

Nama : Ferza Reyaldi
NIM : 09021281924060
Kelas : TI REGULER
Mata Kuliah : Data Mining

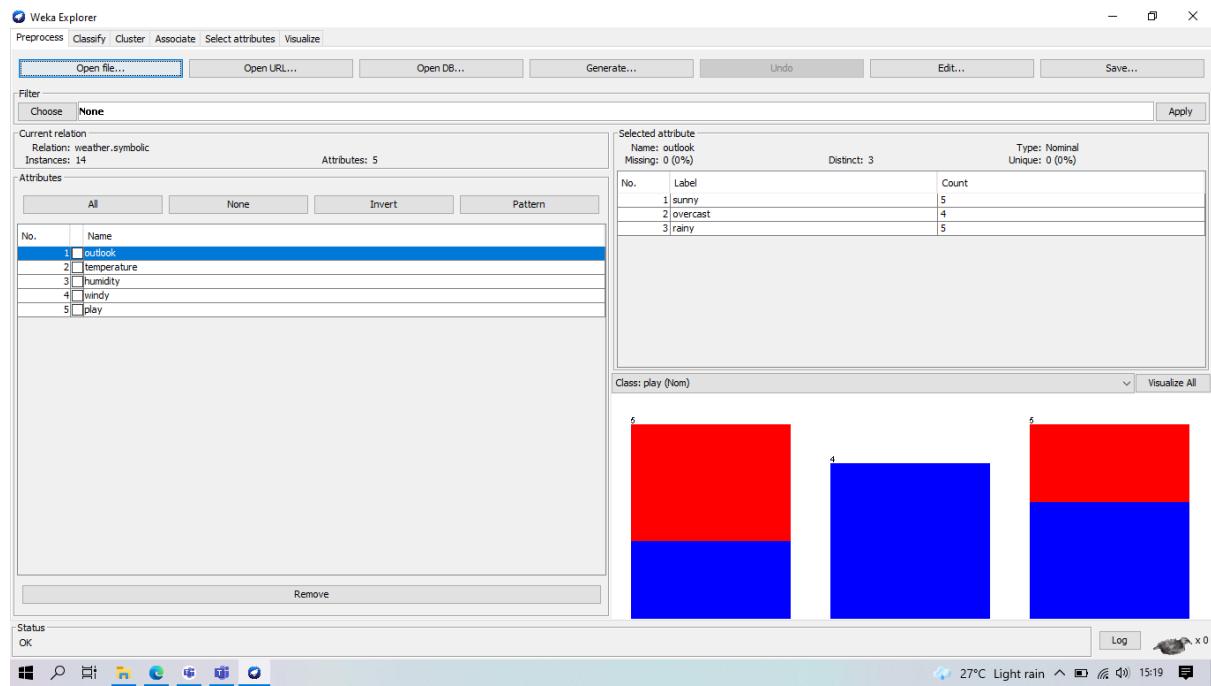
Data Mining with Weka (3.1-5.4)

3.1 Simplicity

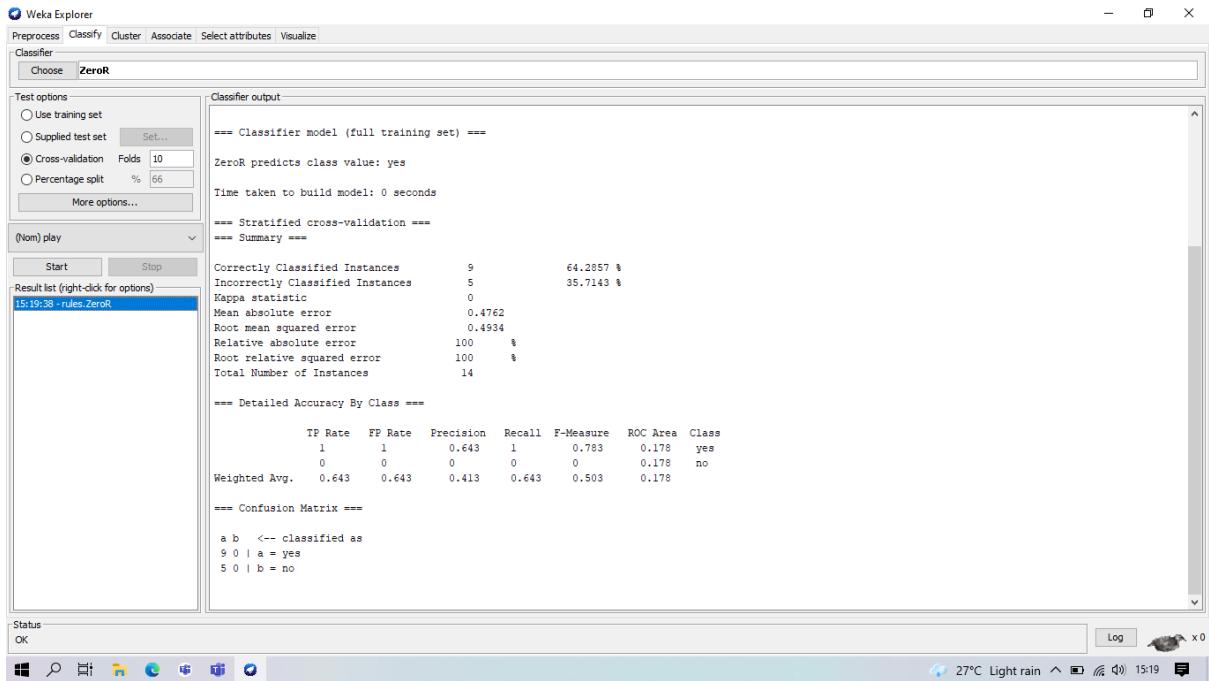
Dalam mencoba suatu algoritma dalam Data Mining diharuskan mencoba dari sesuatu yang paling sederhana terlebih dulu karena mungkin dengan metode paling sederhana telah menunjukkan hasil yang baik. Tidak ada algoritma yang terbaik secara umum, keberhasilan suatu algoritma ditentukan juga dengan domain dimana algoritma tersebut diterapkan.

Salah satu struktur model yang sederhana adalah satu atribut mengerjakan semua pekerjaan, contohnya adalah OneR.

Load dataset weather.



Gunakan ZeroR untuk menentukan baseline accuracnya. Baseline accuracnya adalah 64,3%.



The screenshot shows the Weka Explorer interface with the 'Classifier' tab selected and 'ZeroR' chosen. The 'Test options' panel shows 'Cross-validation' with 'Folds' set to 10. The 'Classifier output' panel displays the following text:

```

==== Classifier model (full training set) ====
ZeroR predicts class value: yes
Time taken to build model: 0 seconds
==== Stratified cross-validation ====
==== Summary ====
Correctly Classified Instances      9      64.2857 %
Incorrectly Classified Instances   5      35.7143 %
Kappa statistic                   0
Mean absolute error               0.4762
Root mean squared error           0.4934
Relative absolute error            100 %
Root relative squared error       100 %
Total Number of Instances         14

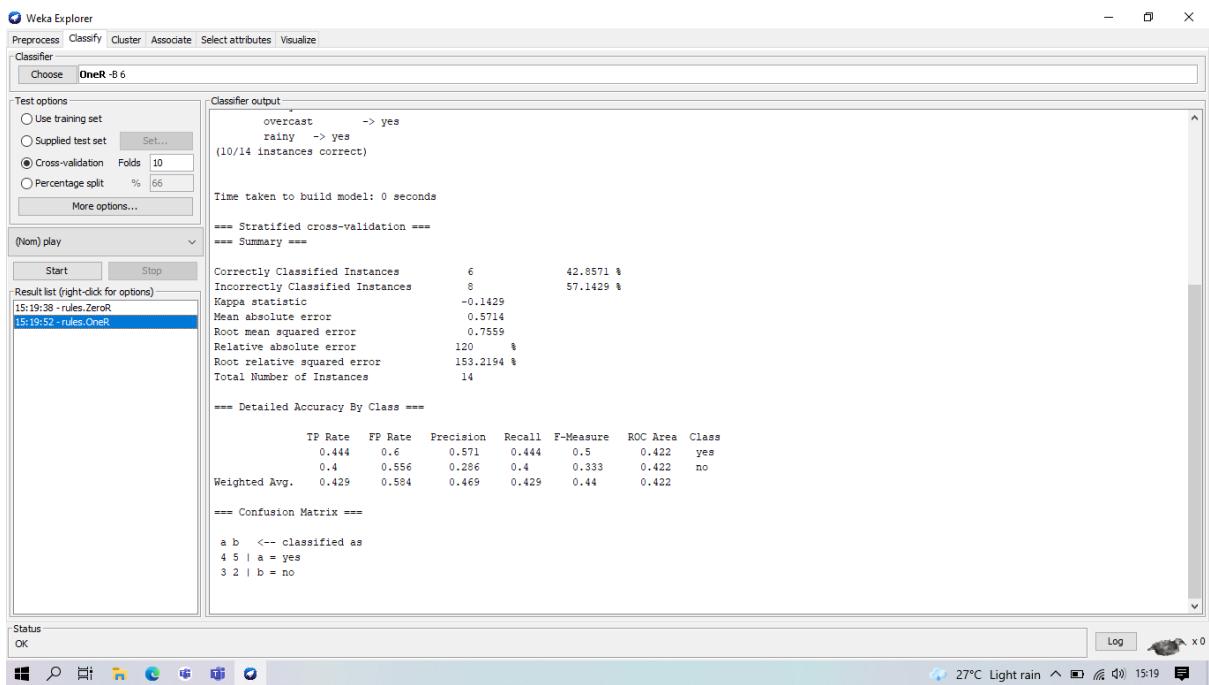
==== Detailed Accuracy By Class ====
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      1          1        0.643      1        0.783      0.178     yes
      0          0        0          0        0          0.178     no
Weighted Avg.   0.643    0.643    0.413    0.643    0.503    0.178

==== Confusion Matrix ====
a b  <- classified as
9 0 | a = yes
5 0 | b = no

```

The status bar at the bottom indicates 'OK'.

Berdasarkan prinsip simplicity, gunakan algoritma yang sederhana terlebih dulu, contohnya OneR. untuk akurasi prediksinya hanya 42,8% (lebih kecil dari baseline accuracy), sehingga bisa disimpulkan bahwa dataset tidak cocok dengan classifier OneR.



