Universidad de Carabobo
Facultad Experimental de Ciencias y Tecnología
Departamento de Computación
Aprendizaje Maquina

Árbol de Decisión

Integrantes
Victor Mendoza CI: 21.476.548

2019, Julio

Estudio del Dataset

Haciendo uso del archivo *zoo.xls* el cual contiene información sobre los animales de un zoológico donde cada uno posee un 17 característica así como dos adicionales donde se indica el nombre del animal y la clase a la que pertenece, se busca realizar un árbol de decisión de 3 a 5 niveles sin utilizar herramientas computacionales que permita clasificarlos en siete clases:

- Mamífero (*Mammal*)
- Ave (*Bird*)
- Reptil (*Reptile*)
- Pez (*Fish*)
- Anfibio (*Amphibian*)
- Insecto (*Insect*)
- Invertebrado (*Invertebrate*)

Un análisis a simple vista del archivo mencionado anteriormente dio como resultado el árbol de decisión **(Fig. 1**) tomando en cuenta las características siguientes:

- *Feather*
- *Milk*
- *Predator*
- *Toothed*
- *Fins*
- *Tails*

Archivo ARFF

Un archivo ARFF (*Attribute-Relation File Format* - Formato de archivo de relación de atributo) es un archivo de texto ASCII que describe una lista de instancias que comparten un conjunto de atributos.
Utilizando el archivo *zoo.xls* se debe construir el archivo *zoo.arff* **(Fig. 2**) que sera utilizado en conjunto con la herramienta ***WEKA*** para su posterior análisis.

