**TEMA 2: Técnicas de minería de datos en Weka. Clasificadores**

**Alumno: Francisco Márquez**

**Actividad: Actividades Tema 3**

**Actividad 3.1. Realiza los histogramas de las calificaciones de bachillerato y calificación final de la prueba, indicando como segundo atributo la convocatoria en la que se presentan los alumnos.**

**Histograma Calificación asignatura 1.**

Graphical user interface, application

Description automatically generated

**Histograma Calificación asignatura 2.**

Graphical user interface, application

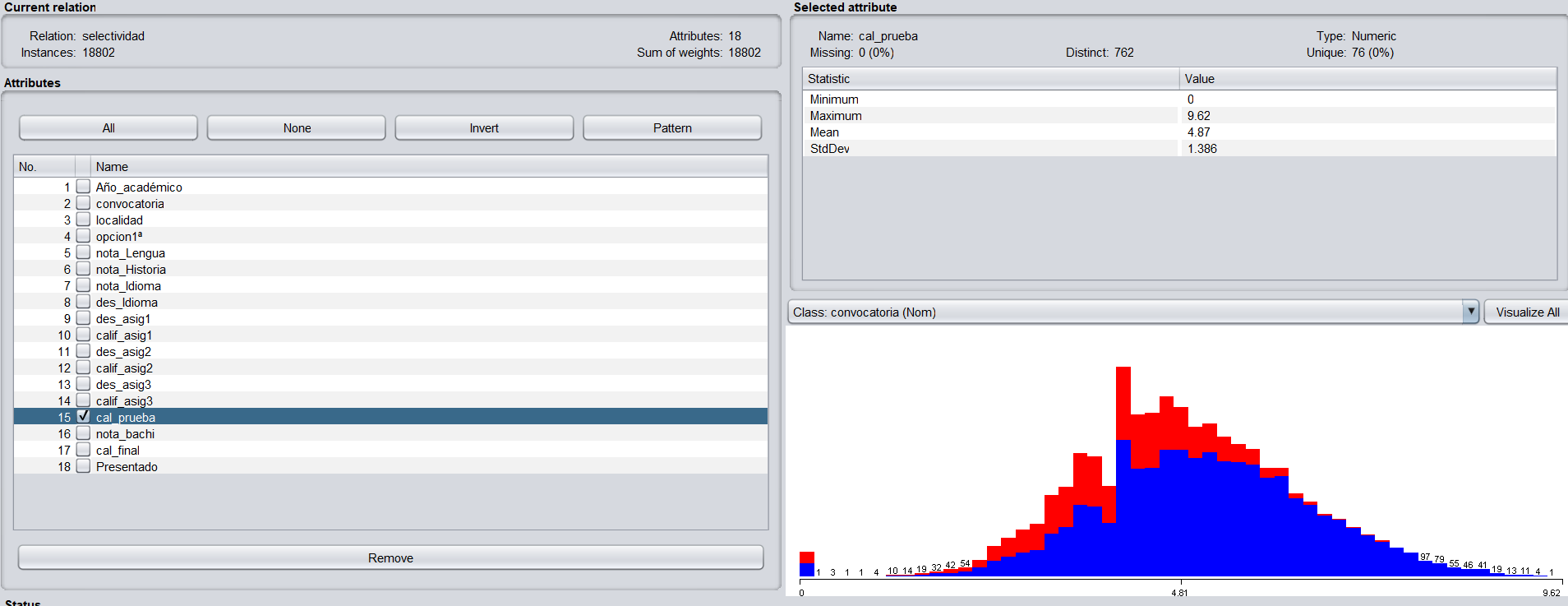
Description automatically generated

**Histograma Calificación asignatura 3.**

Graphical user interface, application

Description automatically generated

**Histograma Calificación prueba.**



**Actividad 3.2. Realiza una nueva discretización de la relación (eliminando el efecto del filtro anterior y dejando la relación original con el botón Undo) que divida las calificaciones en 4 intervalos de la misma frecuencia, lo que permite determinar los cuatro cuartiles (intervalos al 25%) de la calificación en la prueba: los intervalos delimitados por los valores {4, 4.8, 5.76}.**

A continuación, se presenta el resultado de la discretización:

Graphical user interface, application

Description automatically generated

**Actividad 3.3. Utiliza tres filtros de este tipo para seleccionar los alumnos de Getafe y Leganés con una calificación de la prueba entre 6.0 y 8.0. Comprueba el efecto de filtrado visualizando los histogramas de los atributos correspondientes (localidad y calificación en la prueba).**

