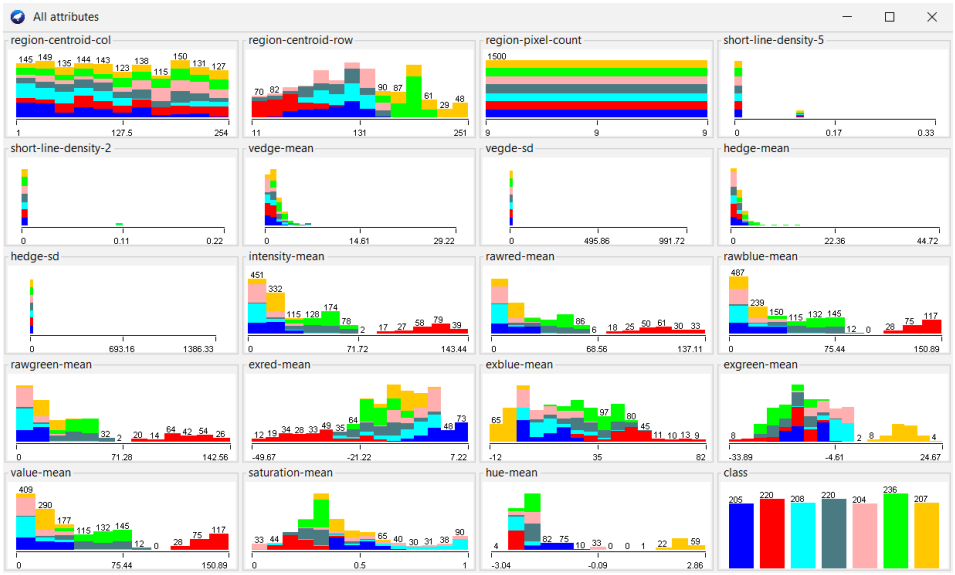
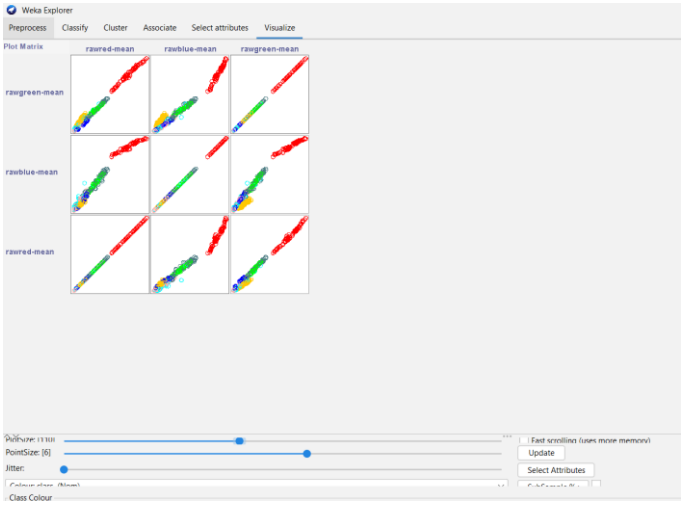
ACTIVITATS :	
ACTIVITAT 1 (5 punts) Temps màxim 1 hora	Analitza el conjunt de dades **segment-challenge.arff** amb **Weka**, i després de provar 3 models troba el millor per poder fer la predicció. Genera un **informe.docx** mostrant comentant els resultats dels 3 models i justificant quin et sembla millor. Ha d'incloure algun gràfic que et sembli significatiu.

Comentarios: Comprobamos que, en este dataset, tenemos un total de 1500 instances (registros) con un total de 20 atributos. El último es el resultado donde comprobamos que hay 7 clases definidas. Todas las clases tienen entre 204 y 220 datos.

Revisando todos los atributos, vemos que hay bastante diferencia entre los valores máximos y mínimos en los atributos.

