

Informe sobre **segment-challenge.arff**

La siguiente información describe el contenido del documento:

1. Title: Image Segmentation data

2. Source Information

-- Creators: Vision Group, University of Massachusetts

-- Donor: Vision Group (Carla Brodley, brodley@cs.umass.edu)

-- Date: November, 1990

3. Past Usage: None yet published

4. Relevant Information:

The instances were drawn randomly from a database of 7 outdoor images. The images were handsegmented to create a classification for every pixel.

Each instance is a 3x3 region.

5. Number of Instances: Training data: 210 Test data: 2100

6. Number of Attributes: 19 continuous attributes

7. Attribute Information:

1. region-centroid-col: the column of the center pixel of the region.
2. region-centroid-row: the row of the center pixel of the region.
3. region-pixel-count: the number of pixels in a region = 9.
4. short-line-density-5: the results of a line extractoin algorithm that counts how many lines of length 5 (any orientation) with low contrast, less than or equal to 5, go through the region.
5. short-line-density-2: same as short-line-density-5 but counts lines of high contrast, greater than 5.
6. vedge-mean: measure the contrast of horizontally adjacent pixels in the region. There are 6, the mean and standard deviation are given. This attribute is used as a vertical edge detector.
7. vegde-sd: (see 6)
8. hedge-mean: measures the contrast of vertically adjacent pixels. Used for horizontal line detection.

9. hedge-sd: (see 8).
10. intensity-mean: the average over the region of $(R + G + B)/3$
11. rawred-mean: the average over the region of the R value.
12. rawblue-mean: the average over the region of the B value.
13. rawgreen-mean: the average over the region of the G value.
14. exred-mean: measure the excess red: $(2R - (G + B))$
15. exblue-mean: measure the excess blue: $(2B - (G + R))$
16. exgreen-mean: measure the excess green: $(2G - (R + B))$
17. value-mean: 3-d nonlinear transformation
of RGB. (Algorithm can be found in Foley and VanDam, Fundamentals
of Interactive Computer Graphics)
18. saturatoin-mean: (see 17)
19. hue-mean: (see 17)
8. Missing Attribute Values: None
9. Class Distribution:
Classes: brickface, sky, foliage, cement, window, path, grass.
30 instances per class for training data.
300 instances per class for test data.

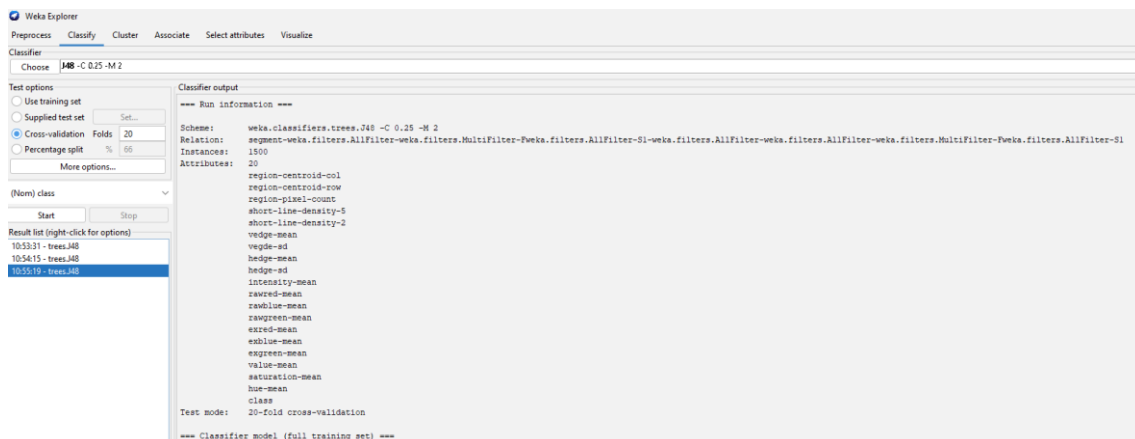
Comenzamos el análisis de diferentes modelos.

