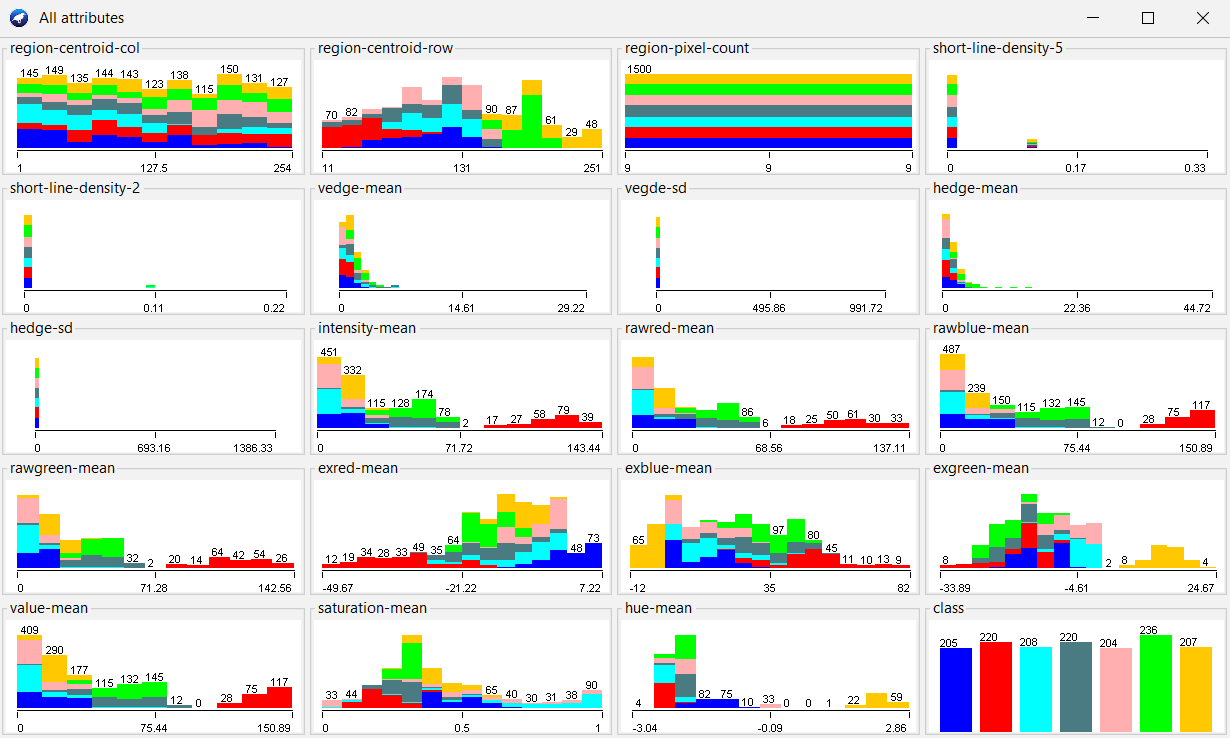


**Comentarios:** Comprobamos que, en este dataset, tenemos un total de 1500 instances (registros) con un total de 20 atributos. El último es el resultado donde comprobamos que hay 7 clases definidas. Todas las clases tienen entre 204 y 220 datos.

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



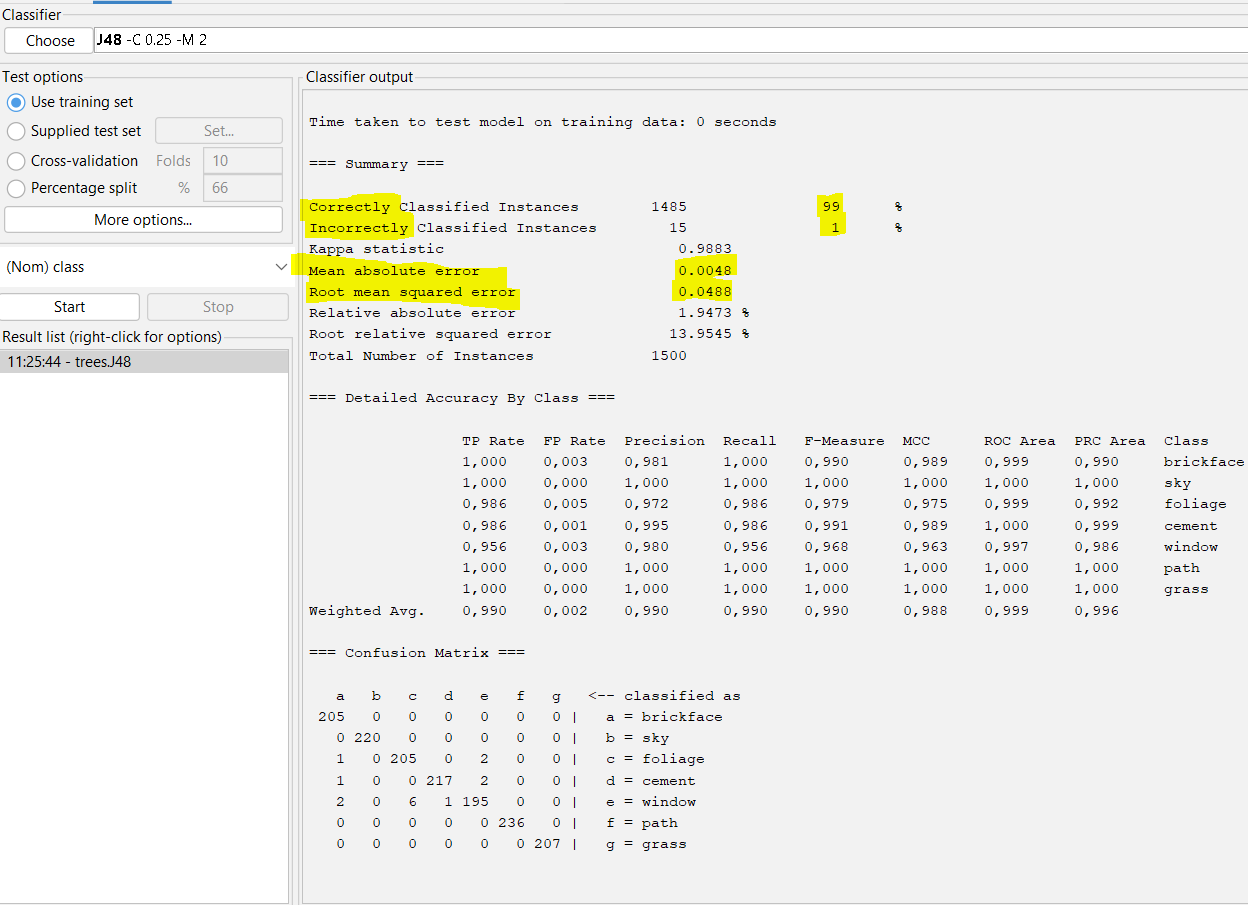
Revisando los atributos raw, he visto que sus distribuciones y sus valores máximos y mínimos son similares por lo que lo he revisado en el visualize también y se confirma que tienen relación.