The screenshot shows the Weka Explorer interface with the 'Classifier' tab selected and 'OneR-B6' chosen. The 'Test options' panel shows 'Cross-validation' with 'Folds' set to 10. The 'Classifier output' panel displays the following text:

```

overcast      -> yes
rainy         -> yes
(10/14 instances correct)

Time taken to build model: 0 seconds
==== Stratified cross-validation ====
==== Summary ====
Correctly Classified Instances      6      42.8571 %
Incorrectly Classified Instances   8      57.1429 %
Kappa statistic                   -0.11429
Mean absolute error               0.5714
Root mean squared error           0.7559
Relative absolute error            120 %
Root relative squared error       153.2194 %
Total Number of Instances         14

==== Detailed Accuracy By Class ====
      TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
      0.444    0.6        0.571    0.444    0.5        0.422     yes
      0.4      0.556    0.286    0.4      0.333    0.422     no
Weighted Avg.   0.429    0.584    0.469    0.429    0.44      0.422

==== Confusion Matrix ====
a b  <- classified as
4 5 | a = yes
3 2 | b = no

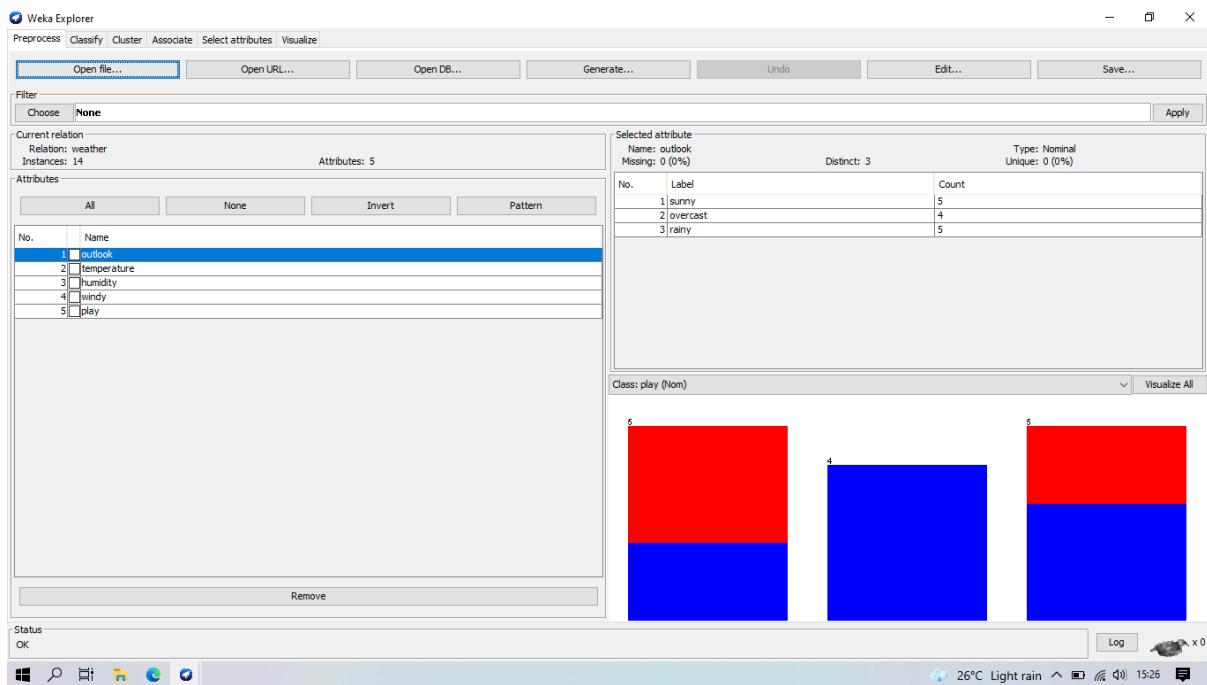
```

The status bar at the bottom indicates 'OK'.

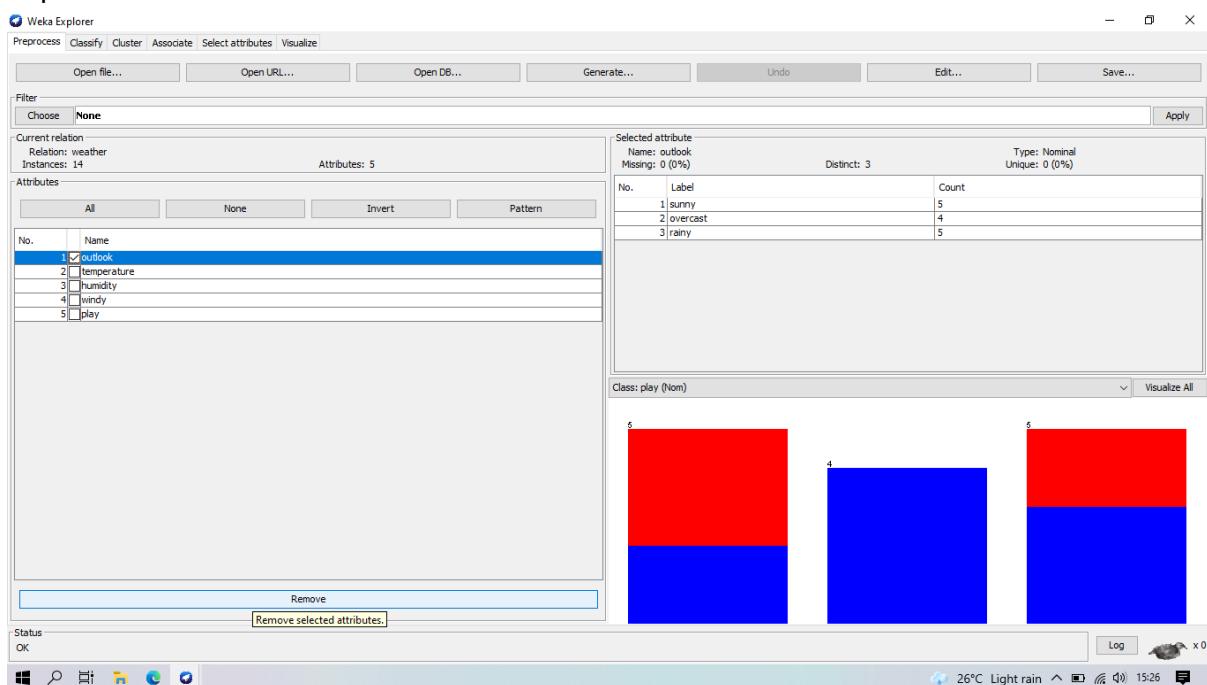
3.2 Overfitting

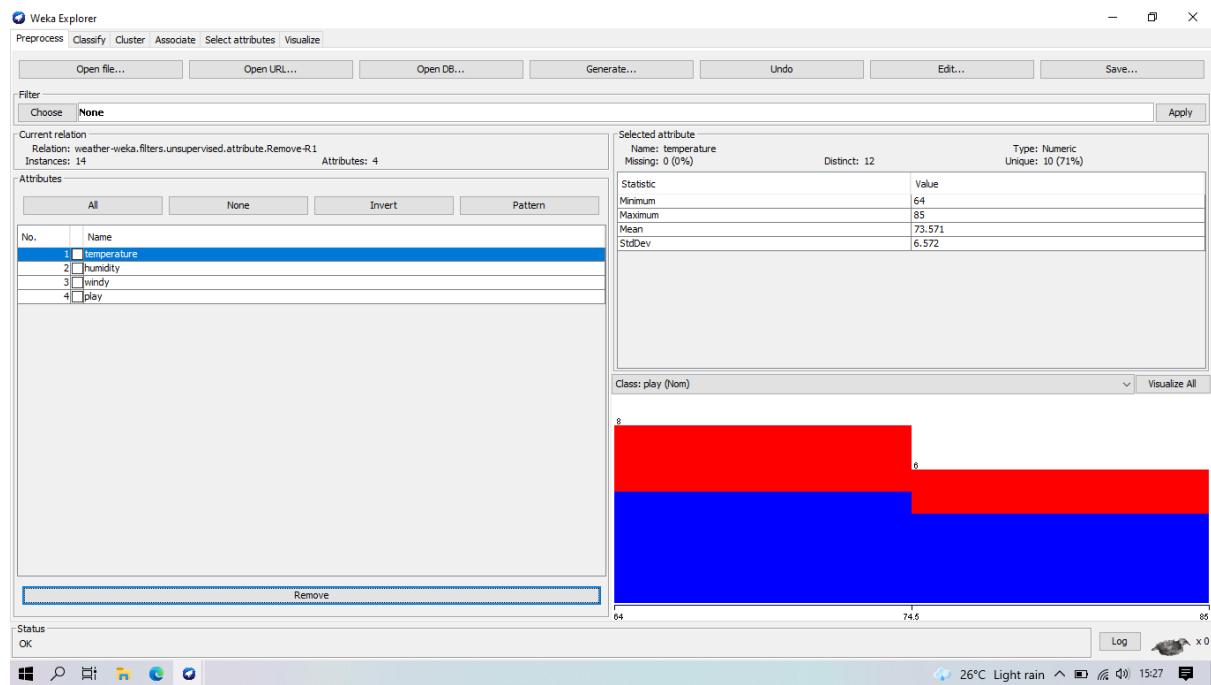
Overfitting sering terjadi karena model terlalu menyesuaikan dengan training data yang ada, sehingga ketika muncul data baru yang independen, hasil yang didapatkan sering kali tidak sesuai harapan.

Load dataset weather.

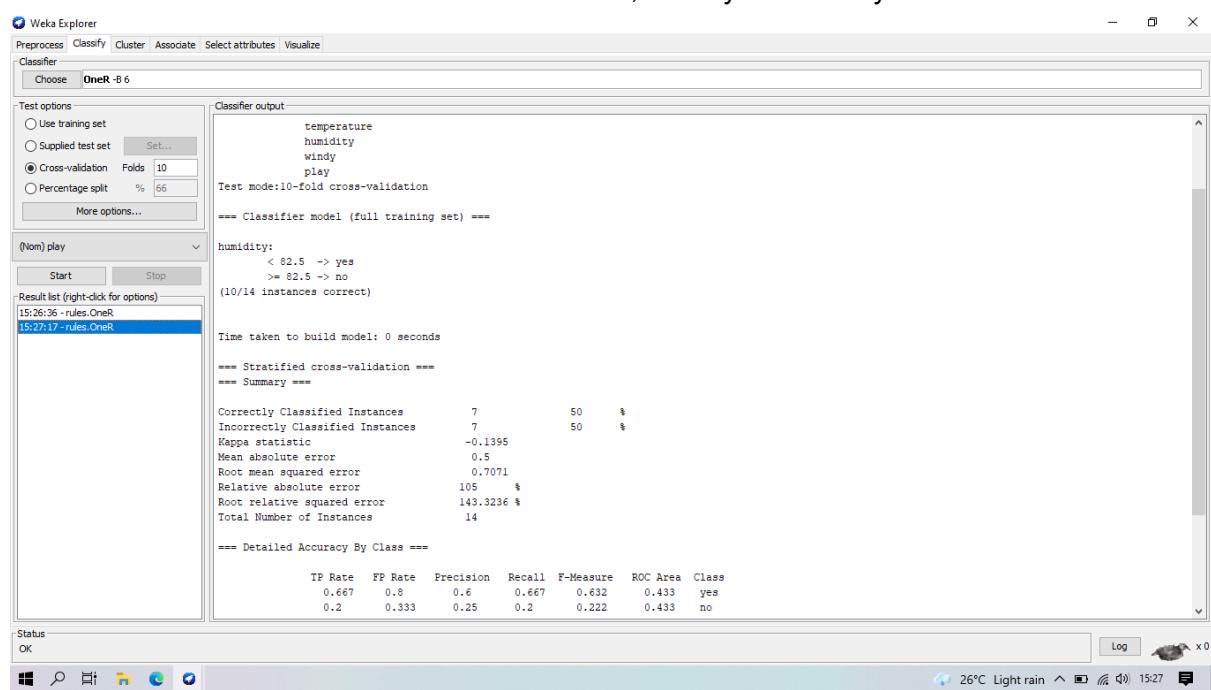


Hapus atribut outlook.

