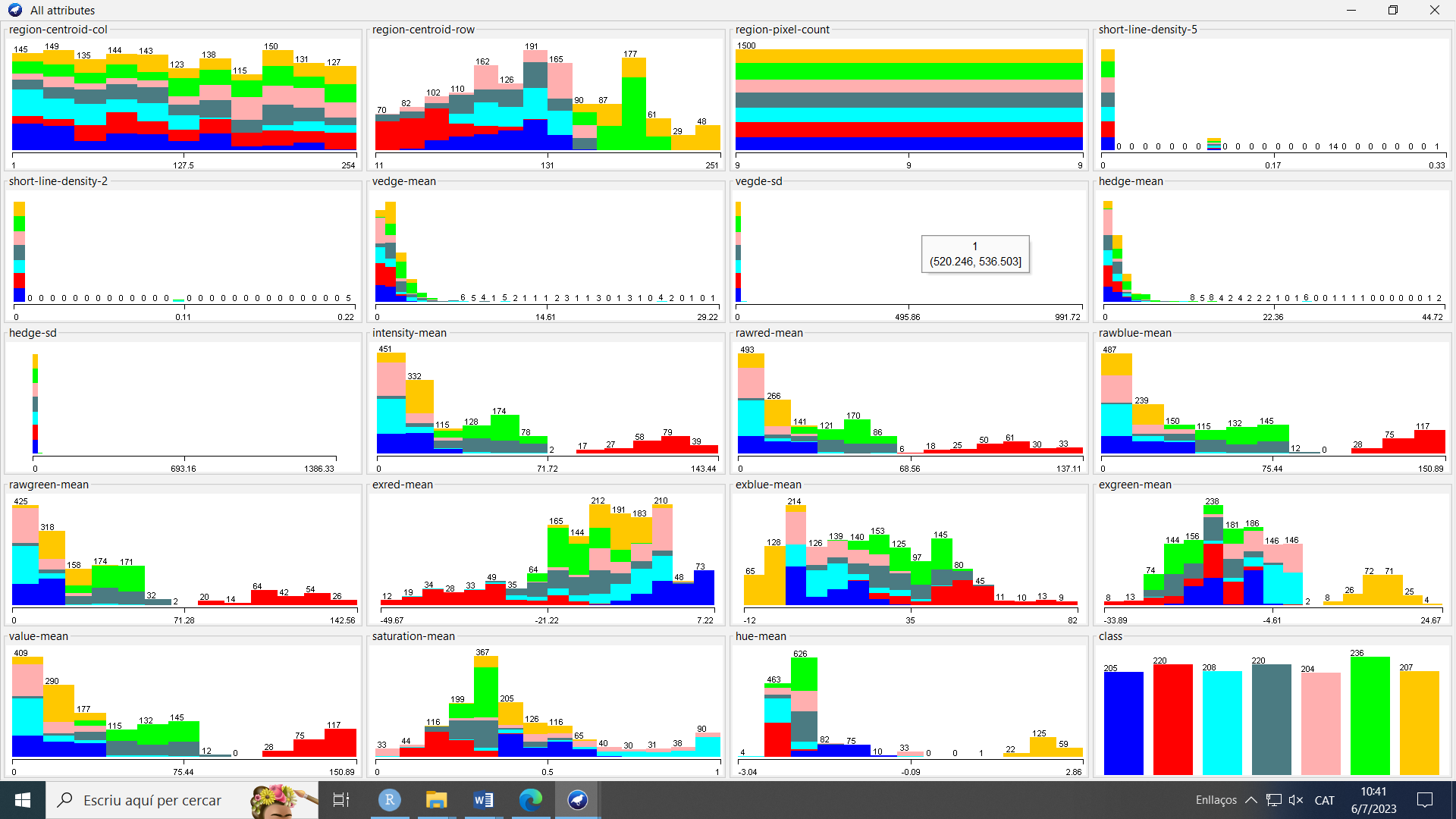
Accedim al fitxer ‘segment-challenge’ en format nadiu de Weka, ARFF per realitzar la pràctica.

