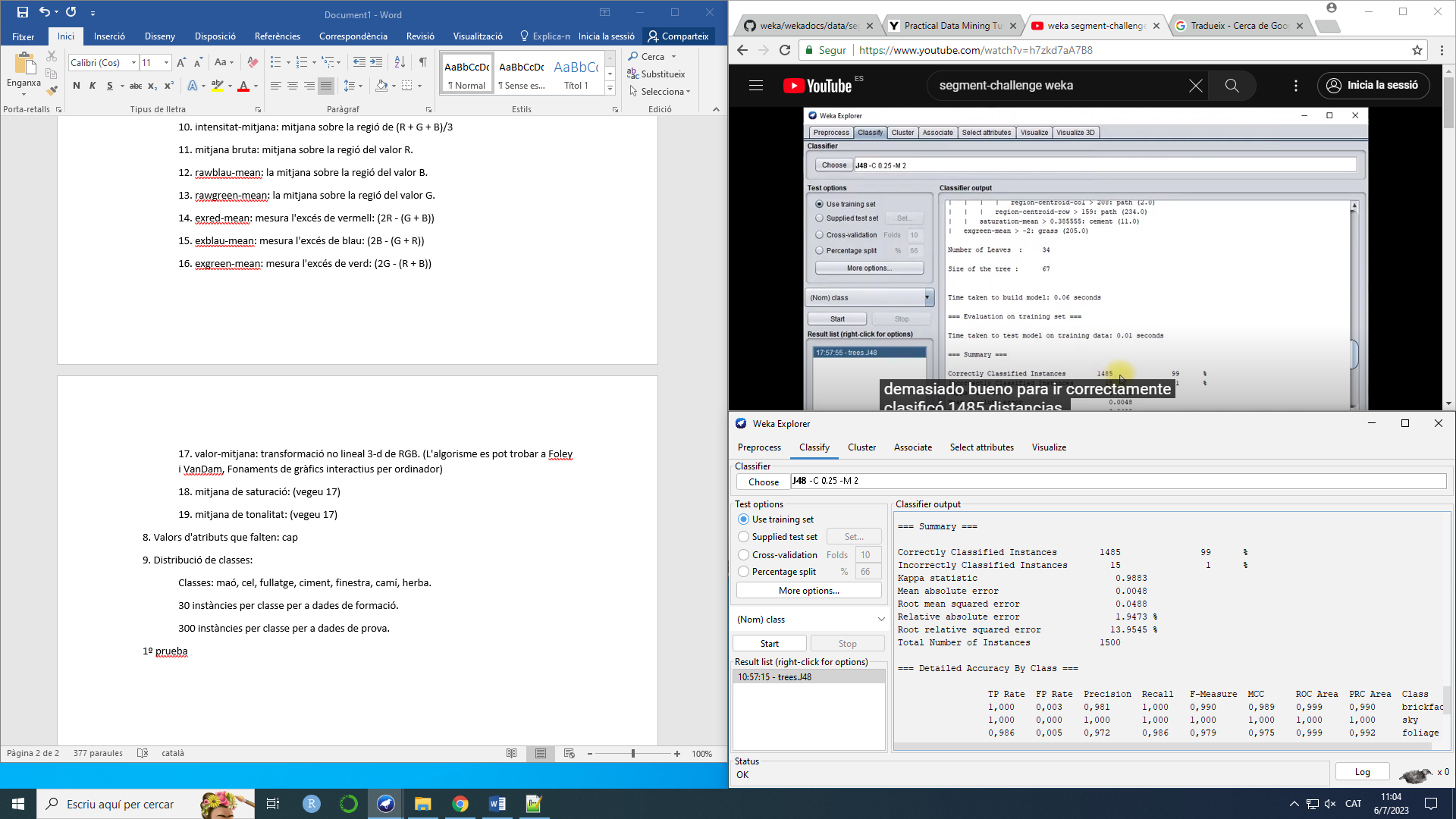
1. Título: Datos de segmentación de imágenes
2. .Información relevante: Las instancias se extrajeron aleatoriamente de una base de datos de 7 imágenes exteriores.