El primer modelo elegido:

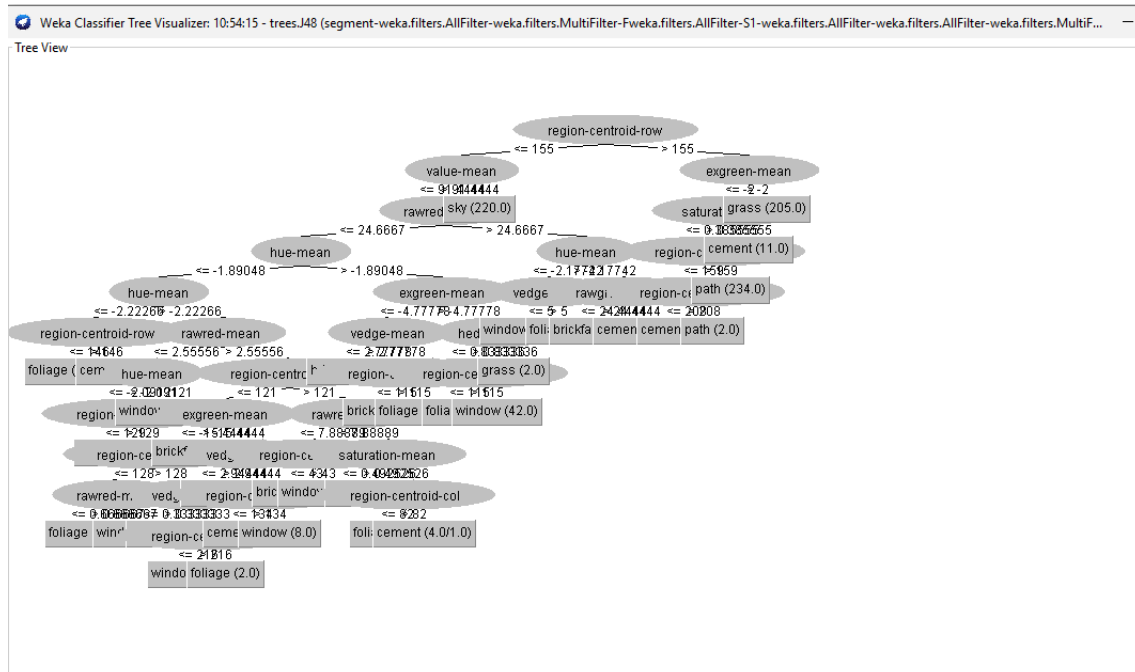
Trees-J 48 para la construcción de un árbol de decisión.

En las opciones de test elegí la opción Cross-validation con Folds=20.

Obtenemos la siguiente información:



En la visualización del árbol obtenemos:



La visualización es intuitiva y fácil de entender.

En el Summary podemos ver como la probabilidad de error es baja, de tan solo un 3.8% de error y un 96.2% de acierto lo que ya nos deja un muy buen valor

```

=== Summary ===

Correctly Classified Instances      1443          96.2  %
Incorrectly Classified Instances    57           3.8  %
Kappa statistic                    0.9556
Mean absolute error                 0.0124
Root mean squared error             0.0989
Relative absolute error             5.0838 %
Root relative squared error        28.2829 %
Total Number of Instances          1500

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
0,966  0,005  0,971  0,966  0,968  0,963  0,988  0,963  brickface
1,000  0,001  0,995  1,000  0,998  0,997  1,000  0,995  sky
0,933  0,011  0,933  0,933  0,933  0,922  0,973  0,914  foliage
0,950  0,011  0,937  0,950  0,944  0,934  0,977  0,947  cement
0,897  0,017  0,893  0,897  0,895  0,878  0,965  0,874  window
0,992  0,000  1,000  0,992  0,996  0,995  1,000  1,000  path
0,990  0,000  1,000  0,990  0,995  0,994  1,000  1,000  grass
Weighted Avg.  0,962  0,006  0,962  0,962  0,962  0,956  0,987  0,957

=== Confusion Matrix ===

  a  b  c  d  e  f  g  <-- classified as
198  0  1  3  3  0  0 |  a = brickface
  0 220  0  0  0  0  0 |  b = sky
  1  1 194  2 10  0  0 |  c = foliage
  2  0  2 209  7  0  0 |  d = cement
  3  0 11  7 183  0  0 |  e = window
  0  0  0  2  0 234  0 |  f = path
  0  0  0  0  2  0 205 |  g = grass

```

En el Summary podemos ver la validación cruzada que nos da información sobre el rendimiento del modelo de clasificación.