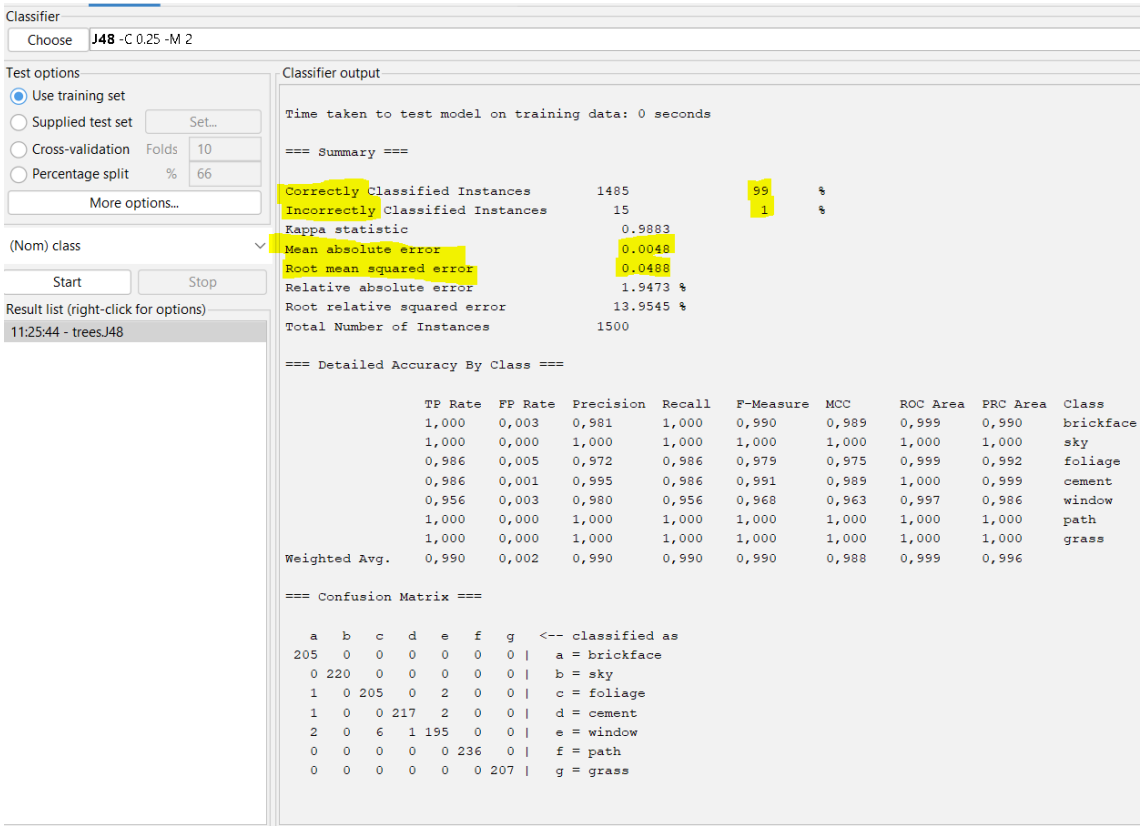


Revisando los atributos raw, he visto que sus distribuciones y sus valores máximos y mínimos son similares por lo que lo he revisado en el visualize también y se confirma que tienen relación.



Ahora vamos a revisar qué algoritmos serían mejor para aplicar a nuestro modelo y predecir los resultados.

- **El primero que vamos a analizar es el modelo de árbol J48** que es un sistema para crear un árbol de clasificación y que nos permite predecir qué el resultado.
  Vamos a utilizar únicamente el set de entrenamiento. Observamos que el error absoluto medio (media aritmética del valor absoluto de todos los errores) es muy reducido (0,0048) y que el RMSE (Root Mean Squared Error) es también reducido (0,0488). El RMSE nos mide la diferencia entre los valores predichos y los reales. No obstante, esto es solo para los valores de entrenamiento del modelo.



Si revisamos los datos con el set de test o realizando un cros validation, vemos que el porcentaje de acierto disminuye.

