas del atributo "node-caps". Busca una función que sirva para reemplazar valores perdidos. Aplícala sobre este atributo. ¿Qué ha sucedido?
      Las estadísticas del atributo *node-caps* antes de aplicar la función son:

Perdidos: 8 (3%) Distintos: 2 Únicos: 0 (0%)

n.º	Valor	Cantidad	Peso
1	*yes*	56	56.0
2	*no*	222	222.0

Las estadísticas del atributo *node-caps* al aplicar la función son:

Perdidos: 0 (0%) Distintos: 2 Únicos: 0 (0%)

n.º	Valor	Cantidad	Peso
1	*yes*	56	56.0
2	*no*	230	230.0

Observamos que los valores perdidos se han reemplazado por el valor *no*. Como explica la propia descripción de la función, en los tipos de datos nominales, reemplaza los valores perdidos por la moda, en este caso, por *no*.

k. Prueba a eliminar, con la función adecuada, todos los atributos de tipo nominal. ¿Cuáles permanecen?

Aplicamos la función *RemoveType* para la que seleccionamos "Delete nominal attributes" de la opción *attributeType*. Como todos los atributos eran nominales, solo permanece el atributo *Class* que, aunque es nominal, es especial por estar marcado como clase y sobrevive al proceso.

l. Abre el fichero "ionosphere.arff"

m. Busca una función que permita eliminar a aquellos atributos "Useless", y prueba a eliminar todos los atributos que apenas, usando una sensibilidad de 99.0. ¿Se elimina alguno?

Aplicamos la función *RemoveUseless* con la opción *maximumVariancePercentageAllowed* a 99.0 y se elimina el atributo *a02*.