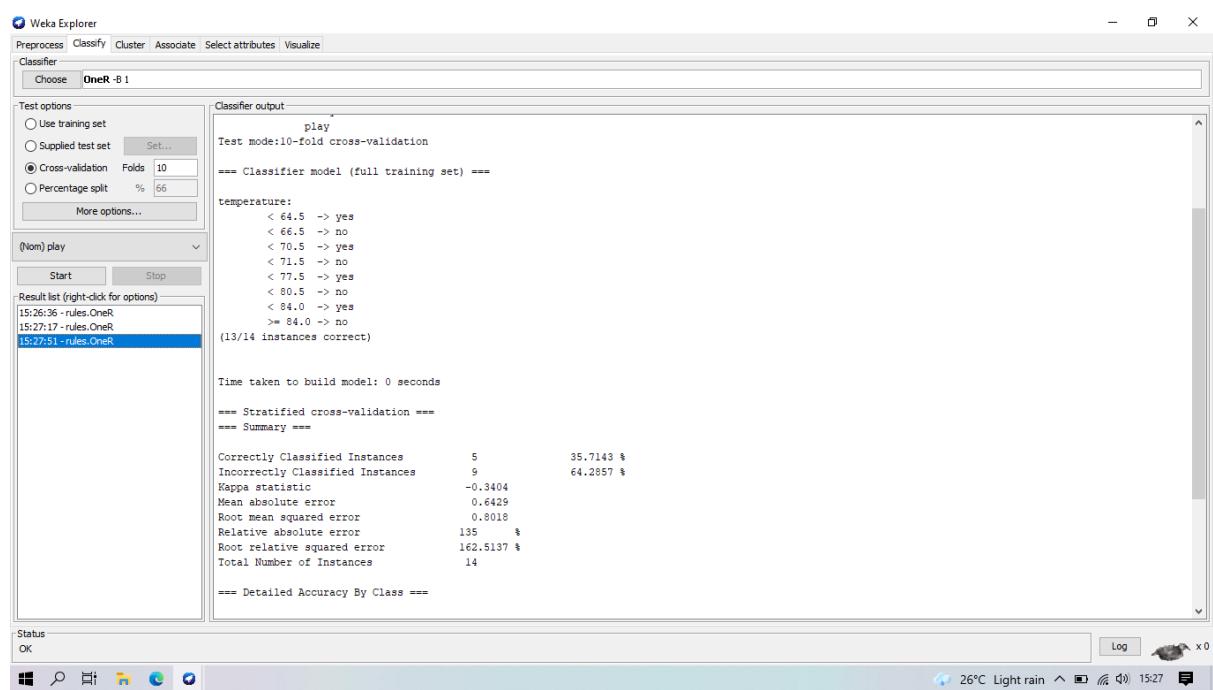
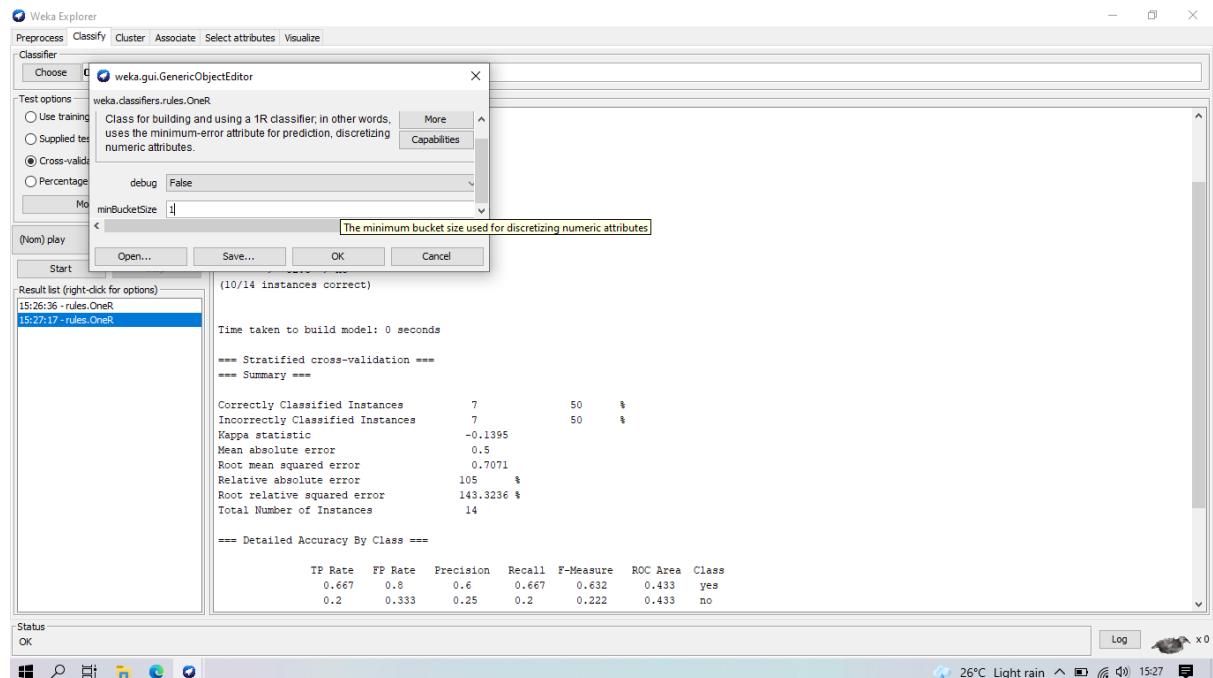




Gunakan classifier OneR untuk membuat model, hasilnya akurasinya 50%.



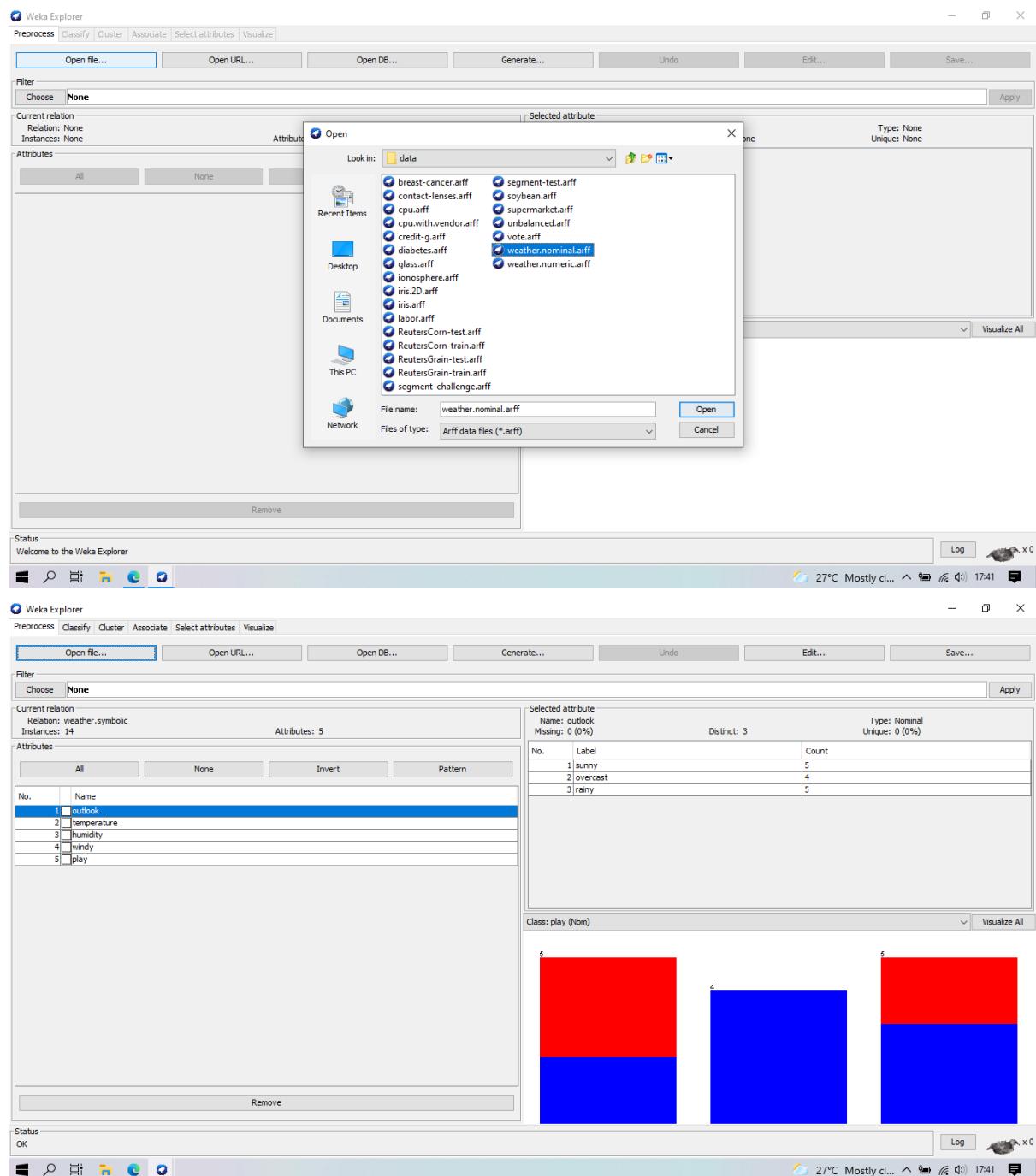
Klik nama classifier, kemudian terbuka jendela GenericObjectEditor. Ubah minBucketSize diubah menjadi 1. Ketika dijalankan kembali, akurasinya hanya bernilai 35,7%.



3.3 Using Probabilities

Salah satu bentuk model yang sederhana adalah menggunakan probabilitas, contohnya adalah algoritma naive bayes.

Load dataset weather nominal.



Pada tab classify, pilih classifier Naive Bayes. Akurasinya adalah 57%.

	sunny	overcast	rainy	[total]
temperature	3.0	5.0	4.0	12.0
humidity	3.0	5.0	4.0	12.0
windy	4.0	3.0	2.0	9.0
[total]	8	6	5	19

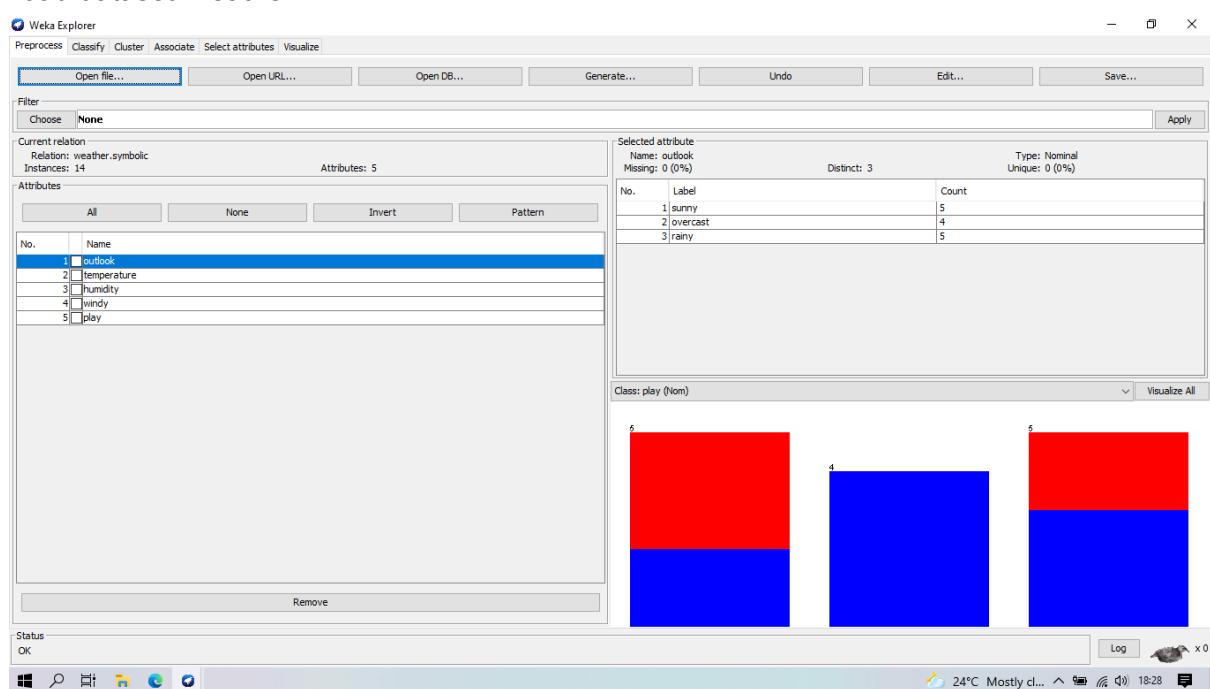
Time taken to build model: 0 seconds
 === Stratified cross-validation ===
 === Summary ===

	Correctly Classified Instances	Incorrectly Classified Instances	Kappa statistic	Mean absolute error	Root mean squared error
	8	6	-0.0244	0.4374	0.4916
	57.1429 %	42.8571 %			

3.4 Decision Trees

Decision Trees adalah satu-satu bentuk model sederhana, yaitu menggunakan beberapa atribut dalam melakukan pembuatan model.