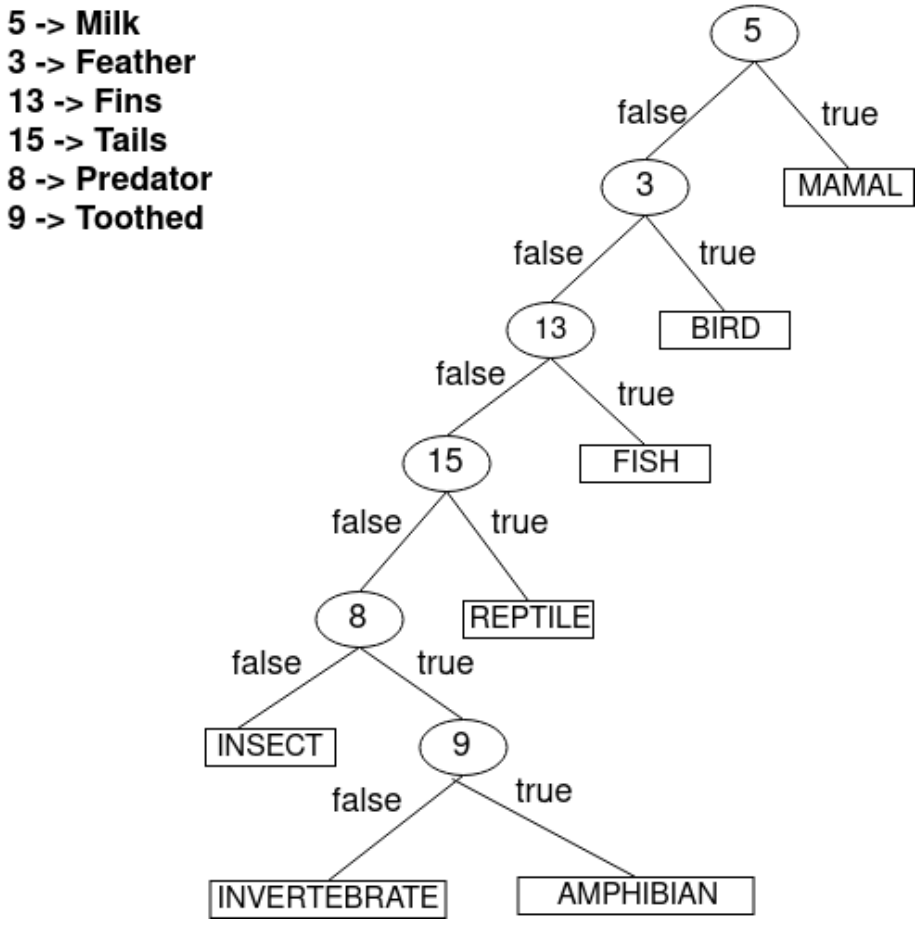

Fig. 1: Árbol de Decisión

WEKA

A continuación se realizara el estudio del dataset haciendo uso del archivo *zoo.arff* creado anteriormente.
Al iniciar el software ***WEKA*** **(Fig. 3**) se seleccionara la opción *explorer* para luego seleccionar el archivo *zoo.arff* en la opcion *Open file...* **(Fig. 4)**. En la interfaz **(Fig. 5)** se puede observar que existen 101 animales (*Instances*) registrados en el dataset.
Se seleccionara la opción del clasificador (*Classify*) en la parte superior de la interfaz para seleccionar el clasificador *J48* en la opción *Choose* **(Fig. 6)**. Una vez seleccionador el clasificador se selecciona el campo donde dice *J48* en donde se puede observar los distintos parámetros y una opción *More* la cual al ser seleccionada nos muestra todos los parámetros que posee el clasificador **(Fig.**

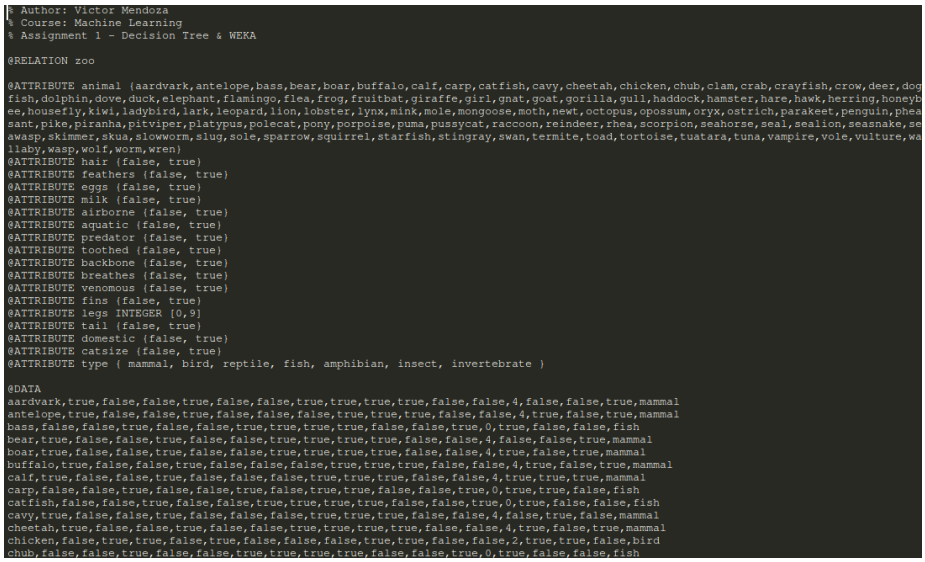

Fig. 2: Árchivo zoo.arff

7). Podemos observar que el clasificador J48 genera arboles de decisiones ya sea podados o no utilizando el algoritmo *C4.5* el cual es una variación del modelo de clasificación *ID3* [**1**], los parámetros que se utilizan en *C4.5* son:

- ***seed***: Utilizada para aleatorizar los datos cuando se utiliza ***reducedErrorPruning***
- ***unpruned***: Indica si se realiza poda
- ***confidenceFactor***: Factor utilizado en la poda, valores mas pequeños incurren en mas poda
- ***numFolds***: Indica la cantidad de datos usados por ***redurecErrorPruning***. Un pliegue se usar para podar, el resto para cultivar el árbol
- ***numDecimalPlaces***: Números de decimales que se utilizaran para la salida de números en el modelo
- ***batchSize***: Número preferido de instancias para procesar si se está realizando la predicción por lotes. Se pueden proporcionar más o menos instancias, pero esto le da a las implementaciones la oportunidad de especificar un tamaño de lote preferido
- ***reducedErrorPruning***: Indica si existe reducción de errores luego en lugar de la reduccion de *C4.5*
- ***useLaplace***: Indica si la cuenta de hojas es suavizada mediante Laplace