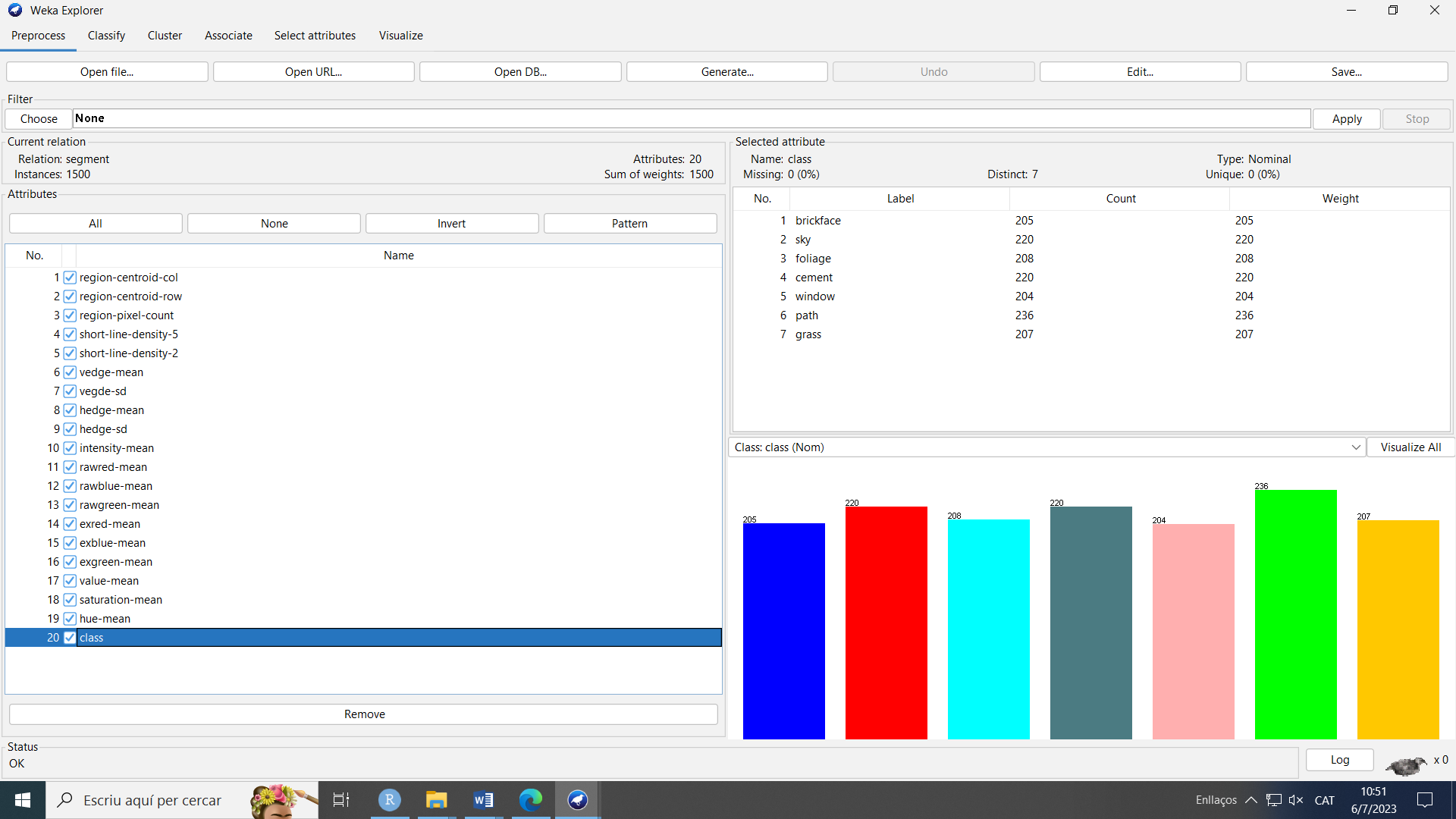
En aquest fitxer s’observa un conjunt de 20 atributs i 1500 files o registres:

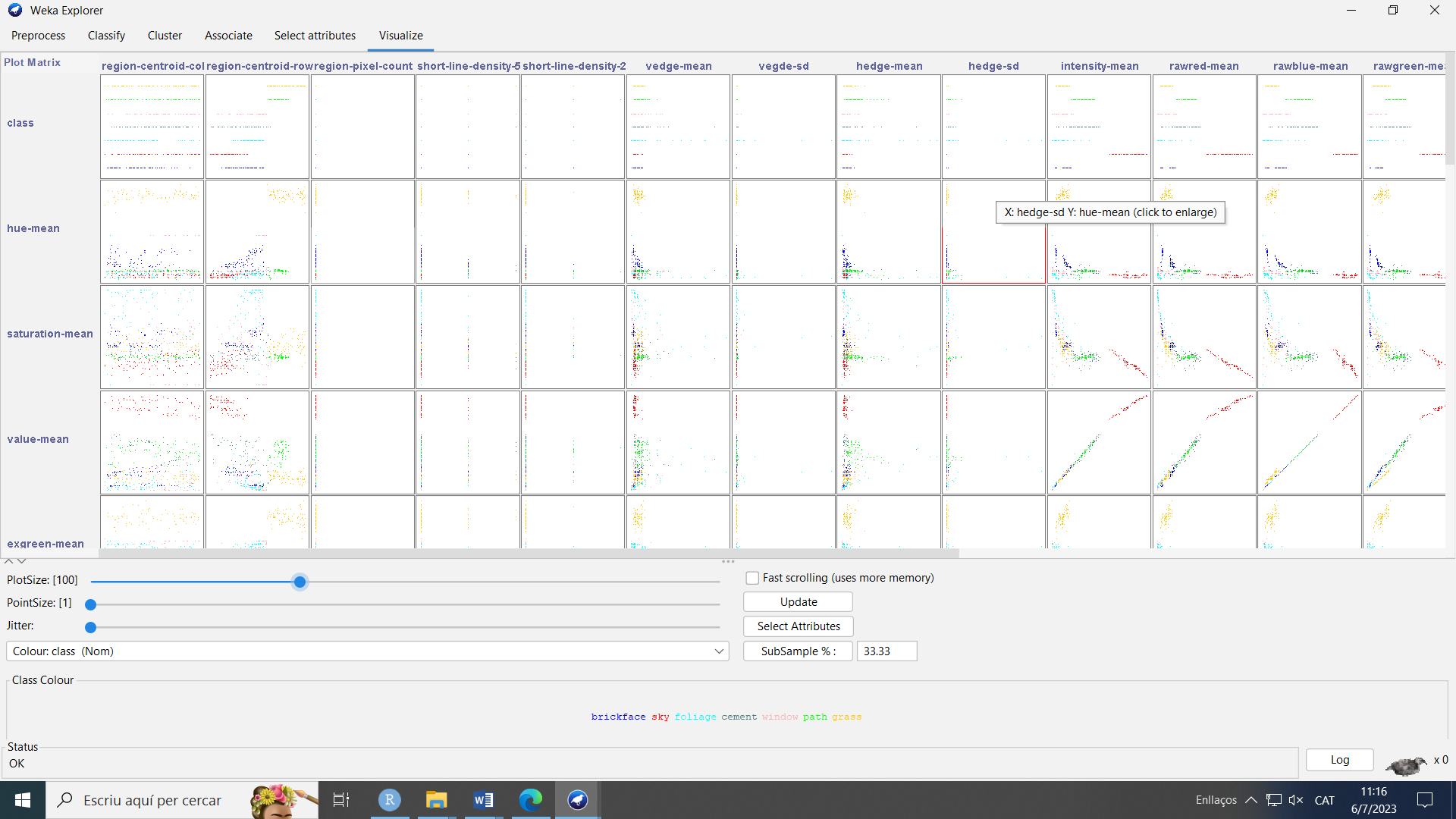
1. region-centroid-col
2. region-centroid-row
3. region-pixel-count
4. short-line-density-5
5. short-line-density-2
6. vedge-mean
7. vedge-sd
8. hedge-mean
9. hedge-sd
10. intensity-mean
11. rawred-mean
12. rawblue-mean
13. rawgreen-mean
14. exred-mean
15. exred-blue mean
16. exred-green mean
17. value-mean
18. saturation-mean
19. hue-mean
20. class