Load dataset Weather.



Pilih Classifier J48, hasil akurasinya 50%.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose J48 -C 0.25 -M 2

Test options

Use training set

Supplied test set Set...

Cross-validation Folds 10

Percentage split % 66

More options...

(Nom) play

Start Stop

Result list (right-click for options)

18:31:09 - trees.J48

Classifier output

outlook
temperature
humidity
windy
play

Test mode: 10-fold cross-validation

==== Classifier model (full training set) ====
J48 pruned tree

outlook = sunny
| humidity = high: no (3.0)
| humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
| windy = TRUE: no (2.0)
| windy = FALSE: yes (3.0)

Number of Leaves : 5

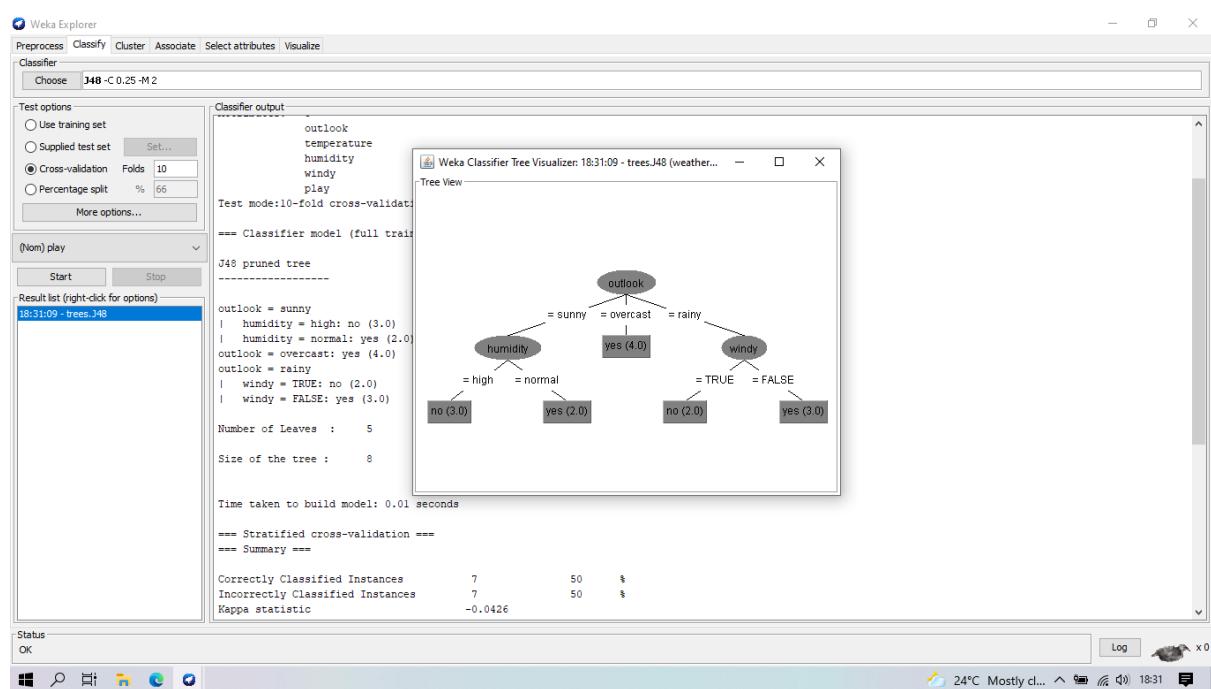
Size of the tree : 8

Time taken to build model: 0.01 seconds

==== Stratified cross-validation ====
==== Summary ====
Correctly Classified Instances 7 50 %
Incorrectly Classified Instances 7 50 %
Kappa statistic -0.0426

Status OK

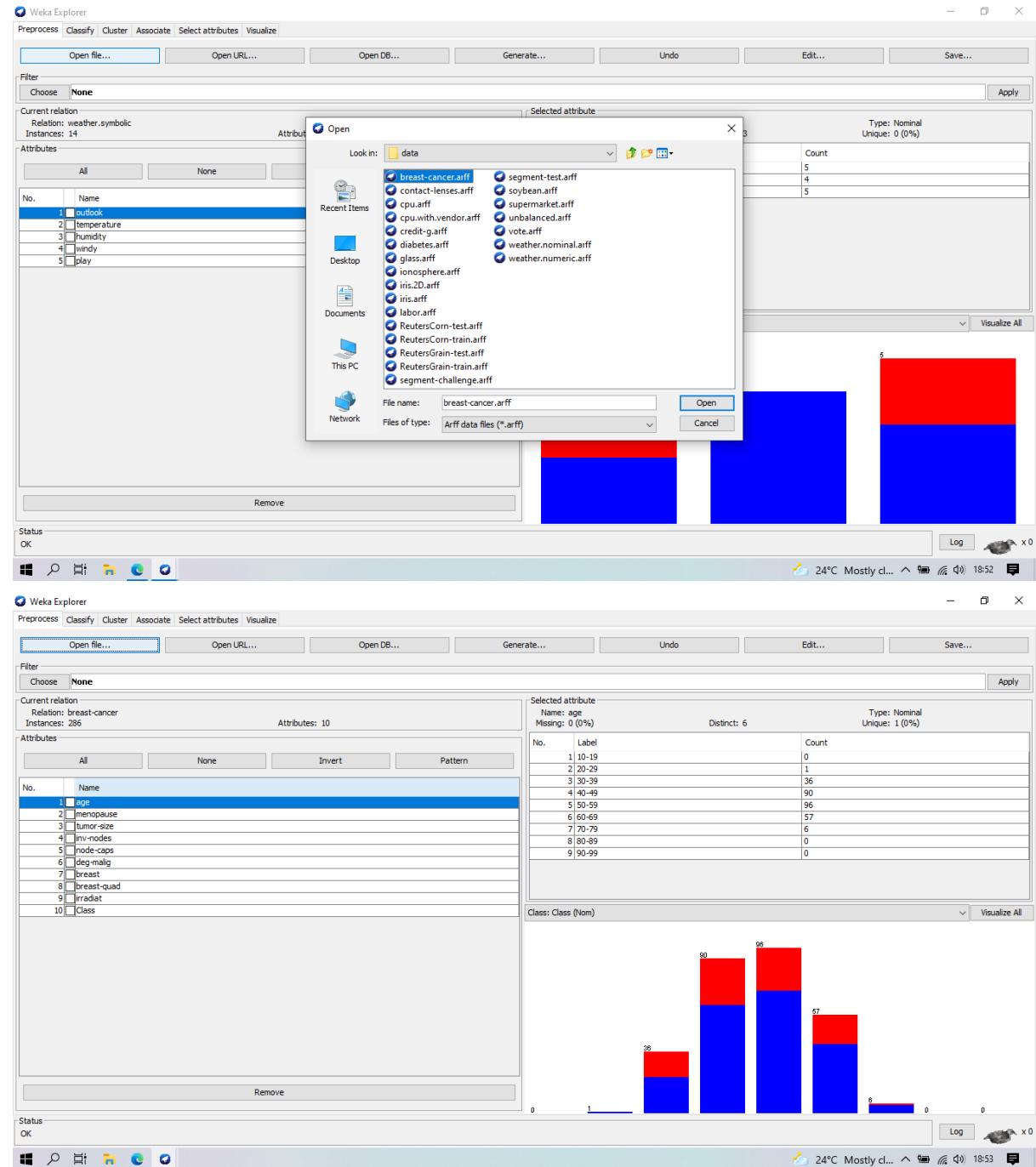
Log x 0



3.5 Pruning Decision Trees

Pruning Decision Trees dimanfaatkan untuk pemangkasan Tree untuk mencegah terlalu banyak cabang pada suatu Tree dan mengoptimalkan akurasi model yang dibuat.

Load dataset breast cancer.



Gunakan classifier J48. Jalankan, akurasi 75,5% jika dilakukan pruning (secara default pruning digunakan dalam melatih model).

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier Choose J48 -C 0.25 -M 2

Test options

- Use training set
- Supplied test set Set...
- Cross-validation Folds 10
- Percentage split % 66
- More options...

(Nom) Class Start Stop

Result list (right-click for options) 18:53:23 - trees.J48

Classifier output

```

I deg-malig = 3: recurrence-events (30.4/7.4)
node-caps = no: no-recurrence-events (228.39/53.4)

Number of Leaves : 4
Size of the tree : 6

Time taken to build model: 0.01 seconds

*** Stratified cross-validation ***
*** Summary ***

Correctly Classified Instances 216 75.5245 %
Incorrectly Classified Instances 70 24.4755 %
Kappa statistic 0.2826
Mean absolute error 0.3676
Root mean squared error 0.4324
Relative absolute error 87.8635 %
Root relative squared error 94.6093 %
Total Number of Instances 286

*** Detailed Accuracy By Class ***

      TP Rate FP Rate Precision Recall F-Measure ROC Area Class
      0.96 0.729 0.757 0.96 0.846 0.584 no-recurrence-events
      0.271 0.04 0.742 0.271 0.397 0.584 recurrence-events
      Weighted Avg. 0.755 0.524 0.752 0.755 0.713 0.584

*** Confusion Matrix ***

  a b <- classified as
193 8 | a = no-recurrence-events

```

Status OK

24°C Mostly cl... 18:53 Log x0

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier Choose J48 -C 0.25 -M 2

Test options

- Use training set
- Supplied test set Set...
- Cross-validation Folds 10
- Percentage split % 66
- More options...

(Nom) Class Start Stop

Result list (right-click for options) 18:53:23 - trees.J48

Classifier output

```

I deg-malig = 3: recurrence-events (30.4/7.4)
node-caps = no: no-recurrence-events (228.39/53.4)

Number of Leaves : 4
Size of the tree : 6

Time taken to build model:
*** Stratified cross-validation ***
*** Summary ***

Correctly Classified Instances 216 75.5245 %
Incorrectly Classified Instances 70 24.4755 %
Kappa statistic 0.2826
Mean absolute error 0.3676
Root mean squared error 0.4324
Relative absolute error 87.8635 %
Root relative squared error 94.6093 %
Total Number of Instances 286

*** Detailed Accuracy By Class ***

      TP Rate FP Rate Precision Recall F-Measure ROC Area Class
      0.96 0.729 0.757 0.96 0.846 0.584 no-recurrence-events
      0.271 0.04 0.742 0.271 0.397 0.584 recurrence-events
      Weighted Avg. 0.755 0.524 0.752 0.755 0.713 0.584

*** Confusion Matrix ***

  a b <- classified as
193 8 | a = no-recurrence-events