**Filtro de localidad para seleccionar Getafe y Leganés**

Graphical user interface, application

Description automatically generated

**Filtro de Calificación de prueba para obtener notas superiores a 6**

Graphical user interface, application

Description automatically generated

**Filtro de Calificación de prueba para obtener notas inferiores a 8**

Graphical user interface, application

Description automatically generated

**Histograma de Localidad**

Graphical user interface, application

Description automatically generated

**Histograma de Calificación de la prueba**

Graphical user interface, application

Description automatically generated

**Actividad 3.4. Visualiza la relación entre las tres asignaturas optativas, y con la opción cursada como color**

**Asignatura 1 con Asignatura 2 con opción**

Graphical user interface, application

Description automatically generated

**Asignatura 1 con Asignatura 3 con opción**

Graphical user interface, application

Description automatically generated

**Asignatura 2 con Asignatura 3 con opción**

Graphical user interface, application

Description automatically generated

**Actividad 3.5. Obtén el árbol de decisión (gráfico) del ejemplo de clasificación anterior.**

Diagram

Description automatically generated

**Actividad 3.6. Compara este resultado con el obtenido al utilizar los otros modos de evaluación del clasificador posibles.**

Al comparar los distinto métodos de clasificación, vemos como los algotimos Decision Table, JRIP y PART ofrecen mejores tasas de aciertos en la clasificación que el algorimo OneR.

OneR:

Correctly Classified Instances 13634 72.5136 %

Decision Table:

Correctly Classified Instances 13904 73.9496 %

JRIP:

Correctly Classified Instances 14050 74.7261 %

PART

Correctly Classified Instances 14714 78.2576 %

ZeroR

Correctly Classified Instances 9476 50.3989 %

**Actividad 3.7. Realiza de nuevo el árbol utilizando en este caso un valor del factor de confianza de 0.05 para la poda y como mínimo número de instancias por nodo 50. Compara los resultados obtenidos.**

=== Classifier model (full training set) ===

J48 pruned tree

------------------

nota\_bachi <= 6.8

| nota\_bachi <= 6.1: '(-inf-4.81]' (7340.0/1794.0)

| nota\_bachi > 6.1

| | convocatoria = J

| | | Año\_académico <= 2000

| | | | opcion1ª = 1: '(-inf-4.81]' (605.0/255.0)

| | | | opcion1ª = 2: '(4.81-inf)' (688.0/295.0)

| | | | opcion1ª = 3: '(-inf-4.81]' (314.0/154.0)

| | | | opcion1ª = 4: '(-inf-4.81]' (774.0/323.0)

| | | | opcion1ª = 5: '(-inf-4.81]' (84.0/33.0)

| | | Año\_académico > 2000: '(4.81-inf)' (1457.0/626.0)

| | convocatoria = S: '(-inf-4.81]' (1222.0/369.0)

nota\_bachi > 6.8

| convocatoria = J: '(4.81-inf)' (6010.0/987.0)

| convocatoria = S

| | nota\_bachi <= 7.5: '(-inf-4.81]' (238.0/101.0)

| | nota\_bachi > 7.5: '(4.81-inf)' (70.0/20.0)

Number of Leaves : 11

Size of the tree : 18

Time taken to build model: 0.25 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.04 seconds

=== Summary ===

Correctly Classified Instances 13845 73.6358 %

Diagram

Description automatically generated

Este modelo mejora del modelo generado con OneR. Los atributos más importantes son la calificación de bachillerato, la convocatoria, y después el año.

**Actividad 3.8. Comenta los resultados sobre la precisión y tamaño del ejemplo anterior.**

**Actividad 3.9. Realiza el problema de clasificación anterior y comenta los resultados obtenidos.**

En primer lugar, realizamos la discretización por la variable ‘mejora’

Graphical user interface

Description automatically generated

Luego realizamos la ejecución del algotimo de clasificación J48. Con el siguiente resultado:

Scheme: weka.classifiers.rules.OneR -B 2

Instances: 18802

Attributes: 7

Año\_académico

convocatoria

localidad

opcion1ª

nota\_bachi

Presentado

mejora

Test mode: evaluate on training data

=== Summary ===

Correctly Classified Instances 10629 56.5312 %

Con base en el resultado obtenido vemos que el poder para la clasificar en forma correcta el atributo es débil, ya que su tasa de acierto es de sólo el 56.5%