- Cross-validation

Classifier output

```
=== Summary ===

Correctly Classified Instances        1436               95.7333 %
Incorrectly Classified Instances        64                4.2667 %
Kappa statistic                          0.9502
Mean absolute error                      0.0138
Root mean squared error                  0.1057
Relative absolute error                  5.6471 %
Root relative squared error             30.2115 %
Total Number of Instances             1500

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0,956    0,004    0,975      0,956    0,966      0,960
                 1,000    0,001    0,995      1,000    0,998      0,997
                 0,942    0,018    0,895      0,942    0,918      0,905
                 0,941    0,009    0,945      0,941    0,943      0,933
                 0,877    0,017    0,891      0,877    0,884      0,866
                 0,987    0,001    0,996      0,987    0,991      0,990
                 0,990    0,000    1,000      0,990    0,995      0,994
Weighted Avg.    0,957    0,007    0,958      0,957    0,957      0,951

=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
 196   0   3   1   5   0   0 |   a = brickface
   0 220   0   0   0   0   0 |   b = sky
   0   1 196   2   9   0   0 |   c = foliage
   2   0   4 207   6   1   0 |   d = cement
   3   0  16   6 179   0   0 |   e = window
   0   0   0   3   0 233   0 |   f = path
   0   0   0   0   2   0 205 |   g = grass
```

- Datos de test.

Classifier output

```
=== Summary ===

Correctly Classified Instances         779               96.1728 %
Incorrectly Classified Instances        31                3.8272 %
Kappa statistic                          0.9553
Mean absolute error                      0.0127
Root mean squared error                  0.1005
Relative absolute error                  5.1771 %
Root relative squared error             28.6807 %
Total Number of Instances              810

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0,992    0,004    0,976      0,992    0,984      0,981
                 1,000    0,000    1,000      1,000    1,000      1,000
                 0,975    0,019    0,902      0,975    0,937      0,926
                 0,973    0,010    0,939      0,973    0,955      0,948
                 0,833    0,007    0,955      0,833    0,890      0,874
                 1,000    0,003    0,979      1,000    0,989      0,988
                 0,976    0,001    0,992      0,976    0,984      0,981
Weighted Avg.    0,962    0,007    0,962      0,962    0,961      0,955

=== Confusion Matrix ===

   a   b   c   d   e   f   g   <-- classified as
 124   0   0   0   1   0   0 |   a = brickface
   0 110   0   0   0   0   0 |   b = sky
   1   0 119   0   2   0   0 |   c = foliage
   1   0   0 107   2   0   0 |   d = cement
   1   0  12   7 105   0   1 |   e = window
   0   0   0   0   0  94   0 |   f = path
   0   0   1   0   0   2 120 |   g = grass
```

Estos serían los datos juntos:

MODELO	Correctly Classified Instances	Incorrectly Classified Instances	Correctly Classified Instances (%)	Mean absolute error	Root mean squared error	Total Number of Instances
Trees J48 - Training Set	1485	15	99,00%	0,0048	0,0488	1500
Trees J48 - Supplied test set	779	31	96,17%	0,0138	0,1057	810
Trees J48 - Cross-Validation (10 folds)	1436	64	95,73%	0,0127	0,1005	1500
Regressió logística simple - Training Set	1445	55	96,33%	0,0182	0,0877	1500
Regressió logística simple - Supplied test set	761	49	93,95%	0,0232	0,1098	810
Regressió logística simple - Cross-Validation (10 folds)	1427	73	95,13%	0,0197	0,0963	1500
Regressió logística - Training Set	1454	46	96,93%	0,0127	0,0777	1500
Regressió logística - Supplied test set	759	51	93,70%	0,0202	0,1131	810
Regressió logística - Cross-Validation (10 folds)	1441	59	96,07%	0,016	0,0989	1500
Multilayer Perceptron - Training Set	1459	41	97,27%	0,0108	0,079	1500
Multilayer Perceptron - Supplied test set	761	49	93,95%	0,02	0,1172	810
Multilayer Perceptron - Cross-Validation (10 folds)	1456	44	97,07%	0,0135	0,0841	1500

Dados estos datos y comparando los modelos de árbol J48, regresión logística (Simple y normal) y perceptron, observamos que, pese a tener buenos datos de entrenamiento, en el test se reduce. Además, sorprende que, en el J48, las instancias correctas en el test sean mayores que en el cros validation, hecho que pasa al contrario en el resto de modelos. Por tanto, conforme más folds pongamos para el resto de modelos, mejor clasificación va a tener.

Teniendo en cuenta estos datos, creo que el modelo multilayer perceptron sería el más adecuado.