Las imágenes se segmentaron manualmente para crear uno de clasificación para cada píxel. Cada instancia es una región 3x3.

1. Número de instancias:
   1. Datos de entrenamiento: 210
   2. Datos de prueba: 2100
2. Número de atributos: 19 atributos continuos
3. Valores de atributos que faltan: ninguna
4. Distribución de clases:
   1. Clases: ladrillo, cielo, follaje, cemento, ventana, camino, hierba.
   2. 30 instancias por clase para datos de formación.
   3. 300 instancias por clase para datos de prueba.

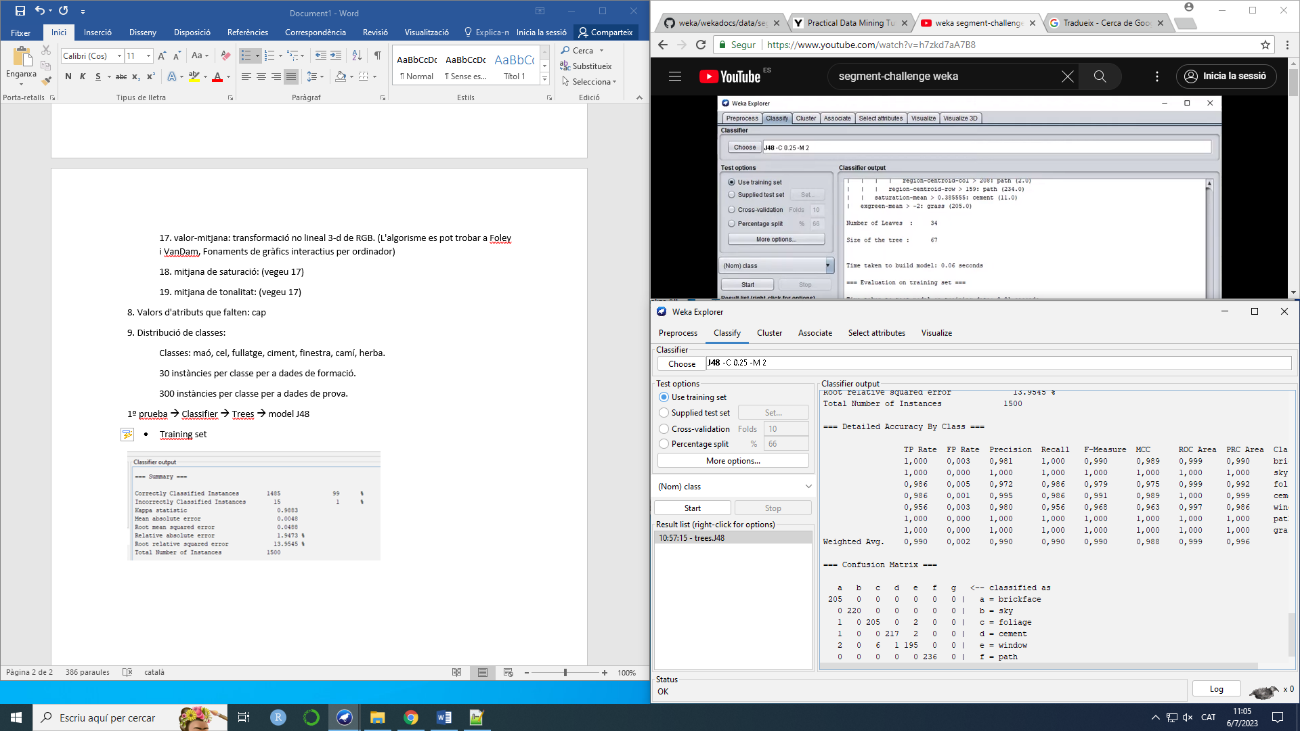
1º prueba 🡪 Classifier 🡪 Trees 🡪 model J48