```

Tree View

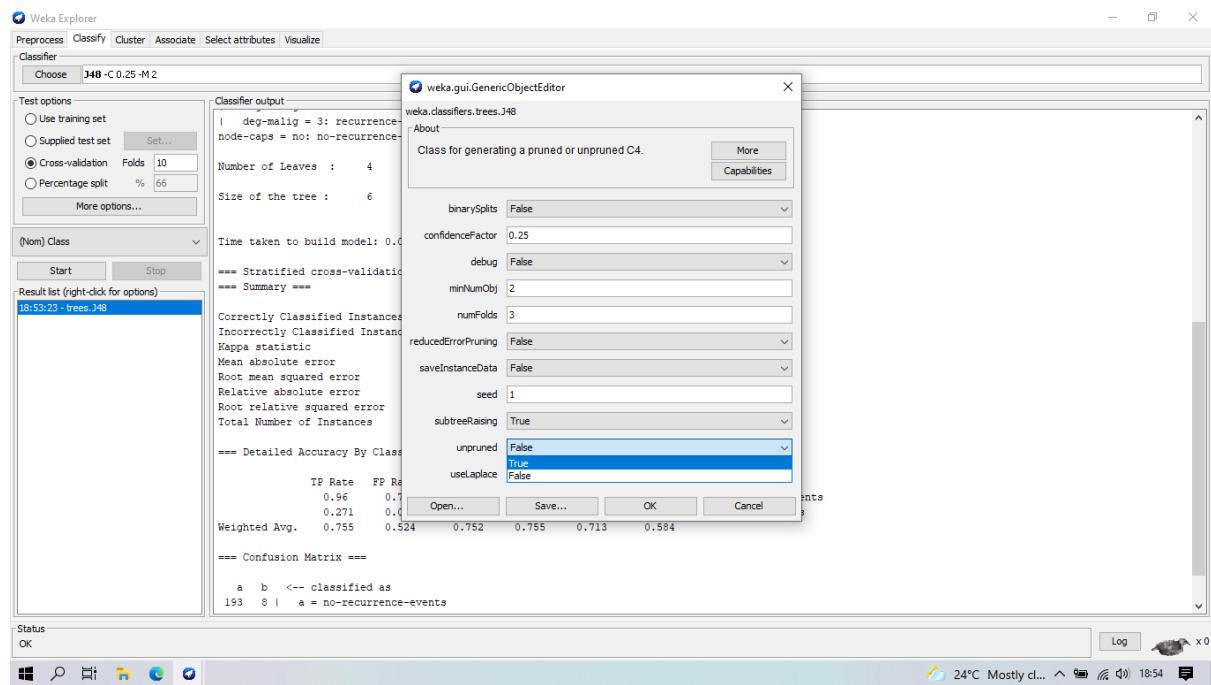
```

graph TD
    Root((node-caps)) -- "yes" --> Node1[deg-malig]
    Root -- "no" --> Node2[no-recurrence-events (228.39/53.4)]
    Node1 -- "1" --> Node3[recurrence-events (1.0)]
    Node1 -- "2" --> Node4[no-recurrence-events (26)]

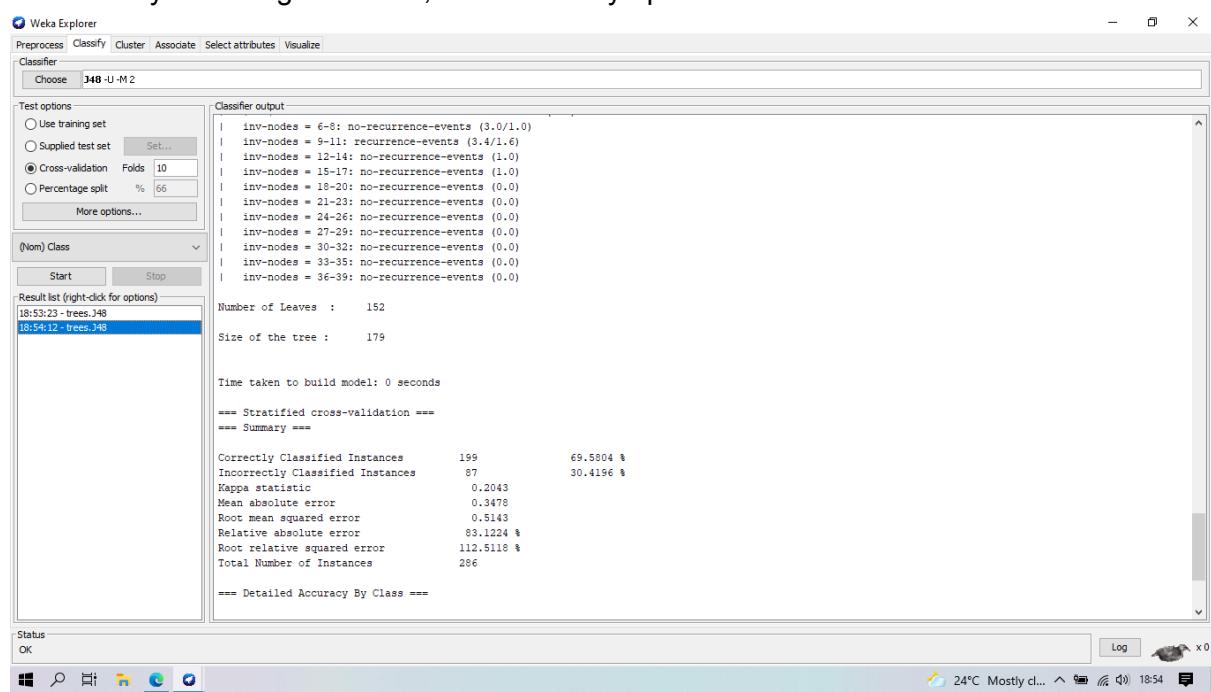
```

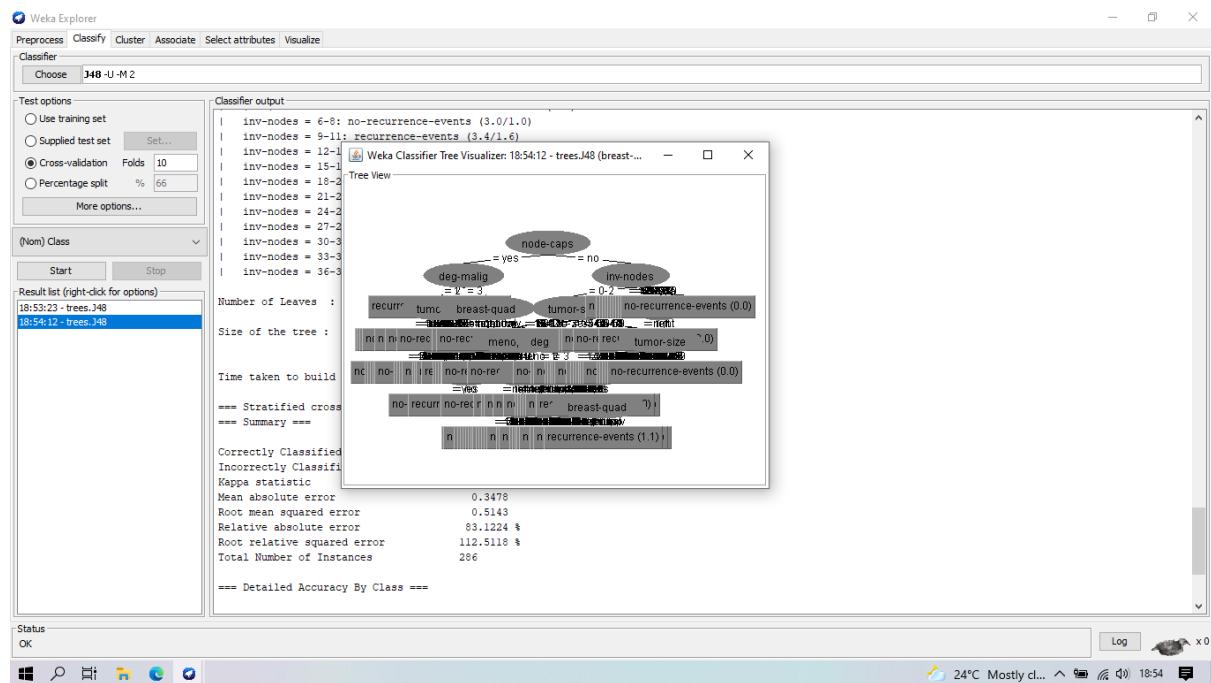
Status OK

24°C Mostly cl... 18:53 Log x0



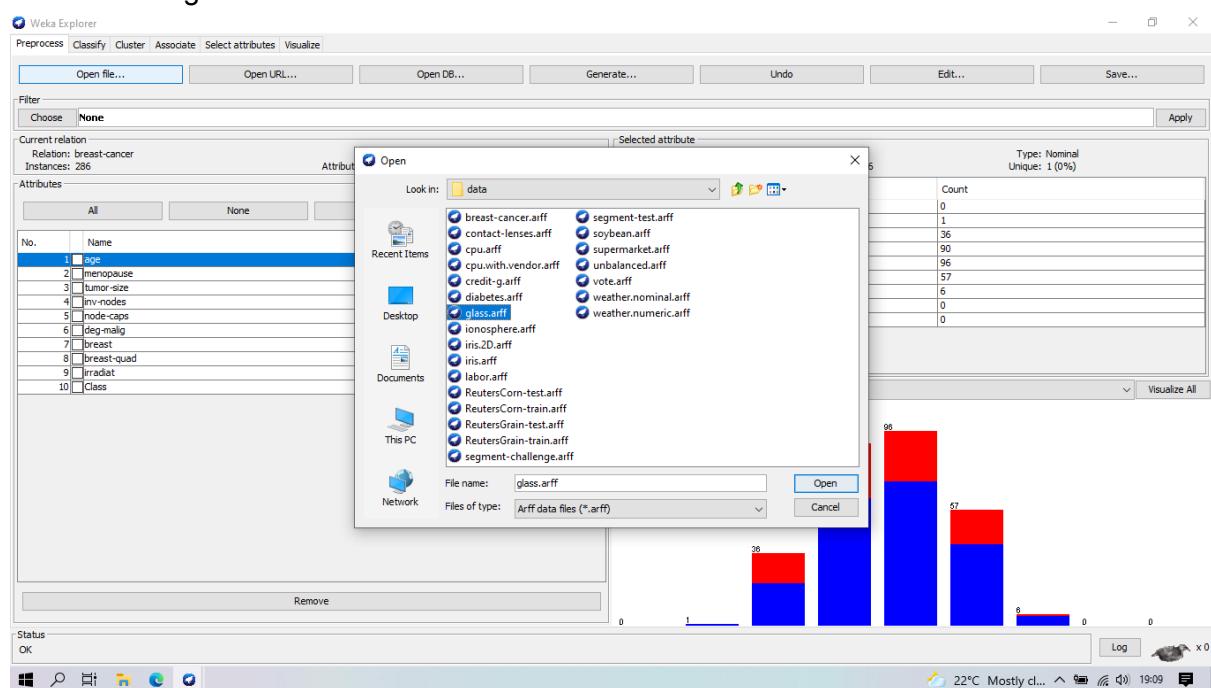
Ketika diatur pruningnya menjadi FALSE, hasil yang didapatkan seperti gambar dibawah. Terlalu banyak ranting dan daun, dan akurasinya pun menurun.