Correctly Classified Instances nos dice que el modelo clasifico correctamente 1443 instancias, que representa un 96.2% de precisión, lo que indica un buen rendimiento general.

Incorrectly Classified Instances, al contrario que el anterior nos dice que el modelo cometió errores en la clasificación de 57 instancias, que representa un 3.8% de error.

Estos datos son muy buenos y apunta a ser un buen modelo elegido

En la Confusión Matrix vemos la cantidad de instancias que están clasificadas correctamente en la diagonal, aunque algunas instancias de la clase a fueron clasificadas incorrectamente como c, d y e.

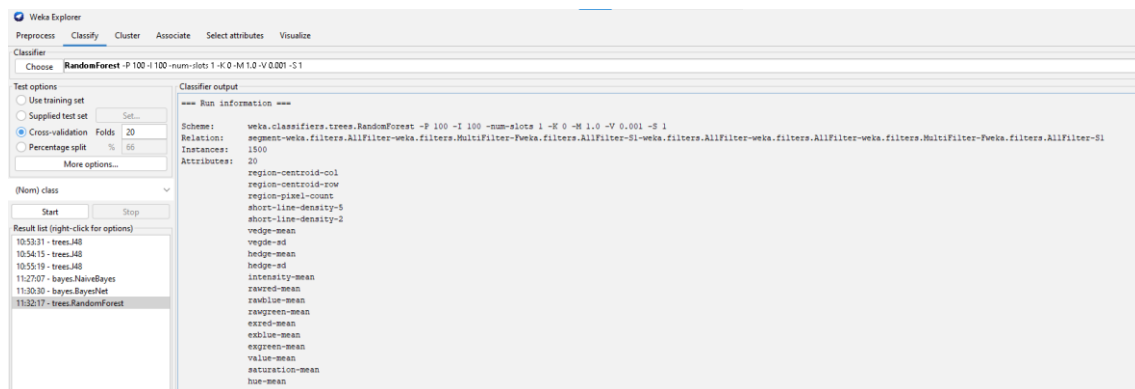
En conclusión, el resultado para este modelo es muy bueno tiene una alta precisión y un buen rendimiento en la clasificación de las instancias

El segundo modelo elegido:

Random Forest, a diferencia del anterior combina muchos árboles de decisión al hacer la clasificación.

En las opciones de test elegí la opción Cross-validation con Folds=20.

Obtenemos la siguiente información:



Del Summary concluimos lo siguiente:

```

=== Summary ===

Correctly Classified Instances      1469           97.9333 %
Incorrectly Classified Instances      31           2.0667 %
Kappa statistic                    0.9759
Mean absolute error                  0.0189
Root mean squared error              0.0761
Relative absolute error              7.7343 %
Root relative squared error         21.7612 %
Total Number of Instances          1500

=== Detailed Accuracy By Class ===

```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1,000	0,001	0,995	1,000	0,998	0,997	1,000	1,000	brickface
	1,000	0,001	0,995	1,000	0,998	0,997	1,000	1,000	sky
	0,962	0,006	0,962	0,962	0,962	0,955	0,998	0,989	foliage
	0,964	0,007	0,959	0,964	0,961	0,955	0,999	0,994	cement
	0,931	0,008	0,945	0,931	0,938	0,929	0,997	0,983	window
	0,996	0,001	0,996	0,996	0,996	0,995	1,000	1,000	path
	1,000	0,000	1,000	1,000	1,000	1,000	1,000	1,000	grass
Weighted Avg.	0,979	0,003	0,979	0,979	0,979	0,976	0,999	0,995	

```

=== Confusion Matrix ===

 a  b  c  d  e  f  g  <-- classified as
205  0  0  0  0  0  0 | a = brickface
 0 220  0  0  0  0  0 | b = sky
 0  1 200  2  5  0  0 | c = foliage
 0  0  1 212  6  1  0 | d = cement
 1  0  7  6 190  0  0 | e = window
 0  0  0  1  0 235  0 | f = path
 0  0  0  0  0  0 207 | g = grass

```

En el Summary podemos ver la validación cruzada que nos da información sobre el rendimiento del modelo de clasificación.