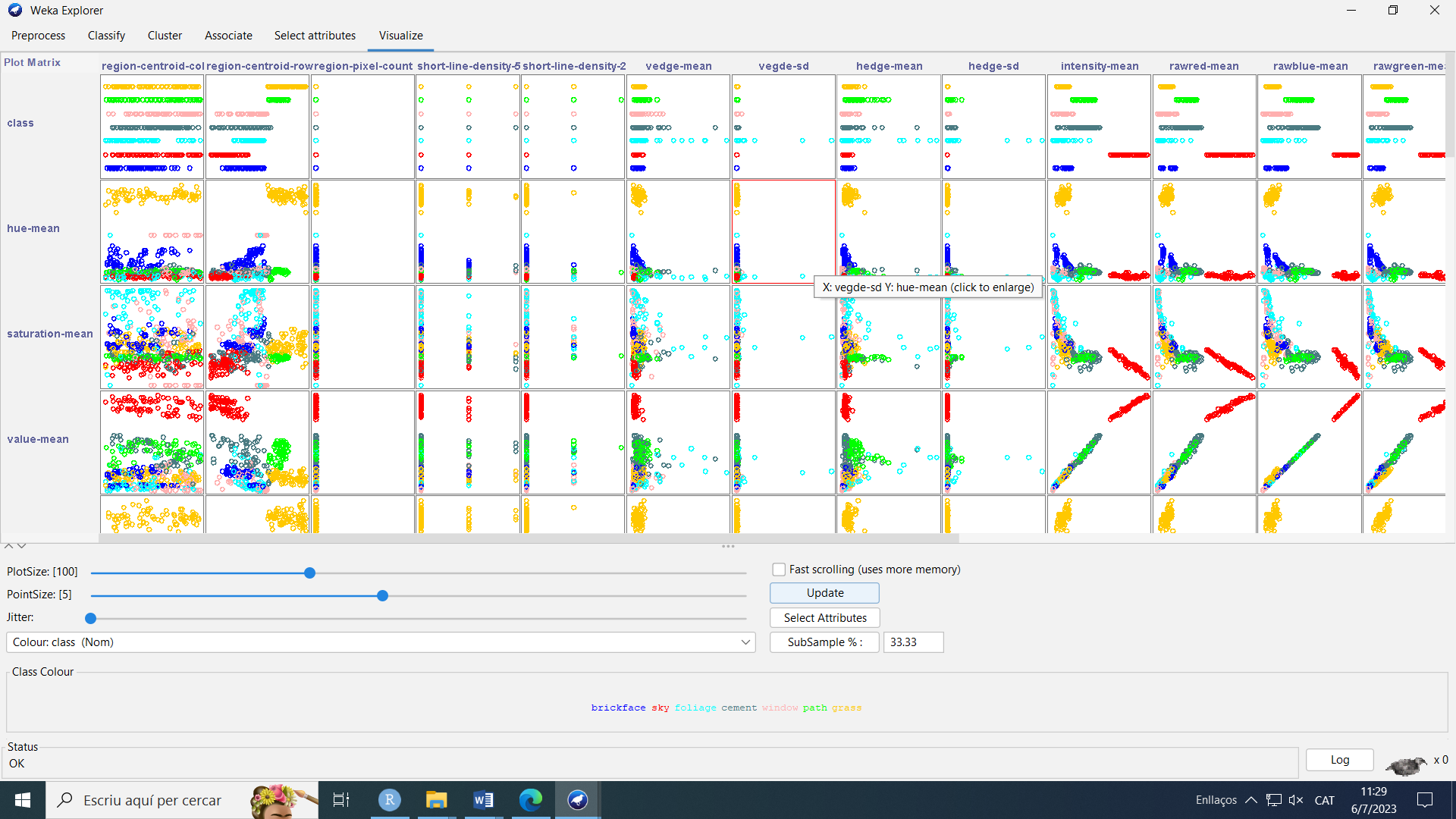
Podem visualitzar aquests atributs en format de gràfics:



Tots els atributs són de tipus numèric excepte l’atribut class (nominal).







Escollim 3 models dins de l’eina WEKA. Per això, accedim a la pestanya Classify:

1. **model LinearRegression amb Cross-validation (15). Escollim l’atribut hue-mean.**

hue-mean =

0.0006 \* region-centroid-col +

0.0022 \* region-centroid-row +

-0.4996 \* short-line-density-5 +

-0.0147 \* vedge-mean +

0.0003 \* vegde-sd +

-0.0188 \* hedge-mean +

0.0003 \* hedge-sd +

0.0141 \* rawred-mean +

-0.0074 \* exred-mean +

-0.0196 \* value-mean +

-1.2252 \* saturation-mean +

-0.0676 \* class=foliage,path,cement,window,brickface,grass +

-0.2478 \* class=path,cement,window,brickface,grass +

0.1779 \* class=cement,window,brickface,grass +

0.1031 \* class=window,brickface,grass +

0.5263 \* class=brickface,grass +

3.2816 \* class=grass +

-1.3075

=== Summary ===

Correlation coefficient 0.984

Mean absolute error 0.1892

Root mean squared error 0.2728

Relative absolute error 16.9044 %

Root relative squared error 17.8172 %

Total Number of Instances 1500

1. **model Random Forest amb Percentage Split (66%). Escollim l’atribut hue-mean.**

Scheme: weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Relation: segment

Instances: 1500

Attributes: 20

region-centroid-col

region-centroid-row

region-pixel-count

short-line-density-5

short-line-density-2

vedge-mean

vegde-sd

hedge-mean

hedge-sd

intensity-mean

rawred-mean

rawblue-mean

rawgreen-mean

exred-mean

exblue-mean

exgreen-mean

value-mean

saturation-mean

hue-mean

class

Test mode: split 66.0% train, remainder test

=== Summary ===

Correlation coefficient 0.9974

Mean absolute error 0.0501

Root mean squared error 0.1127

Relative absolute error 4.4039 %

Root relative squared error 7.2865 %

Total Number of Instances 510

1. **model MultilayerPerceptron amb Cross-validation (10). Escollim l’atribut hue-mean.**