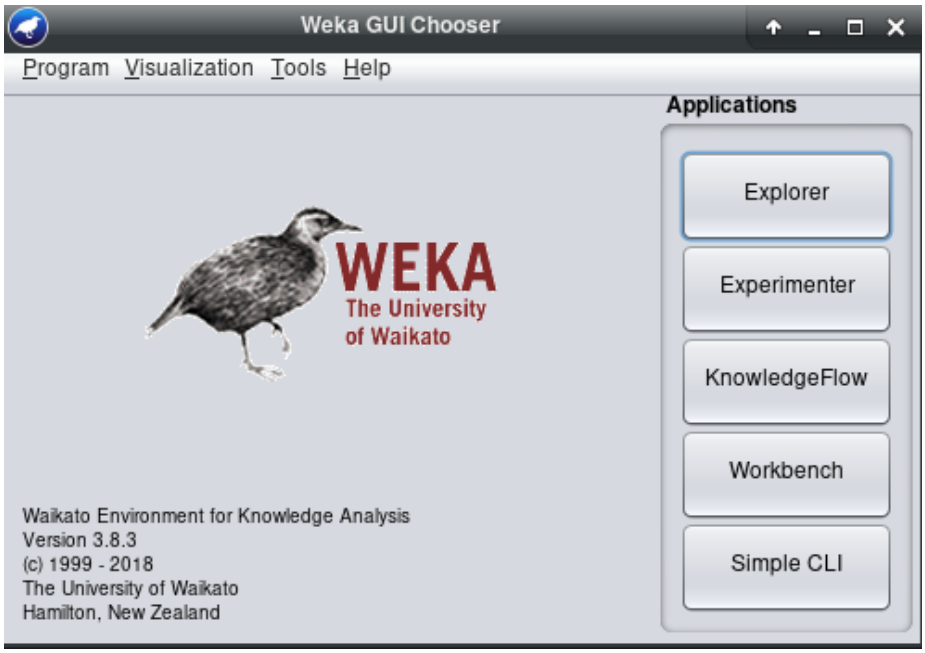

Fig. 3: Interfaz WEKA

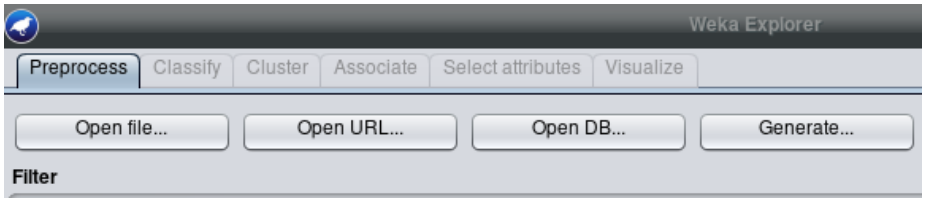

Fig. 4: Abrir archivo en WEKA

- ***doNotMakeSplitPointActualValue***: Si es verdadero, el punto de división no se reubica en un valor de datos real. Esto puede producir importantes incrementos de velocidad para grandes conjuntos de datos con atributos numéricos
- ***debug***: Si se establece en verdadero, el clasificador puede generar información adicional en la consola
- ***subtreeRaising***: Si considera la operación de aumento de subárbol al podar
- ***saveInstanceData***: Ya sea para guardar los datos de entrenamiento para la visualización