Correctly Classified Instances nos dice que el modelo clasifico correctamente 1469 instancias, que representa un 97.9% de precisión, indicando así un muy buen rendimiento general. Mejor que el anterior modelo.

Incorrectly Classified Instances, al contrario que el anterior nos dice que el modelo cometió errores en la clasificación de tan solo 31 instancias, que representa un 2.06% de error.

En la Confusión Matrix vemos la cantidad de instancias que están clasificadas correctamente en la diagonal, y con muy pocos errores. La tasa de falsos positivos es mínima.

En conclusión, el resultado para este modelo es muy bueno. El modelo Random Forest clasifica bien la mayoría de las instancias y concuerda con las clasificaciones reales.

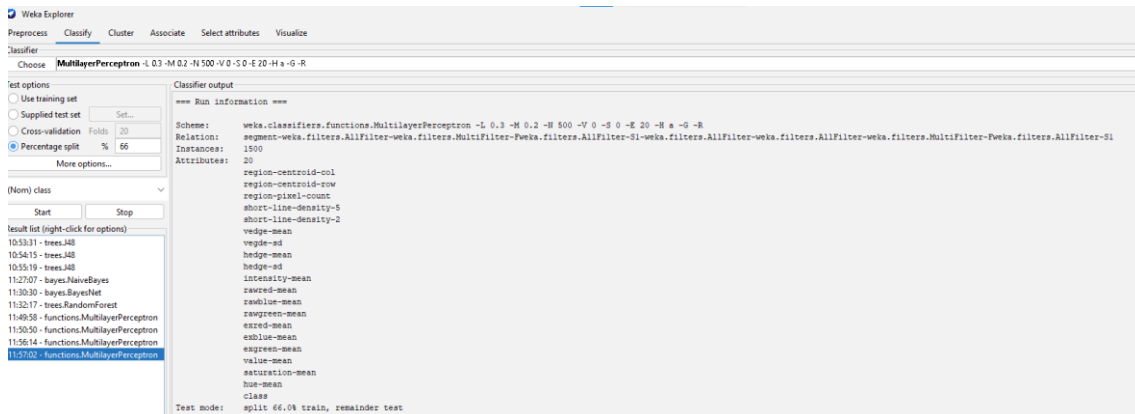
Así este modelo es más confiable que el anterior analizado.

El tercer modelo elegido:

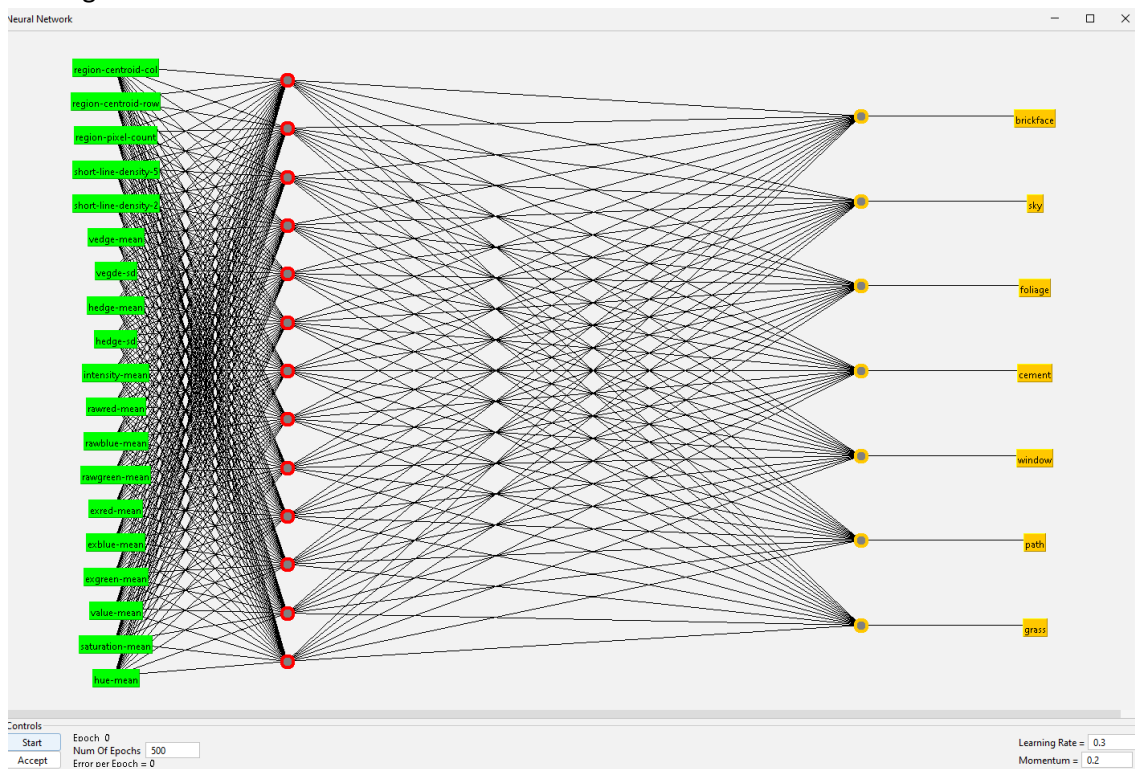
Multilayer Perceptron, es una Red Neuronal Artificial (RNA) es un modelo computacional que pretende simular el funcionamiento del cerebro. Se utiliza para resolver problemas de varios tipos, incluyendo problemas de predicción de series de tiempo, regresión y clasificación entre otros.

En las opciones de test elegí la opción Percentage Split 66%

Obtenemos la siguiente información:



Y la siguiente red neuronal:



Del Summary concluimos lo siguiente:

```

=== Summary ===

Correctly Classified Instances      79          15.4902 %
Incorrectly Classified Instances    431          84.5098 %
Kappa statistic                     0
Mean absolute error                 0.2449
Root mean squared error             0.35
Relative absolute error             100.0313 %
Root relative squared error         100.0259 %
Total Number of Instances          510

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
0,000  0,000  ?      0,000  ?      ?      0,607  0,165  brickface
0,000  0,000  ?      0,000  ?      ?      0,220  0,098  sky
0,000  0,000  ?      0,000  ?      ?      0,291  0,097  foliage
1,000  1,000  0,155  1,000  0,268  ?      0,533  0,154  cement
0,000  0,000  ?      0,000  ?      ?      0,550  0,138  window
0,000  0,000  ?      0,000  ?      ?      0,631  0,207  path
0,000  0,000  ?      0,000  ?      ?      0,082  0,079  grass
Weighted Avg.  0,155  0,155  ?      0,155  ?      ?      0,411  0,134

=== Confusion Matrix ===

 a  b  c  d  e  f  g  <-- classified as
0  0  0  64  0  0  0  | a = brickface
0  0  0  81  0  0  0  | b = sky
0  0  0  73  0  0  0  | c = foliage
0  0  0  79  0  0  0  | d = cement
0  0  0  60  0  0  0  | e = window
0  0  0  80  0  0  0  | f = path
0  0  0  73  0  0  0  | g = grass

```

Correctly Classified Instances nos dice que el modelo clasifico correctamente 79 instancias, que representa un 15.5% de precisión, indicando así un rendimiento nefasto. Lo que indica que el modelo no logra clasificar correctamente la mayoría de las instancias.

Incorrectly Classified Instances, el modelo cometió errores en la clasificación de tan solo 431 instancias, que representa un 84.5% de error.

En la Confusión Matrix, vemos que el modelo clasifica todas las instancias en la clase cement y no consigue distinguir las demás clases.

Como conclusión este modelo basado en la red neuronal no es efectivo para este tipo de datos, ya que tiene un bajo rendimiento y clasifica todas las instancias en una única clase.

Conclusión final

Como conclusión final de los tres modelos analizados, el modelo Random Forest clasifica bien la mayoría de las instancias y concuerda con las clasificaciones reales.

Así este modelo es el más confiable de los anteriores analizados.