Scheme: weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Relation: segment

Instances: 1500

Attributes: 20

region-centroid-col

region-centroid-row

region-pixel-count

short-line-density-5

short-line-density-2

vedge-mean

vegde-sd

hedge-mean

hedge-sd

intensity-mean

rawred-mean

rawblue-mean

rawgreen-mean

exred-mean

exblue-mean

exgreen-mean

value-mean

saturation-mean

hue-mean

class

Test mode: 10-fold cross-validation

**Degut a que hi ha molts nodes (sigmoid) només adjjuntem el node 0 (primer) i el node 13 (últim).**

Linear Node 0

Inputs Weights

Threshold 0.8927153419921684

Node 1 -1.5261847033234963

Node 2 -0.5162860532713383

Node 3 0.0014425529008760662

Node 4 -0.2294167355677497

Node 5 1.7555201468345962

Node 6 0.520849272983069

Node 7 -0.45653244318939706

Node 8 0.24172344409503368

Node 9 0.6461052890644875

Node 10 -1.2757975448429808

Node 11 -0.6609971826097701

Node 12 0.4687573508642638

Node 13 -0.6990219594321967

Sigmoid Node 13

Inputs Weights

Threshold -0.4223512735131205

Attrib region-centroid-col -0.14669887729730138

Attrib region-centroid-row 0.2681553186881885

Attrib region-pixel-count -0.047296182811103774

Attrib short-line-density-5 0.10354372981308593

Attrib short-line-density-2 0.24808284602980218

Attrib vedge-mean 0.022766178398779834

Attrib vegde-sd 0.03247337929543042

Attrib hedge-mean 0.23508481288076524

Attrib hedge-sd 0.32944630137104286

Attrib intensity-mean 0.35393290029757674

Attrib rawred-mean 0.3555631711474571

Attrib rawblue-mean 0.20327865520291757

Attrib rawgreen-mean 0.4550096418523835