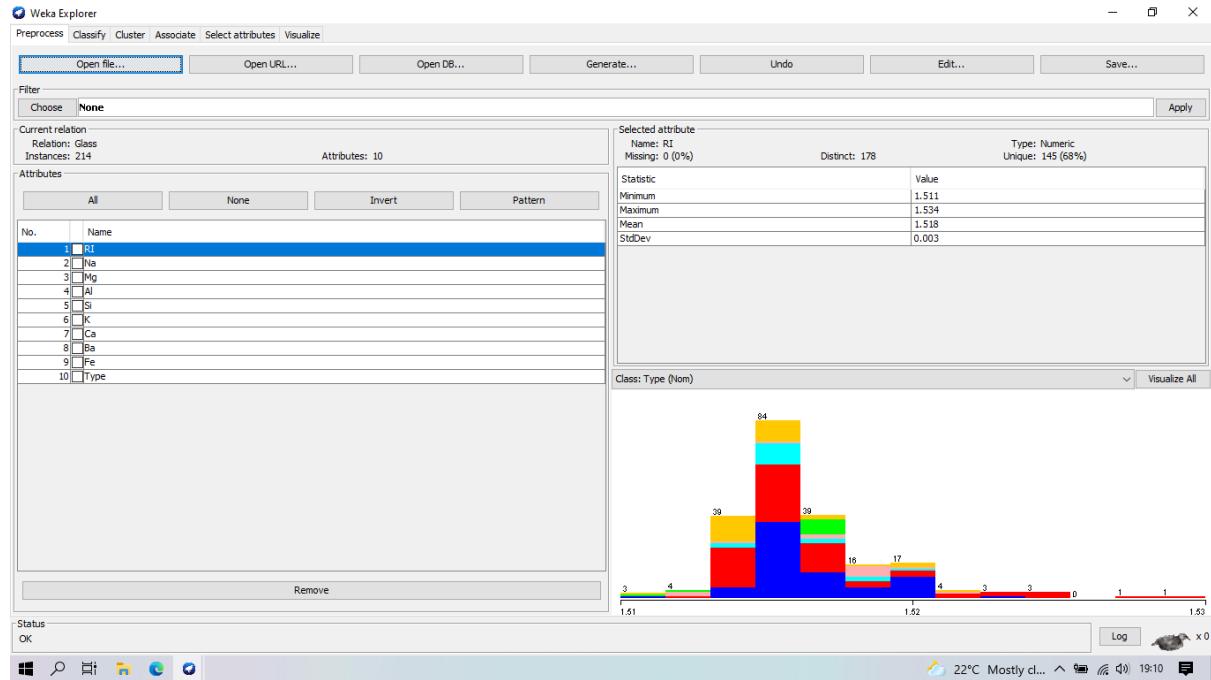




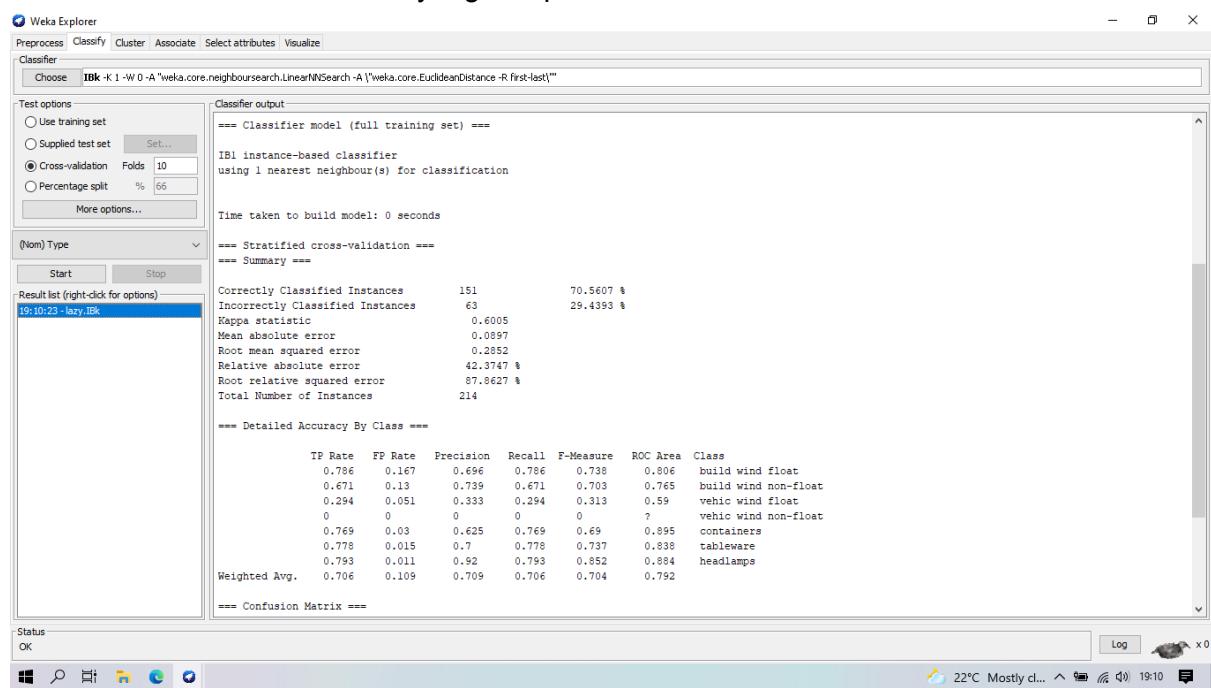
3.6 Nearest Neighbor

Load dataset glass.

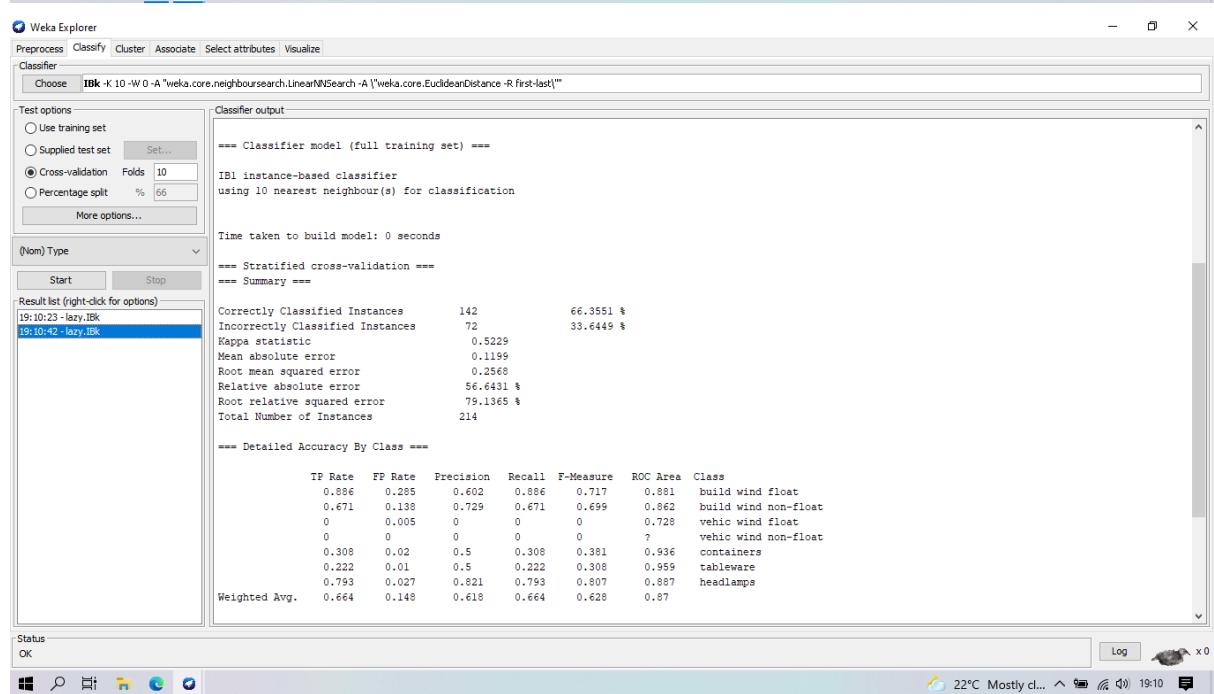
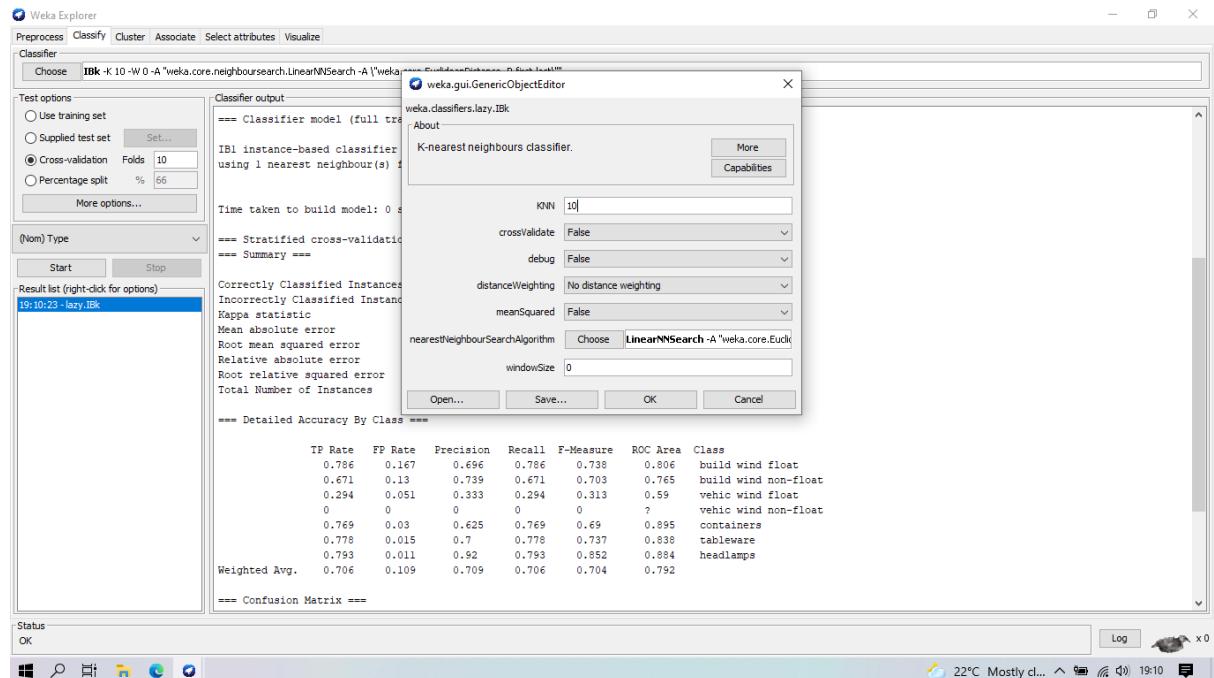




Gunakan classifier IBk. akurasi yang didapatkan 70,5%.



Ubah nilai KNN menjadi 10. (Hal ini berarti suatu instance baru yang ditambahkan, akan mencari 10 instance dengan jarak terdekat). Akurasi sekitar 66%, lebih kecil dibandingkan KNN bernilai 1.



Ubah nilai KNN menjadi 100. (Hal ini berarti suatu instance baru yang ditambahkan, akan mencari 100 instance dengan jarak terdekat). Akurasi semakin jelek, hanya sekitar 34,6%,

The screenshot displays two separate runs of the Weka Explorer interface, each showing the results of a K-Nearest Neighbors (IBk) classification model.