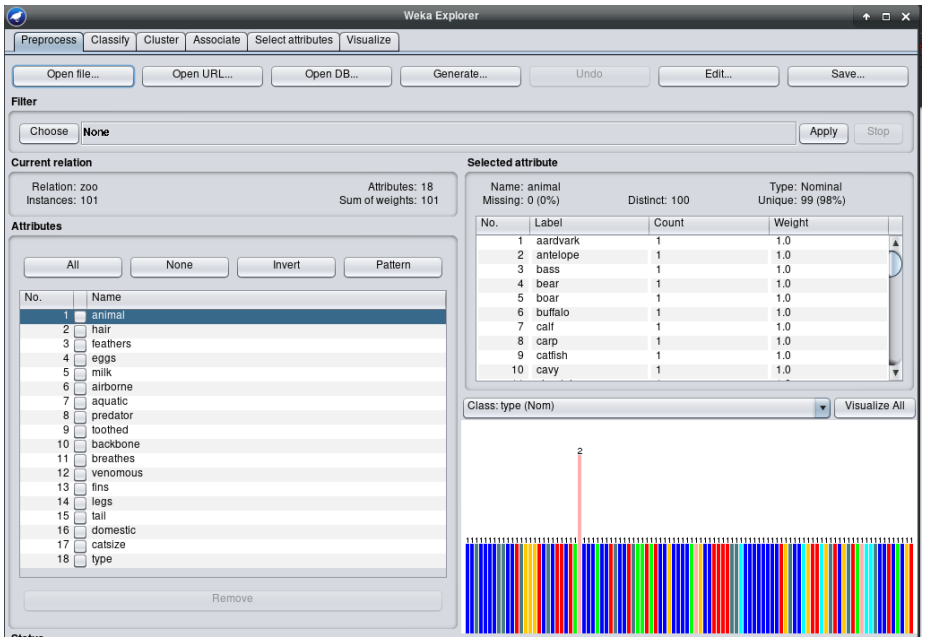

Fig. 5: Visualización del Dataset en WEKA

Fig. 6: Selección del Clasificador J48

- ***binarySplits***: Ya sea para usar divisiones binarias en atributos nominales al construir los árboles
- ***doNotCheckCapabilities***: Si se establece, las capacidades del clasificador no se verifican antes de que se construya el clasificador (puede reducir el tiempo de ejecución)
- ***minNumObj***: Mínimo numero de instancias por hoja
- ***useMDLcorrection***: Si la corrección MDL se usa al encontrar divisiones en atributos numéricos
- ***collapseTree***: Si se eliminan partes que no reducen el error de entrenamiento

Adicionalmente tenemos las opciones de prueba (*Test options*) **(Fig. 8)** con las siguientes opciones:

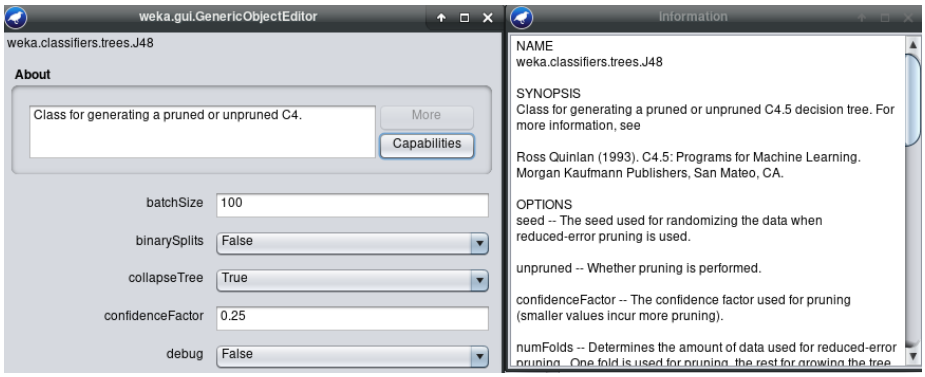

Fig. 7: Parámetros del J48

- ***Use training set***: Significa que pondrá a prueba sus conocimientos sobre los mismos datos que aprendió
- ***Supplied test set***: Es un archivo externo que puede utilizar como conjunto de entrenamiento
- ***Cross-validation Folds***: Divide los datos y separa el x% de los datos para el aprendizaje y el resto para la prueba
- ***Percentage Split***: Funciona como muchas divisiones porcentuales. Dobla los datos en 10 pliegues (por ejemplo) y repite 10 el siguiente proceso: Use 9 pliegues para aprender y deje 1 pliegue para la prueba

