

PREGUNTA 4

Del anterior problema en PYTHON y WEKA realice un árbol de decisión basado en entropía, compare los resultados, esto debe repetirse al menos 10 veces y comparar las matrices de confusión y obtener la confiabilidad. Se recomienda comparar media y mediana de los resultados.

Classifier

Choose **J48 - C 0.25 - M 2**

Test options

☐ Use training set

☐ Supplied test set **Set...**

☒ Cross-validation **Folds** **10**

☐ Percentage split **%** **66**

More options...

(Nom) price_range

Start **Stop**

Result list (right-click for options)

- 21:12:16 - trees.J48
- 21:12:18 - trees.J48
- 21:12:20 - trees.J48
- 21:12:27 - trees.J48
- 21:12:36 - trees.J48
- 21:13:08 - trees.J48
- 21:13:13 - trees.J48
- 21:27:57 - trees.J48
- 21:27:59 - trees.J48
- 21:28:02 - trees.J48
- 21:28:03 - trees.J48
- 21:28:05 - trees.J48
- 21:28:07 - trees.J48

Classifier output

```

=== Summary ===

Correctly Classified Instances      1685           84.25 %
Incorrectly Classified Instances    315           15.75 %
Kappa statistic                    0.79
Mean absolute error                0.0849
Root mean squared error            0.2707
Relative absolute error            22.6434 %
Root relative squared error        62.5265 %
Total Number of Instances         2000

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
      -----  -
      0,790    0,063    0,806    0,790    0,798      0,732    0,892    0,723    uno
      0,764    0,071    0,783    0,764    0,773      0,699    0,878    0,707    dos
      0,896    0,042    0,877    0,896    0,886      0,848    0,950    0,859    tres
      0,920    0,034    0,900    0,920    0,910      0,880    0,960    0,877    cero
Weighted Avg.   0,843    0,053    0,841    0,843    0,842      0,790    0,920    0,791

=== Confusion Matrix ===

  a  b  c  d  <-- classified as
395  54  0  51 |  a = uno
 55 382  63  0 |  b = dos
  0  52 448  0 |  c = tres
 40   0  0 460 |  d = cero
    
```

Status

OK **Log** x 0

Con weka obtenemos un nivel del 84,25% de clasificaciones correctas.

Mientras que con python obtenemos una exactitud del 0,85%, lo que nos muestra que los resultados no estan muy distantes.

```
[[140 29  0  0]
 [ 19 120 19  5]
 [  1  9 139  9]
 [  0  1 10 159]]
```

Exactitud: 0.85

Informe de clasificación

	precision	recall	f1-score	support
Clase 0	0.88	0.83	0.85	169
Clase 1	0.75	0.74	0.75	163
Clase 2	0.83	0.88	0.85	158
Clase 3	0.92	0.94	0.93	170
accuracy			0.85	660
macro avg	0.84	0.84	0.84	660
weighted avg	0.85	0.85	0.84	660

Ahora repitiendo el proceso 10 veces en Python obtenemos la siguiente tabla.

#	Exactitud
1	0.85
2	0.87
3	0.87
4	0.86
5	0.86
6	0.87
7	0.86
8	0.86
9	0.84
10	0.88
MEDIA	0.862

Ordenando la tabla para obtener la mediana:

#	Exactitud
1	0.84
2	0.85
3	0.86
4	0.86
5	0.86
6	0.86
7	0.87
8	0.87
9	0.87
10	0.88
MEDIA	0.862



MEDIANA

Como se puede observar la media y la mediana son muy similares, por lo cual diríamos que tenemos un 86% de confiabilidad para predecir datos.