Top Run (K=100):

- Classifier Output:**
 - IBk - K 100 - W 0 -A "weka.core.neighboursearch.LinearNNSearch -A "\weka.core.EuclideanDistance -R first-last"
 - IBk instance-based classifier using 100 nearest neighbour(s)
 - K-nearestneighbours classifier.
 - Configuration:**
 - KNN: 100
 - crossValidate: False
 - debug: False
 - distanceWeighting: No distance weighting
 - meanSquared: False
 - nearestNeighbourSearchAlgorithm: Choose LinearNNSearch -A "\weka.core.EuclideanDistance -R first-last"
 - windowSize: 0
- Summary:**
 - Time taken to build model: 0 seconds
 - Correctly Classified Instances: 74 (34.5794 %)
 - Incorrectly Classified Instances: 140 (65.4206 %)
 - Kappa statistic: 0.0112
 - Mean absolute error: 0.1887
 - Root mean squared error: 0.3099
 - Relative absolute error: 89.1058 %
 - Root relative squared error: 95.4781 %
 - Total Number of Instances: 214
- Detailed Accuracy By Class:**

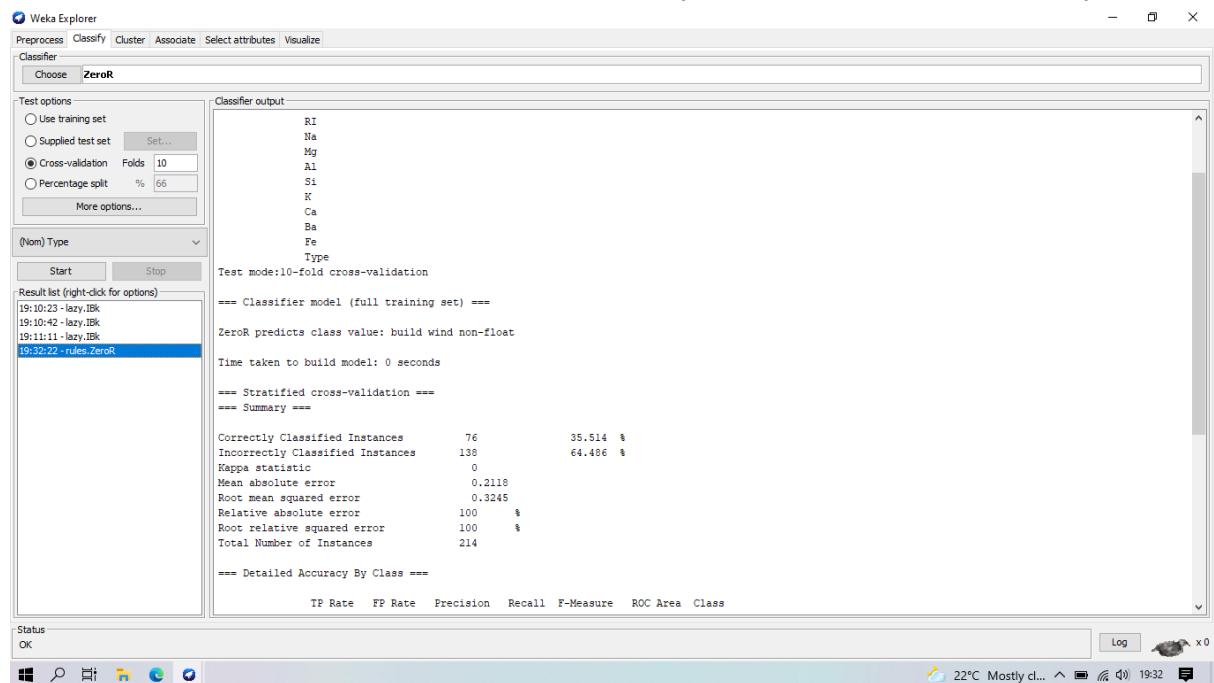
TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.886	0.285	0.602	0.886	0.717	0.881	build wind float
0.671	0.138	0.729	0.671	0.699	0.862	build wind non-float
0	0.005	0	0	0	0.728	vehic wind float
0	0	0	0	0	?	vehic wind non-float
0.308	0.02	0.5	0.308	0.381	0.936	containers
0.222	0.01	0.5	0.222	0.308	0.959	tableware
0.793	0.027	0.821	0.793	0.807	0.887	headlamps
Weighted Avg.	0.664	0.148	0.618	0.664	0.628	

Bottom Run (K=10):

- Classifier Output:**
 - IBk - K 10 - W 0 -A "weka.core.neighboursearch.LinearNNSearch -A "\weka.core.EuclideanDistance -R first-last"
 - IBk instance-based classifier using 10 nearest neighbour(s) for classification
 - Test mode:10-fold cross-validation
 - K-nearestneighbours classifier.
- Summary:**
 - Time taken to build model: 0 seconds
 - Correctly Classified Instances: 74 (34.5794 %)
 - Incorrectly Classified Instances: 140 (65.4206 %)
 - Kappa statistic: 0.0112
 - Mean absolute error: 0.1887
 - Root mean squared error: 0.3099
 - Relative absolute error: 89.1058 %
 - Root relative squared error: 95.4781 %
 - Total Number of Instances: 214
- Detailed Accuracy By Class:**

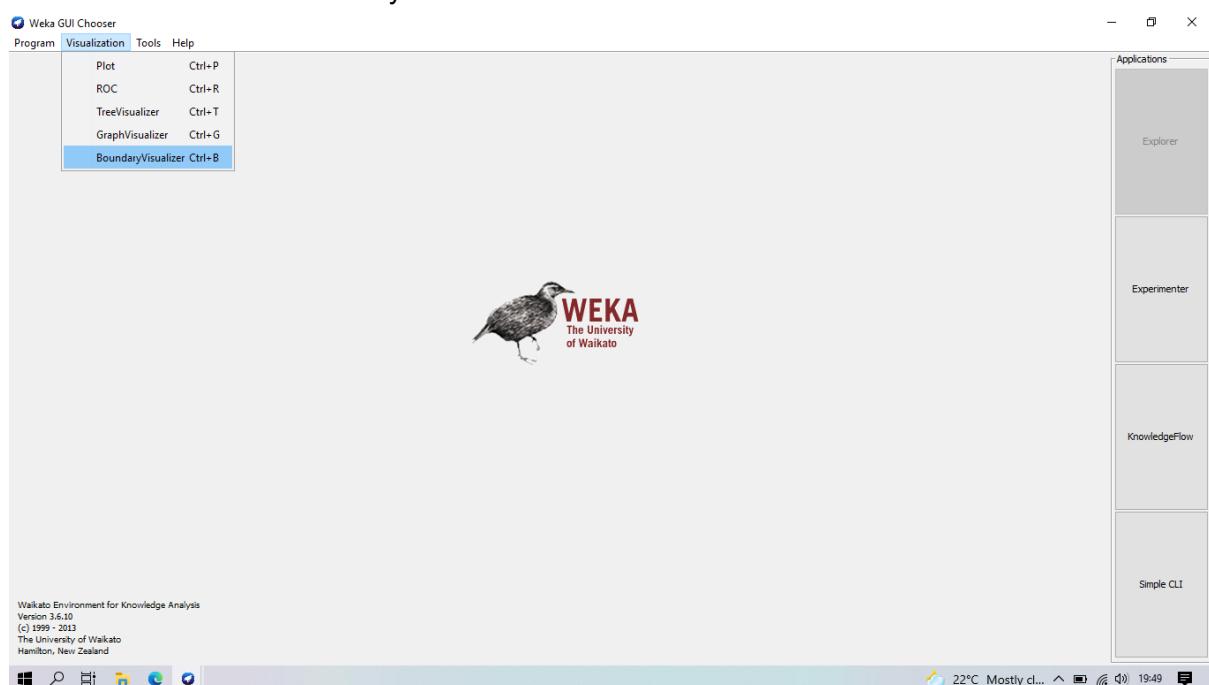
TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.886	0.285	0.602	0.886	0.717	0.881	build wind float
0.671	0.138	0.729	0.671	0.699	0.862	build wind non-float
0	0.005	0	0	0	0.728	vehic wind float
0	0	0	0	0	?	vehic wind non-float
0.308	0.02	0.5	0.308	0.381	0.936	containers
0.222	0.01	0.5	0.222	0.308	0.959	tableware
0.793	0.027	0.821	0.793	0.807	0.887	headlamps
Weighted Avg.	0.664	0.148	0.618	0.664	0.628	

Setelah dianalisis, ketika nilai k = 100, nilai akurasinya mendekati baseline accuracy.