En el mismo menú de las opciones de pruebas al momento de seleccionar *More options* podemos encontrar las opciones de evaluación del clasificador **(Fig. 9)** con los que se puede conocer la eficiencia del clasificador ya sea observando la salida del modelo, de las divisiones de entrenamiento, estadísticas por clase, evaluación de las mediciones de la entropía, matriz de confusión, también podemos almacenar las predicciones para visualización, gráficas del error proporcional al margen, salida de la predicción, evaluación del cruce, estas son las características mas relevantes al momento evaluar al clasificador utilizando esta herramienta.

Estudio del Clasificador

Utilizando los parámetros por defecto y adicionando vistas adicionales del modelo para la salida se realizo la clasificación **(Fig. 10)** obteniendo un 92.0792% **(Fig. 11)** de clasificaciones correctas y 7.9208% clasificaciones incorrectas.
Podemos observar en la matriz de confusión **(Fig. 12)** las clases que fueron confundidas por el clasificador J48:

- La clase de reptiles fue confundida con los anfibios
- La clase pez fue confundida con la clase reptil

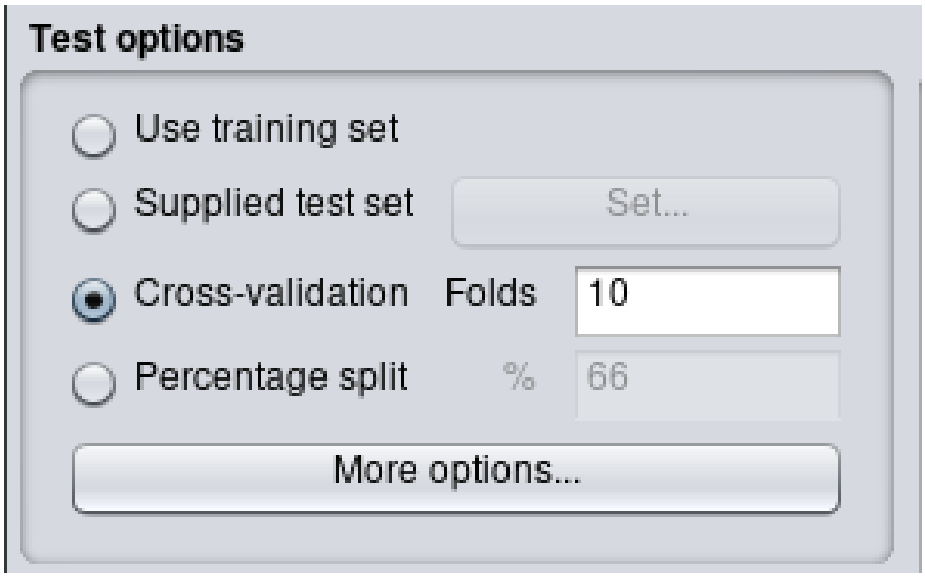

Fig. 8: Opciones de Prueba

- La clase insecto fue confundida con la clase reptil y la de invertebrados
- La clase invertebrados fue confundida la clase insecto

A continuación se realizo la clasificación colocando el parámetro *binarySplits* **(Fig. 13)** a *true*. Con este reajuste de parámetros se realizo la clasificación **(Fig. 14)** obteniendo un 91.0891% **(Fig. 15)** de clasificaciones correctas y 8.9109% clasificaciones incorrectas
Podemos observar en la matriz de confusión **(Fig. 16)** que con la variación de parámetro se confundieron las mismas clases con respecto al primer modelo realizado. Adicionalmente podemos observar una redacción en el porcentaje de clasificación correctas en el segundo modelo con respecto al primero .