* Training set
* Número de capes(hojas): 34
* Tamaño del árbol: 67



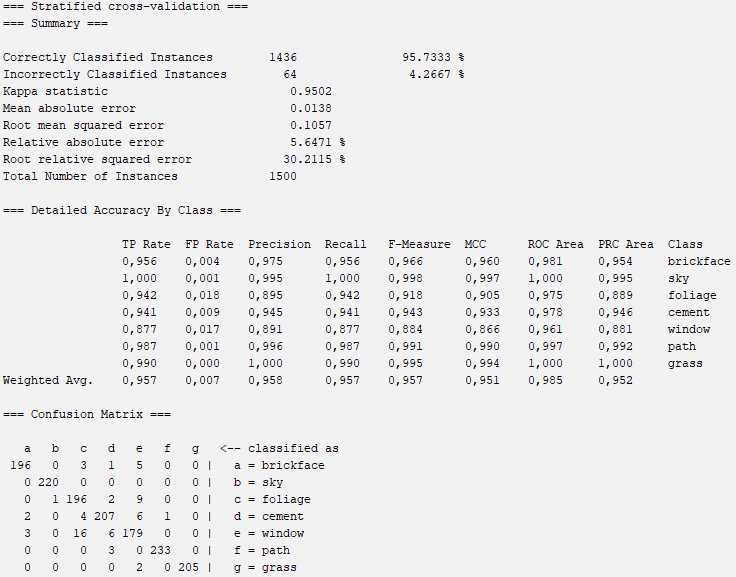
Parece que el modelo nos muestra que ha clasificado de manera correcta un total de 1485 instancias con un 99% en la fase de training set y de manera incorrecta ha clasificado 15 instancias representado el 1%.

Aunque los datos son bastante buenos, quizás es demasiado bueno, sospecho que esto es debido a que no están balanceadas las muestras.



De manera más visual, weka nos indica cuál es la matriz de correlación que ha obtenido y nos desengrana las instancias clasificadas de manera casi perfecta.

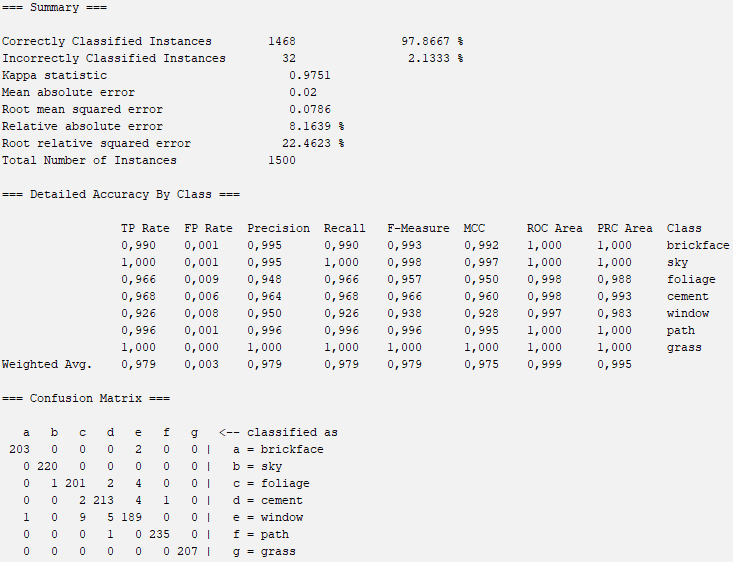
Se observa que aquellas instancias que no se han clasificado de manera correcta se encuentran en la columna “a” y con un total de 4 instancias mal clasificadas, en la columna “c” con un total de 6 instancias erróneas, en la columna “d” con un total de 1 instancia mal clasificada y en la columna “e” un total de 4

 2º prueba 🡪 Classifier 🡪 Trees 🡪 model J48

* Cross-validation 🡪 Folds: 10
* Número de capes(hojas): 34
* Tamaño del árbol: 67

En esta ocasión la validación cruzada 10 veces proporciona una precisión media del clasificador y en esta ocasión los resultados parecen que han empobrecido, con una tasa de clasificación correcta de un 96% aproximadamente frente a un 4.3% mal clasificadas.