**Actividad 3.10. Utilizando el fichero weather.nominal.arff , ejecuta el algoritmo de clasificación Id3 en los 3 casos siguientes:**

* **Use training set**
* **Cross validation**
* **Percentage split**

**Describe el árbol obtenido. ¿Con que método de validación se han obtenido mejores porcentajes de bien clasificados?**

A continuación, se presentan los resultados obtenidos de los tres casos solicitados:

1. Training set:

=== Run information ===

Scheme: weka.classifiers.trees.Id3

Relation: weather.nominal

Instances: 14

Attributes: 5

outlook

temperature

humidity

windy

play

Test mode: evaluate on training data

=== Summary ===

Correctly Classified Instances 14 100 %

2. Cross validation:

=== Run information ===

Scheme: weka.classifiers.trees.Id3

Relation: weather.nominal

Instances: 14

Attributes: 5

outlook

temperature

humidity

windy

play

Test mode: 10-fold cross-validation

=== Summary ===

Correctly Classified Instances 12 85.7143 %

3. Percentage Split:

=== Run information ===

Scheme: weka.classifiers.trees.Id3

Relation: weather.nominal

Instances: 14

Attributes: 5

outlook

temperature

humidity

windy

play

Test mode: split 66.0% train, remainder test

=== Summary ===

Correctly Classified Instances 3 60 %

Por los resultados obtenidos, vemos como la mayor tasa de clasificación correcta se obtiene usando el conjunto de entrenamiento, con un 100%.

**Actividad 3.11. Aplica los siguientes clasificadores sobre el fichero de datos Drug1n.arff:**

* **ZeroR**
* **OneR**
* **Ibk**
* **NaiveBayes**
* **Id3**
* **j48**

**La validación se realizará sobre el mismo conjunto de aprendizaje. ¿Cuáles son los modelos que proporcionan los mejores resultados? ¿Has conseguido ejecutar todos los algoritmos? ¿Qué problemas has encontrado? ¿Cómo se pueden resolver?**

A continuación, se presentan los algoritmos y los % de clasificados correctamente que obtuvieron mejores resultados en la clasificación:

IBK 100%

J48 97%

Naive Bayes 91.5%

El algoritmo id3 no permite la ejecución. Ya que de entrada los atributos y la clase deben ser nominales. Para ejecutarlo en el preprocesamiento, se debe aplicar un cambio en los atributos numéricos aplicando el siguiente el filtro ***NumericToNominal*** a los atributos numéricos.

Graphical user interface, application

Description automatically generated

**Actividad 3.12.**

**Debes contestar de la forma más formal posible, además recuerda incluir capturas de pantalla de los pasos intermedios.**

**1. Obtención de los datos**

**Descarga el conjunto de datos iris.arff. Abre el fichero de datos con un editor, y estudia su contenido:**

Text

Description automatically generated

A picture containing text

Description automatically generated

* **¿Cuántos atributos caracterizan los datos de esta tabla de datos?**

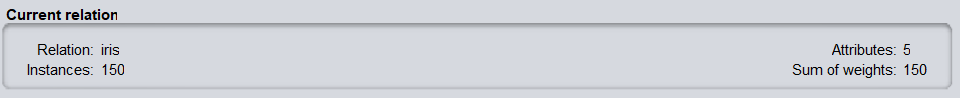
4 atributos: 1. sepal length in cm, 2. sepal width in cm, 3. petal length in cm, 4. petal width in cm

* **Si suponemos que queremos predecir el último atributo a partir de los anteriores, ¿estaríamos ante un problema de clasificación o de regresión?**

Clasificación. El problema de clasificación consiste en predecir una determinada clase (categórica) para un objeto, en donde se conoce la clase verdadera de cada uno de los ejemplos que se utilizan para construir el clasificador.

**2. Estudio estadístico de los datos**

* **Abre el fichero iris.arff en el Explorer de WEKA. Recuerda que en la sección attributes se puede pinchar sobre cada atributo para obtener información estadística del mismo.**



* **¿Cuál es el rango de valores del atributo *petalwitdth*?**

El rango de Petal Witdh es 2.5-0.1 = 2.4

Graphical user interface, application, table

Description automatically generated

* **¿Con la información que puedes obtener visualmente, ¿qué atributos crees que son los que mejor permitirían predecir el atributo *class*?**

Considero que los mejores atributos para predecir ‘class’ serian:

Petalwidht y PetalLength, porque los valores de class en estos atributos se observan mejor diferenciados:

Petalwidht/class:

Chart

Description automatically generated

PetalLength/class:

Chart

Description automatically generated

Sepalwidth/class:

Chart

Description automatically generated

Sepallength:

Chart

Description automatically generated

**3. Aplicación de filtros**

* **Aplica el filtro filters/unsupervised/attribute/normalize sobre el conjunto de datos. ¿Qué efecto tiene este filtro?**

Este filtro normaliza los datos de atributos numéricos. No se observa un cambio significativo en la distribución del conjunto de datos excepto porque ahora los valores presentan un rango de valores que van de 0 a 1 como se muestra para el caso de Petalwidth:

Table

Description automatically generated with low confidence

* **Aplica el filtro filters/unsupervised/instance/RemovePercentage sobre el conjunto de datos. ¿Qué efecto tiene este filtro?**

Al aplicar el filtro con el valor por defecto, este remueve al 50% de las instancias:

Graphical user interface, text

Description automatically generated

* **Graba el conjunto de datos como iris2.arff.**

Text

Description automatically generated

* **Aplica el filtro filters/unsupervised/attribute/Discretize sobre el conjunto de datos. ¿Qué efecto tiene este filtro?**

La discretización aplicada con valores por defecto genera que los valores de las variables originales sean agrupados en 10 clases, aquí el ejemplo de com queda la variable sepallength:

Graphical user interface, application

Description automatically generated

**4. Visualización**

* **Carga el conjunto de datos iris2.arff. Pulsa la pestaña Visualize. Aumenta Point Size a 5 para visualizar los datos mejor. Aumenta el valor de Jitter, ¿qué efecto tiene?**

Aumenta la dispersión de los valores de cada instancia de acuerdo a cada grupo de la variable clase :

Calendar

Description automatically generated with low confidence

**5. Clasificación**

**5.1. Clasificador ZeroR**

**Carga el conjunto de datos iris.arff. Selecciona el clasificador ZeroR y Use trainning set.**

* **¿Qué modelo genera el clasificador ZeroR?**

El modelo obtenido se basa en la moda de la variable clase.

* **¿Cuántas instancias del conjunto de entrenamiento clasifica bien?**

**50 instancias.**

* ¿Qué porcentaje de instancias clasifica bien?

El 33% de las instancias bien clasificadas.

Correctly Classified Instances 50 33.3333 %

* **¿Qué crees que indica la matriz de confusión?**

a b c <-- classified as

50 0 0 | a = Iris-setosa

50 0 0 | b = Iris-versicolor

50 0 0 | c = Iris-virginica

Sólo clasificó bien a las de la especie que tomo como modelo, al ser las tres clases iguales tomó la primera.

**5.2. Clasificador J48**

**Carga el conjunto de datos iris.arff. Selecciona el clasificador J48 y Use trainning set.**

* **¿Cuántas hojas tiene el árbol generado con J48?**

5 Hojas

* **¿Cuántas instancias del conjunto de entrenamiento clasifica bien?**

147 instancias.

* **¿Qué porcentaje de instancias clasifica bien?**

98%

* **Pulsar el botón de *More Options* y selecciona la opción *Output predictions*. ¿En qué instancias se ha equivocado?**

En las siguientes: 71, 107 y 130

* **Obtén el gráfico correspondiente al árbol generado.**

Diagram

Description automatically generated

* **¿Cómo podrías reducir el tamaño de este árbol en caso de que fuese necesario?**

Activando la opción ReducedErrorPrunning y cambiando el número del Folds a 2.

**5.3. Clasificador ID3**

**Carga el conjunto de datos iris.arff. Selecciona el clasificador ID3 y utilízalo para generar un árbol de decisión.**

* **¿Has podido ejecutar el algoritmo ID3 sobre el conjunto de datos directamente? ¿Por qué?**

No, porque el algoritmo exige que los atributos y la clase sean nominales.

* **¿Qué acciones has llevado a cabo para poder ejecutarlo?**

Para ejecutarlo en el preprocesamiento, se debe aplicar un cambio en los atributos numéricos aplicando el siguiente filtro:

Graphical user interface, application

Description automatically generated

* **¿Qué porcentaje de éxito sobre el conjunto de entrenamiento has obtenido?**

100%.

Correctly Classified Instances 150 100 %

* **¿Qué porcentaje de éxito obtienes si utilizas como mecanismo de evaluación la validación cruzada?**

77%

Correctly Classified Instances 116 77.3333 %