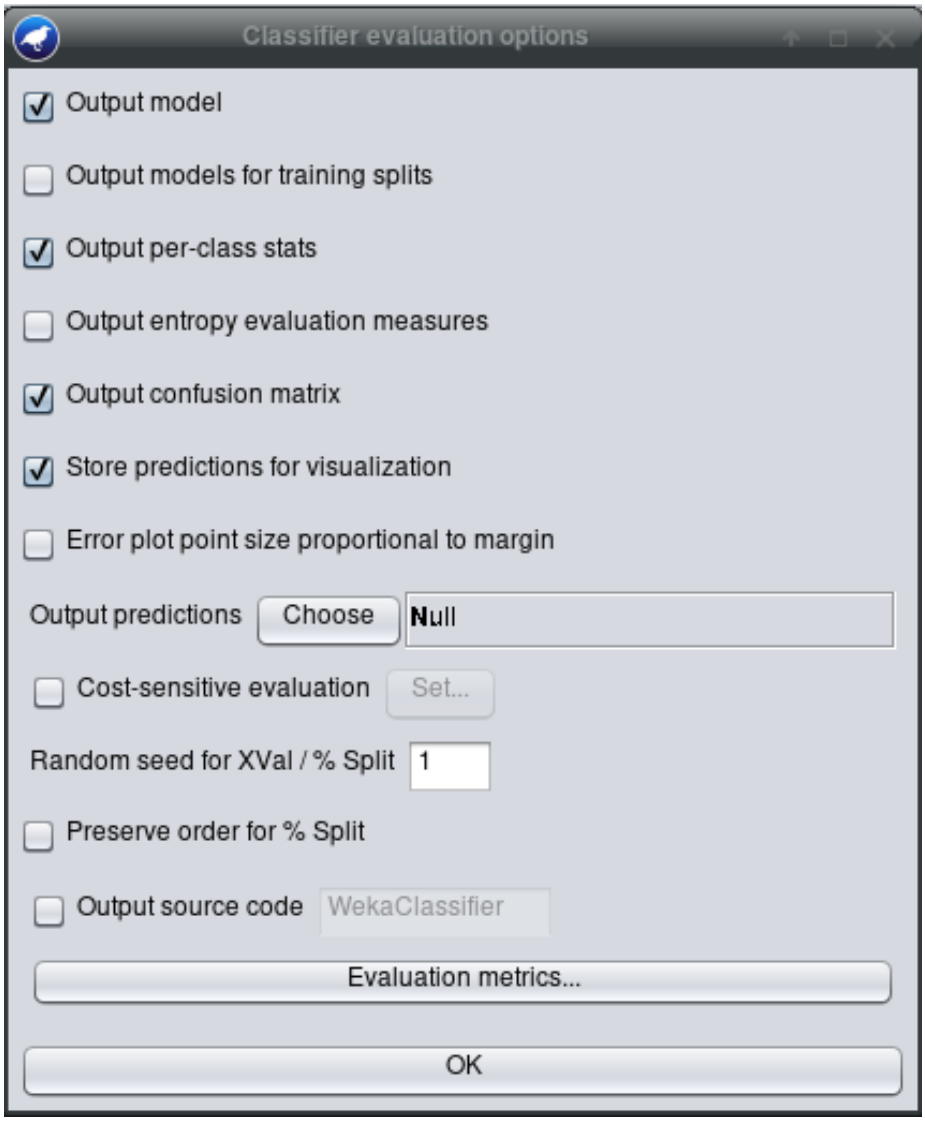

Fig. 9: Opciones de Evaluación del Clasificador

```
=== Classifier model for fold 10 ===

J48 pruned tree
------------------

feathers = false
|   milk = false
|   |   backbone = false
|   |   |   airborne = false
|   |   |   |   predator = false
|   |   |   |   |   legs <= 2: invertebrate (2.0)
|   |   |   |   |   legs > 2: insect (2.0)
|   |   |   |   predator = true: invertebrate (7.0)
|   |   |   airborne = true: insect (6.0)
|   |   backbone = true
|   |   |   fins = false
|   |   |   |   tail = false: amphibian (2.0)
|   |   |   |   tail = true: reptile (5.0/1.0)
|   |   |   fins = true: fish (12.0)
|   milk = true: mammal (37.0)
feathers = true: bird (18.0)

Number of Leaves  :     9

Size of the tree :      17
```