Attrib exred-mean -0.492148966449577

Attrib exblue-mean -0.19486613091378513

Attrib exgreen-mean 0.9149835963872106

Attrib value-mean 0.14957616476676444

Attrib saturation-mean -0.6641035552287147

Attrib class=brickface 0.7478966879925226

Attrib class=sky 0.047743988324158104

Attrib class=foliage 0.26330724163865227

Attrib class=cement -0.09514716459226591

Attrib class=window 1.0277363488587201

Attrib class=path 0.17973523325689938

Attrib class=grass -0.1957476081085655

Class

Input

Node 0

=== Summary ===

Correlation coefficient 0.9988

Mean absolute error 0.0446

Root mean squared error 0.0751

Relative absolute error 3.9892 %

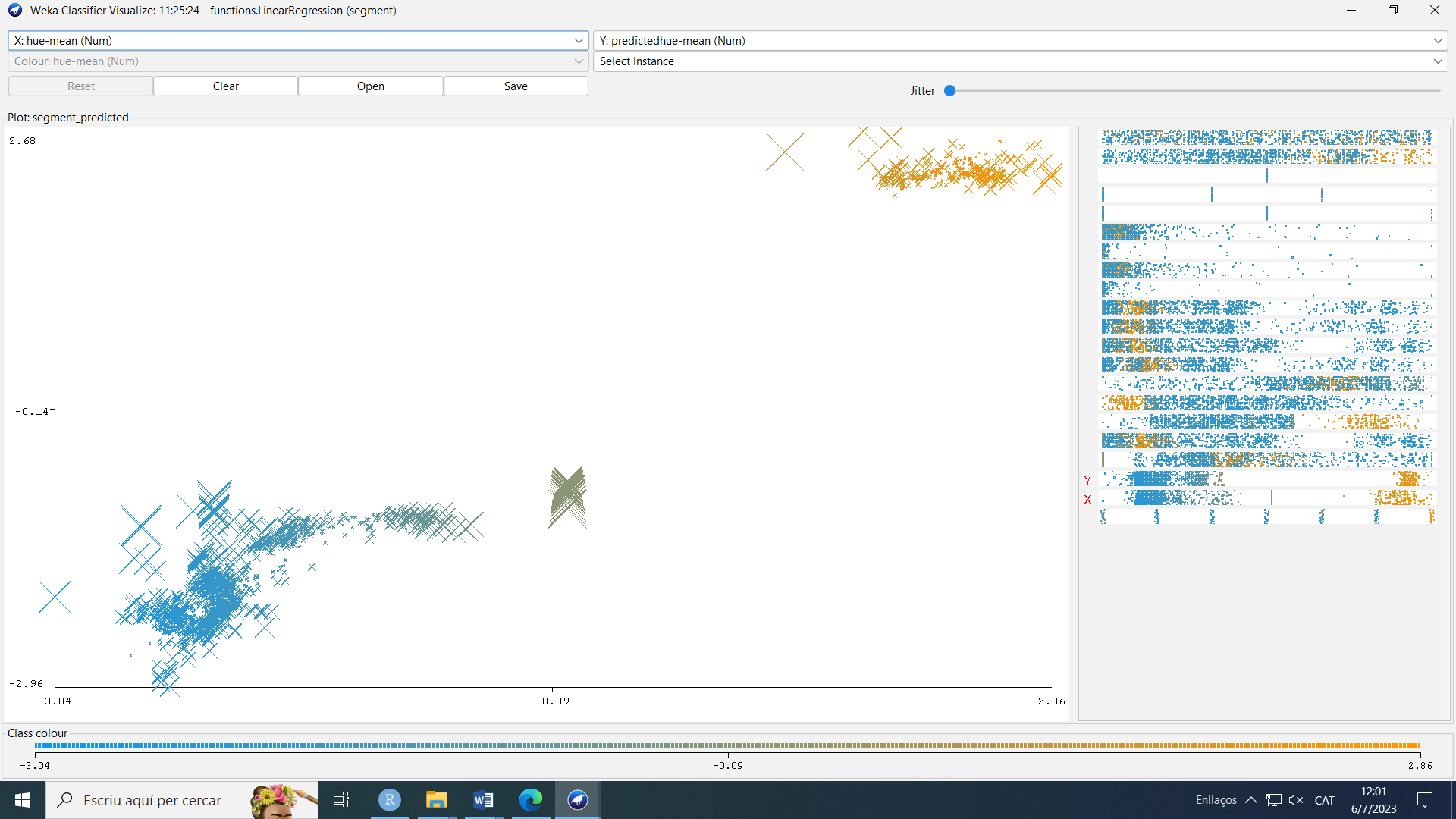
Root relative squared error 4.9047 %

Total Number of Instances 1500

Caracteristiques dels 3 models:

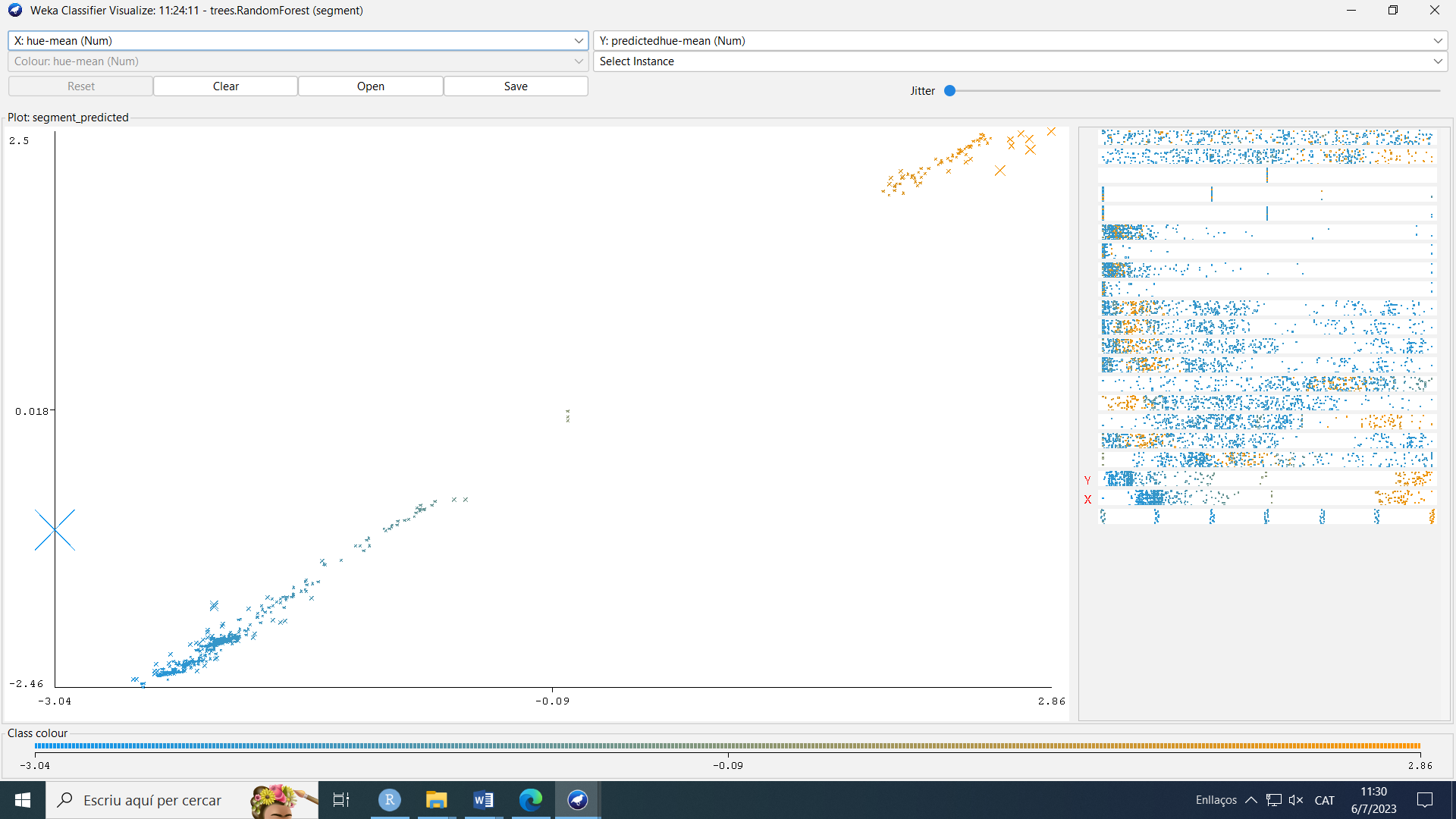
1. El model **LinearRegression** s'utilitza per predir el valor d'una variable en funció del valor d'una altra variable. La variable que es vol predir s'anomena variable dependent. La variable que utilitzeu per predir el valor de l'altra variable s'anomena variable independent. En **LinearRegression**, les relacions es modelen mitjançant funcions de predicció lineal els paràmetres del model desconeguts s'estimen a partir de les dades. Aquests models s'anomenen models lineals. Amb més freqüència, s'assumeix que la mitjana condicional de la resposta donats els valors de les variables explicatives (o predictors) és una funció afí d'aquests valors; amb menys freqüència, s'utilitza la mediana condicional o algun altre quantil.