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

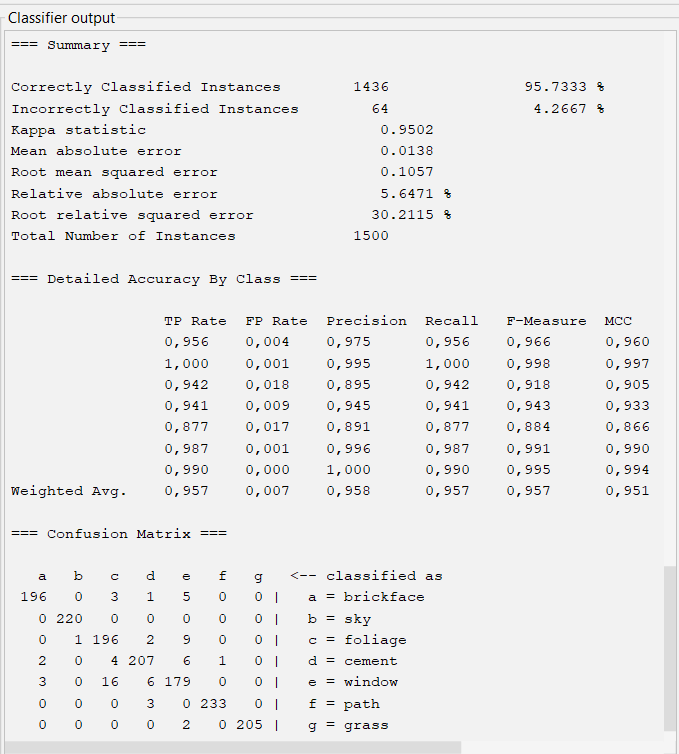
* **El primero que vamos a analizar es el modelo de árbol J48** que es un sistema para crear un árbol de clasificación y que nos permite predecir qué el resultado.

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.

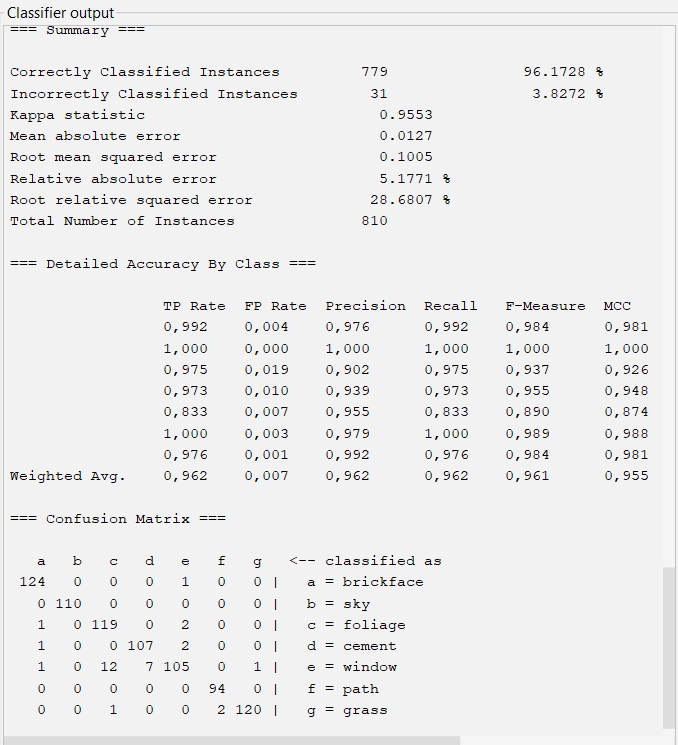


Si revisamos los datos con el set de test o realizando un cros validation, vemos que el porcentaje de acierto disminuye.

* Cross-validation



* Datos de test.



Estos serían los datos juntos:



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.