3a prueba 🡪 Classifier 🡪 Trees 🡪 Randomforest

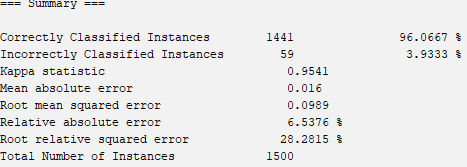
* Cross-validation 🡪 Folds: 10

Los resultados obtenidos indican que el modelo random forest es capaz con un 98% aproximadamente de clasificar bien 1468 instancias frente al 2% que clasifica mal (32 instancias)

Comparado con el modelo j48 en modo cross-validation, éste modelo, es mucho más preciso a la hora de hacer la clasificación. Vamos mejorando

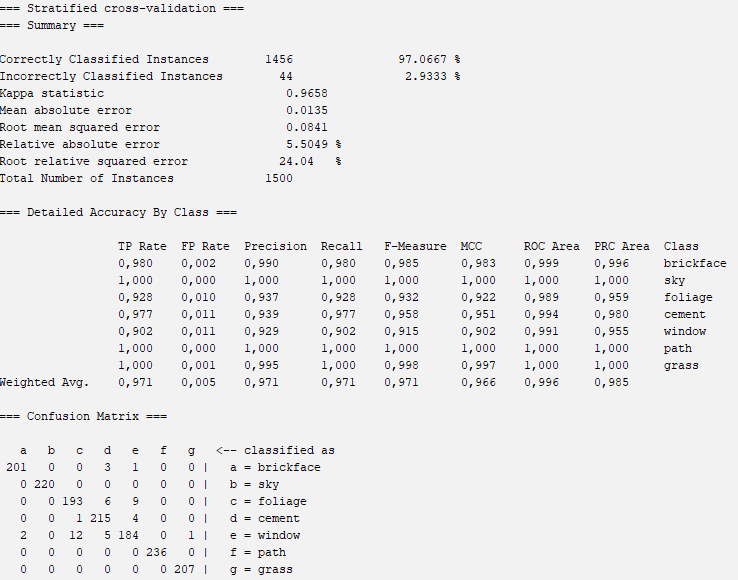
4º prueba 🡪 Classifier 🡪 functions 🡪 Logistic

* Cross-validation 🡪 Folds: 10



El modelo logístico revela un porcentaje de resultados similar a la que expresa el modelo de árbol j48, ambos utilizando la misma opción de cross-validation, por lo que descarto la función logística.

## 5º prueba 🡪 Classifier 🡪 functions 🡪 multilayer Perceptron ##

* Cross-validation 🡪 Folds: 10

he observado que el modelo multilayer Perceptron devuelve unos porcentajes de clasificación bastante buenos, en cambio, si nos fijamos en la matriz de correlación las instancias mal clasificadas son más abundantes en columnas en concreto, lo cual no es un buen modelo para llevar a cabo el análisis de los datos, ya que por ejemplo los otros modelos que he visto anteriormente, a pesar de contar con un mayor grado de porcentaje de error, comparado con el MP, la cantidad de error acumulados por columna es menor.

Conclusión: El modelo que mejor se ajusta a la hora de clasificar las instancias correctamente, parece que es el Random Forest con la opción de cross-validation. Eso es possible ya que las muestras no están balanceadas. Gracias a que el modo de cross-validation añade aleatoriedad en la comparación de las muestras permite hacer una predicción más ajustada.