En aquesta gràfica, es visualitza 3 concentracions de valors: en la parte inferior de manera molt disseminada, en la parte inferior molt concentrada en unes coordinades (vertical -0.5 i horitzontal -0.09) i a la part superior (vertical 2.68).



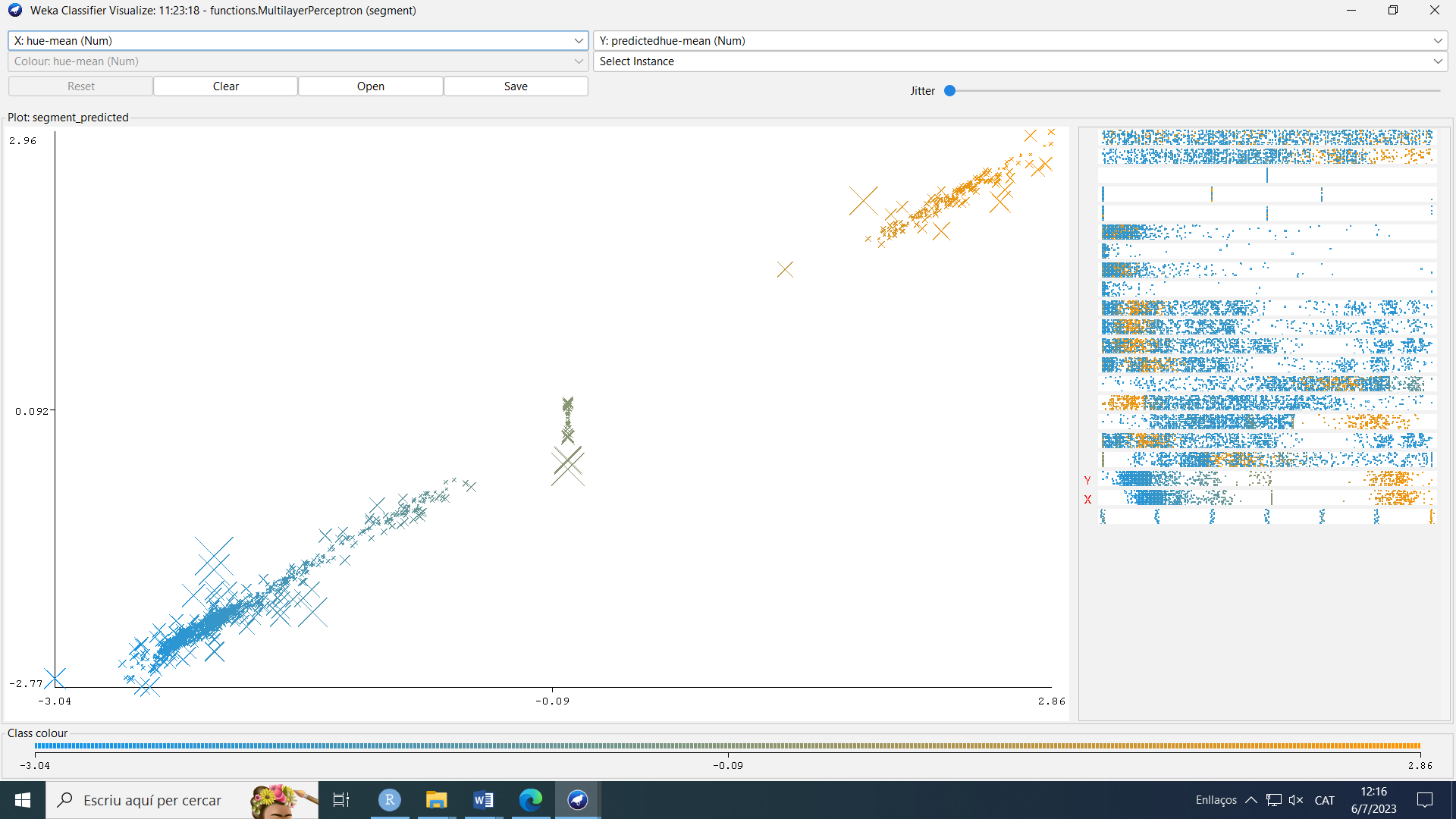
1. **RandomForest** és un algorisme d'aprenentatge automàtic d'ús comú, que combina la sortida de múltiples arbres de decisió per arribar a un únic resultat. La seva facilitat d'ús i flexibilitat han impulsat la seva adopció, ja que gestiona problemes tant de classificació com de regressió. Per a les tasques de classificació, la sortida de **RandomForest** és la classe seleccionada per la majoria dels arbres. Per a les tasques de regressió, es retorna la predicció mitjana o mitjana dels arbres individuals. **RandomForest** s'utilitza sovint com a models de caixa negra a les empreses, ja que generen prediccions raonables en una àmplia gamma de dades i requereixen poca configuració.