Fig. 10: Árbol Construido por el Clasificador

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          93               92.0792 %
Incorrectly Classified Instances         8                7.9208 %
Kappa statistic                          0.8955
K&B Relative Info Score                 87.9805 %
K&B Information Score                  214.4524 bits      2.1233 bits/instance
Class complexity | order 0             243.7499 bits      2.4134 bits/instance
Class complexity | scheme             4308.1767 bits     42.6552 bits/instance
Complexity improvement     (Sf)      -4064.4267 bits    -40.2418 bits/instance
Mean absolute error                      0.0225
Root mean squared error                  0.14
Relative absolute error                 10.2478 %
Root relative squared error             42.4398 %
Total Number of Instances              101
```

Fig. 11: Porcentaje de Clasificación

```
=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 41  0  0  0  0  0  0 |  a = mammal
  0 20  0  0  0  0  0 |  b = bird
  0  0  3  1  0  1  0 |  c = reptile
  0  0  0 13  0  0  0 |  d = fish
  0  0  1  0  3  0  0 |  e = amphibian
  0  0  0  0  0  5  3 |  f = insect
  0  0  0  0  0  2  8 |  g = invertebrate
```

Fig. 12: Matriz de Confusión para el Árbol Construido por el Clasificador

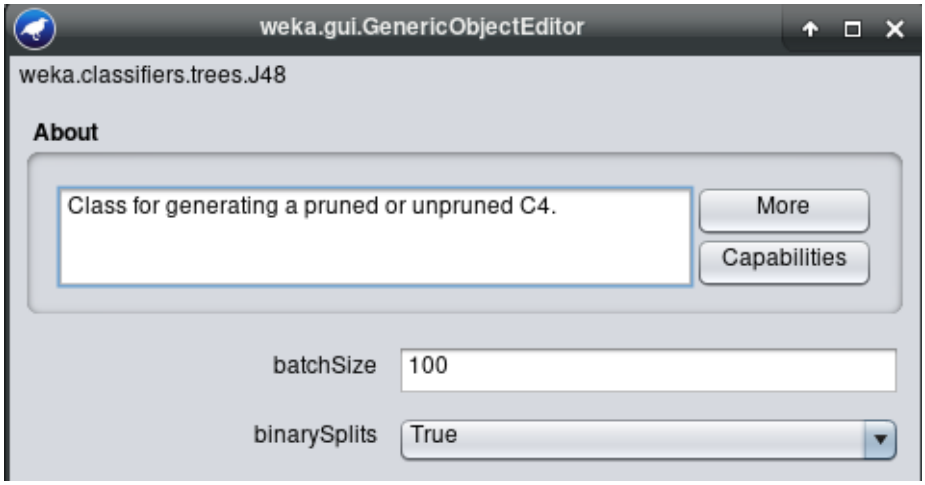

Fig. 13: Variación del Parámetro *binarySplits*

```
=== Classifier model for fold 10 ===

J48 pruned tree
------------------

feathers = false
|   milk = false
|   |   backbone = false
|   |   |   airborne = false
|   |   |   |   predator = false
|   |   |   |   |   legs <= 2.0: invertebrate (2.0)
|   |   |   |   |   legs > 2.0: insect (2.0)
|   |   |   |   predator != false: invertebrate (7.0)
|   |   |   airborne != false: insect (6.0)
|   |   backbone != false
|   |   |   fins = false
|   |   |   |   animal = frog: amphibian (2.0)
|   |   |   |   animal != frog: reptile (5.0/1.0)
|   |   |   fins != false: fish (12.0)
|   milk != false: mammal (37.0)
feathers != false: bird (18.0)