La gràfica del model **RandomForest** mostra 2 concentracions separades de valors en forma de línia en diagonal (a la part inferior i a la part superior). La concentració dels valors de la part inferior es va difuminant a mesura que aquests valors s’apropen a 0 en la vertical i en l’horitzontal. Afegir l’existència d’un petit nucli de valors al centre de la gràfica.



1. **MultilayerPerceptron** és una classe de xarxa neuronal artificial (ANN) completament connectada. El terme MLP (Multi Layer Perceptron) s'utilitza de manera ambigua, de vegades vagament per significar qualsevol ANN anticipada, de vegades estrictament per referir-se a xarxes compostes de múltiples capes de perceptrons (amb activació de llindar). Els perceptrons multicapa de vegades s'anomenen col·loquialment xarxes neuronals "vainilla", especialment quan tenen una sola capa oculta.

La gràfica del model **MultilayerPerceptron** s’assembla molt a l’anterior gràfica del model **RandomForest**. Destacar una major intensitat dels valors a les 3 concentracions de dades (sobretot, a les parts inferior i superior).



Comparativa de models

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MODELS | ***CROSS-VALIDATION*** | ***Correlation coefficient*** | ***Mean absolute error*** | ***Root mean squared error*** | ***Relative absolute error*** | ***Root relative squared error*** |
| **LinearRegression** | SI | 0.984 | 0.1892 | 0.2728 | 16.9044 % | 17.8172 % |
| **RandomForest** | NO | 0.9974 | 0.0501 | 0.1127 | 4.4039 % | 7.2865 % |
| **MultilayerPerceptron** | SI | 0.9988 | 0.0446 | 0.0751 | 3.9892 % | 4.9047 % |

Temps d’entrega de resultats

**LinearRegression=>** Time taken to build model: 0.01 seconds

**RandomForest =>** Time taken to build model: 0.18 seconds

**MultilayerPerceptron =>** Time taken to build model: 2.74 seconds

Analitzant la taula comparativa de models (**LinearRegression**, **RandomForest** i **MultilayerPerceptron**) es demostra com el model **MultilayerPerceptron** té millors resultats i percentatges de tots 3 models. A continuació, es troba el model **RandomForest** i els pitjors valors obtinguts són del model **LinearRegression**. El coeficient de correlació del model **MultilayerPerceptron** és gairebé 0.999 i aquest model té uns percentatges de *Relative absolute error* i *Root relative squared error* força baixos. En la meva opinió, el **MultilayerPerceptron** seria el millor model de tots 3 per aquest data set ‘segment-challenge’. Els altres models donen uns valors de *Mean absolute error* i *Root mean squared error* superiors a 0.04 i 0.07 respectivament.

Visualitzant el temps d’entrega dels models, el **MultilayerPerceptron** té un temps molt superior (superant els 2.5 segons). Els altres 2 models (**LinearRegression** i **RandomForest**) construeixen el model de manera immediata i instantània. Encara que el temps d’entrega és molt important i en aquesta ocasió, penalitza el model **MultilayerPerceptron** esculliria aquest model per davant dels altres dos models: **LinearRegression** i **RandomForest**.

El millor model per aplicar seria el model **MultilayerPerceptron**.