Number of Leaves  :     9

Size of the tree :      17
```

Fig. 14: Árbol Construido por el Clasificador Variando el Parámetro *binarySplits*

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          92               91.0891 %
Incorrectly Classified Instances         9                8.9109 %
Kappa statistic                          0.8824
K&B Relative Info Score                 87.0279 %
K&B Information Score                  212.1304 bits      2.1003 bits/instance
Class complexity | order 0             243.7499 bits      2.4134 bits/instance
Class complexity | scheme             4310.4986 bits     42.6782 bits/instance
Complexity improvement     (Sf)      -4066.7487 bits    -40.2648 bits/instance
Mean absolute error                      0.0247
Root mean squared error                  0.1463
Relative absolute error                 11.2801 %
Root relative squared error             44.3562 %
Total Number of Instances              101
```

Fig. 15: Porcentaje de Clasificación Variando el Parámetro *binarySplits*

```
=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 41  0  0  0  0  0  0 |  a = mammal
  0 20  0  0  0  0  0 |  b = bird
  0  0  3  1  0  1  0 |  c = reptile
  0  0  0 13  0  0  0 |  d = fish
  0  0  2  0  2  0  0 |  e = amphibian
  0  0  0  0  0  5  3 |  f = insect
  0  0  0  0  0  2  8 |  g = invertebrate
```

Fig. 16: Matriz de Confusión Variando el Parámetro *binarySplits*

Conclusión

Gracias a la herramienta *WEKA* podemos estudiar distintos conjuntos de datos con la posibilidad de pre procesador los datos para luego construir distintos modelos los cuales pueden basarse en un conjunto de clasificadores con diferentes parámetros los cuales pueden ser variados de acuerdo a las necesidades que se buscan. Adicionalmente existen diversas formas de poder probar los clasificadores generados ya sea con el mismo conjunto de datos de entrenamiento o utilizando unos externos para comprobar la eficiencia de estos con datos desconocidos.

En este caso se trabajo con el clasificador *J48* el cual nos permite generar un modelo de clasificación haciendo uso de la técnica de arboles de decisiones, este clasificador utilizado se basa en el algoritmo *C4.5* el cual es una variación del algoritmo *ID3*. En el primer modelo construido se pudo comprar que la eficiencia de clasificación con respecto al mismo conjunto de datos con el cual fue entrenado fue ≈ 93 mientras que el segundo modelo el cual se le vario el parámetro *binarySplits* obtuvo ≈ 92.

A pesar de que ambos modelos obtuvieron una eficiencia relativamente buena cabe destacar que esta se realizo con los mismo datos de entrenamiento que permitieron generar los modelos, por lo cual la eficiencia con respecto a nuevos datos fuera del conjunto de entrenamiento no necesariamente sera la misma.

Adicionalmente se puede observar que ambos modelos a pesar de tener una eficiencia alta la matriz de confusión nos permite observar que algunas de las clases de clasificación que posee ambos modelos suelen confundirse generando así cierto margen de error o confusión entre datos de clasificación mas complejos. Otra cosa que se puede observar en la matriz es que los clasificadores no necesariamente clasificaran toco el conjunto de datos.

Por ultimo podemos destacar que la herramienta utilizada presenta una gran cantidad de características y modelos que nos permite análisis, clasificar y probar los diferentes clasificadores siendo así de gran utilidad al momento de seleccionar un modelo adecuado respecto a los datos que se estén trabajando, esto debido a que nos permite variar los parámetros de cada clasificador así de poder observar como varían en cada iteración haciendo capaz de poder buscar y optimizar de cierta manera los clasificadores a utilizar.

Referencias

[1] Steven L. Salzberg. C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16(3):235–240, Sep 1994.