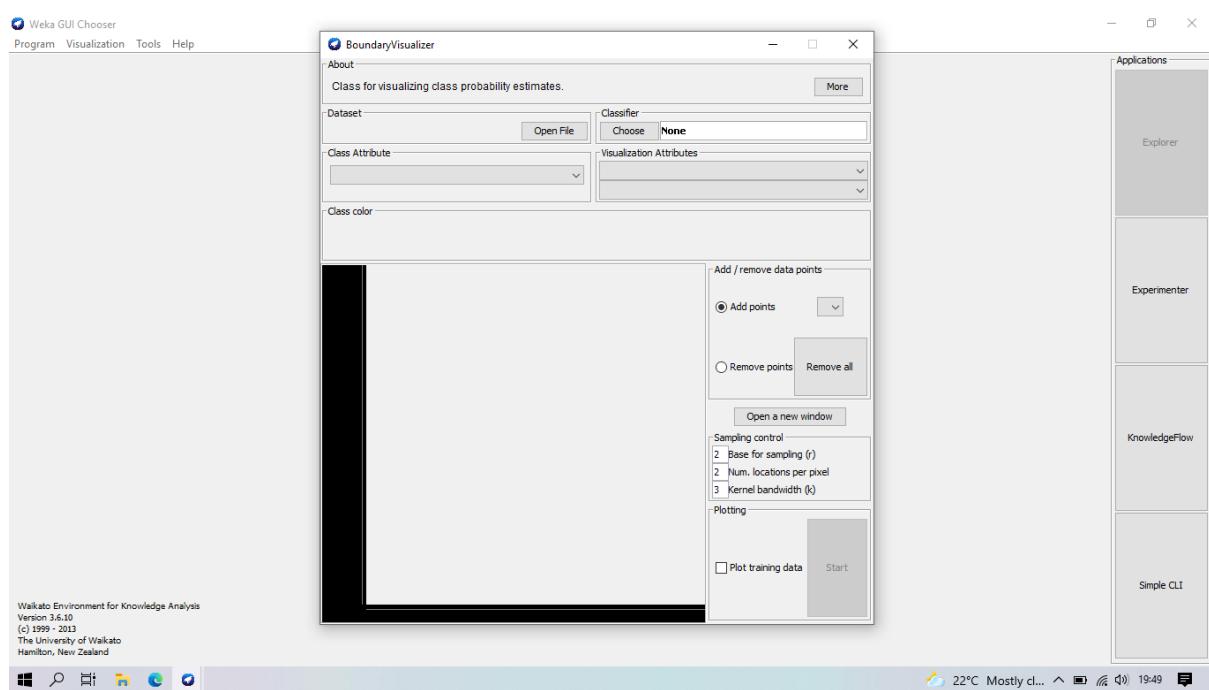


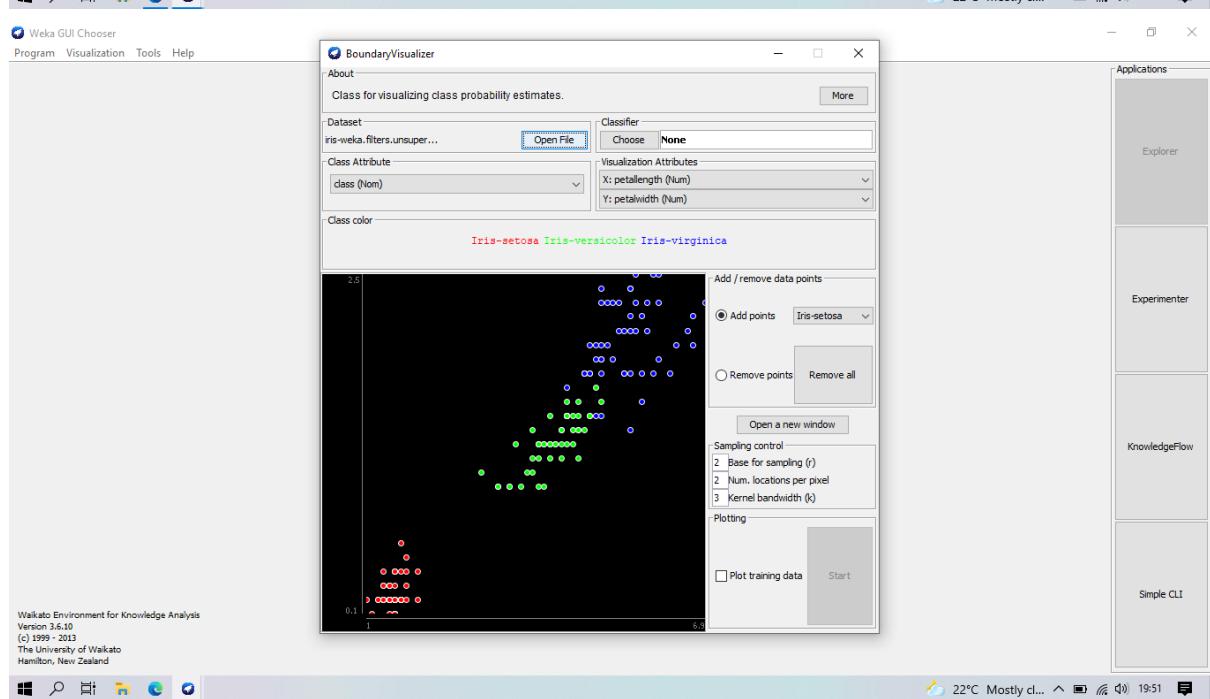
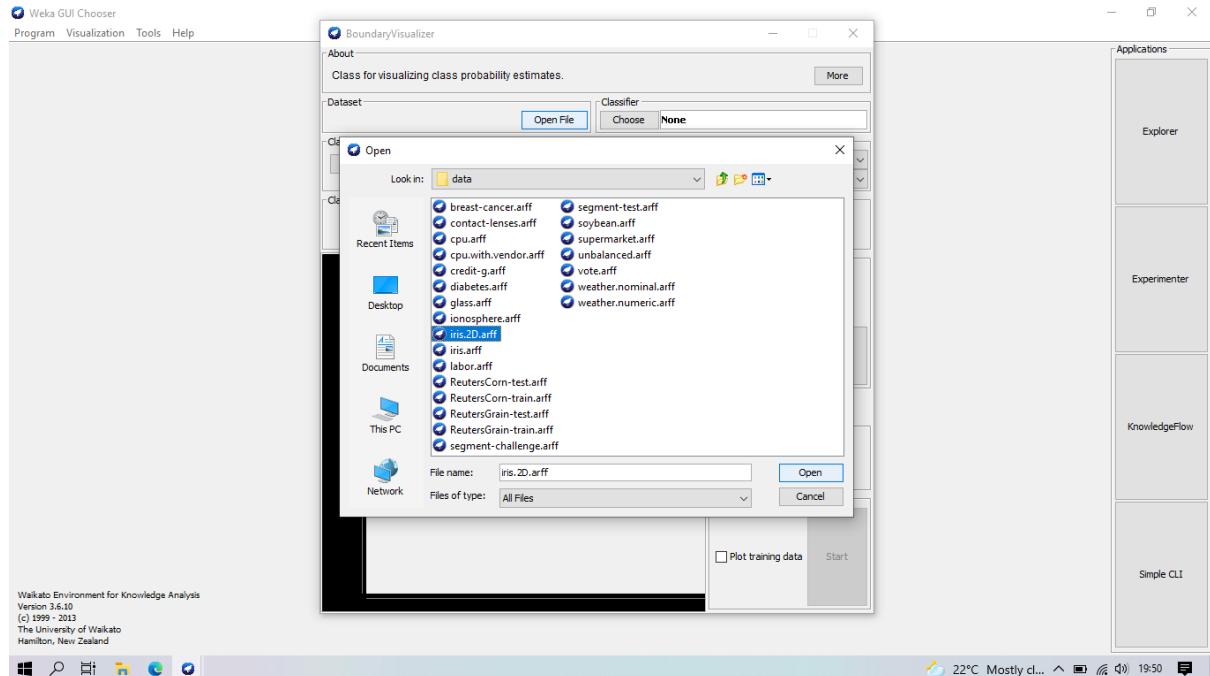
4.1 Classification Boundaries

Buka Visualization > BoundaryVisualizer

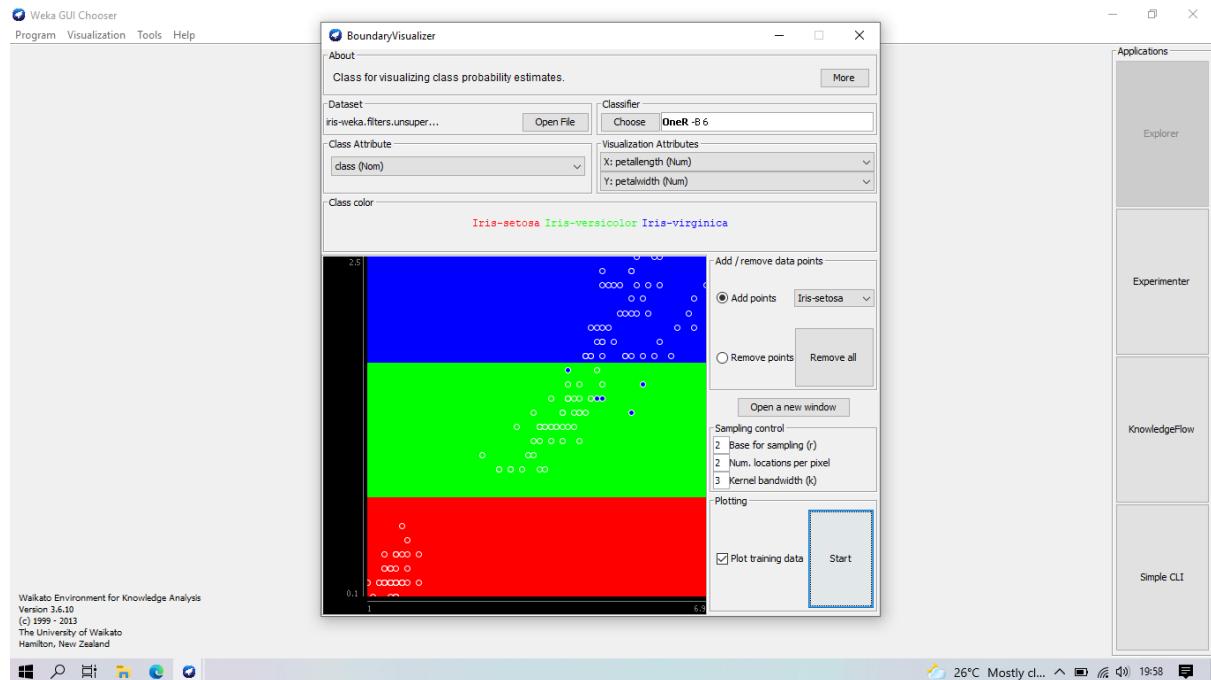


Buka dataset iris 2D.

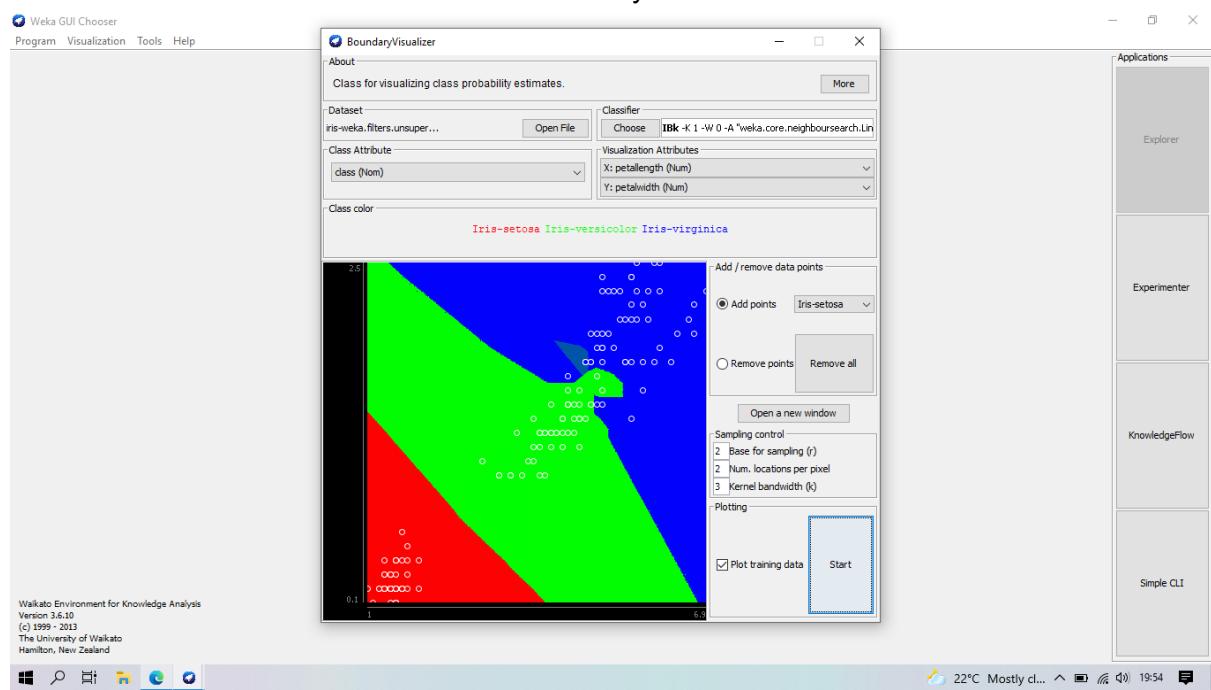




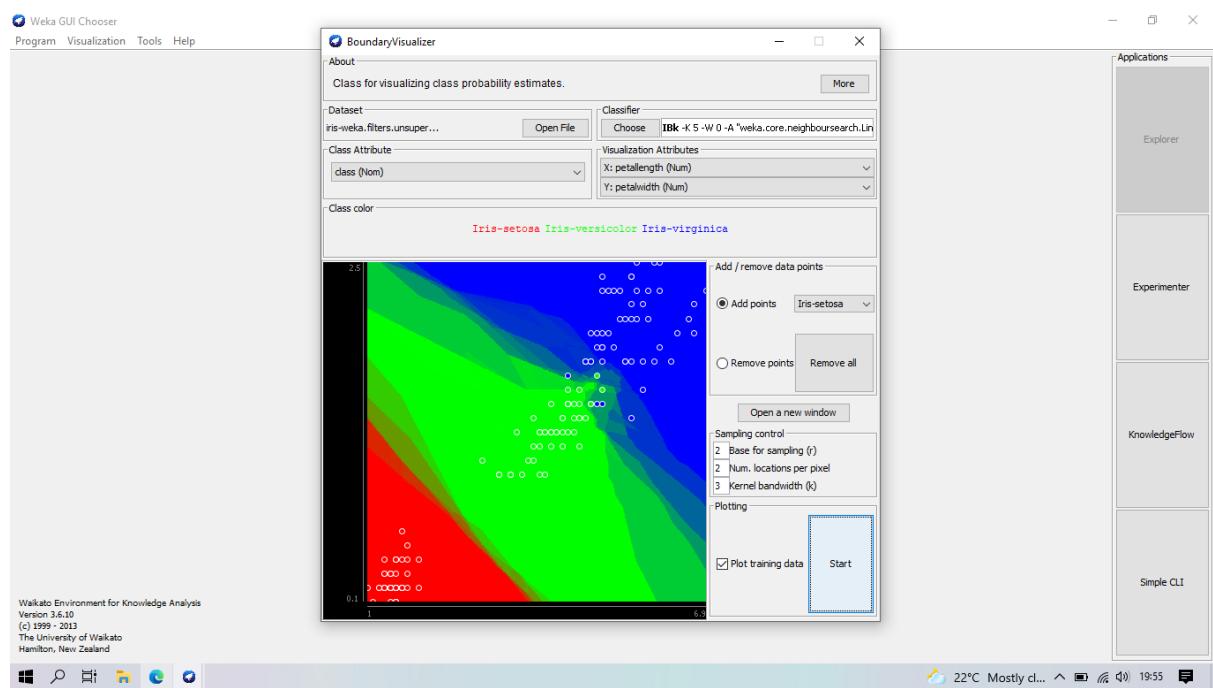
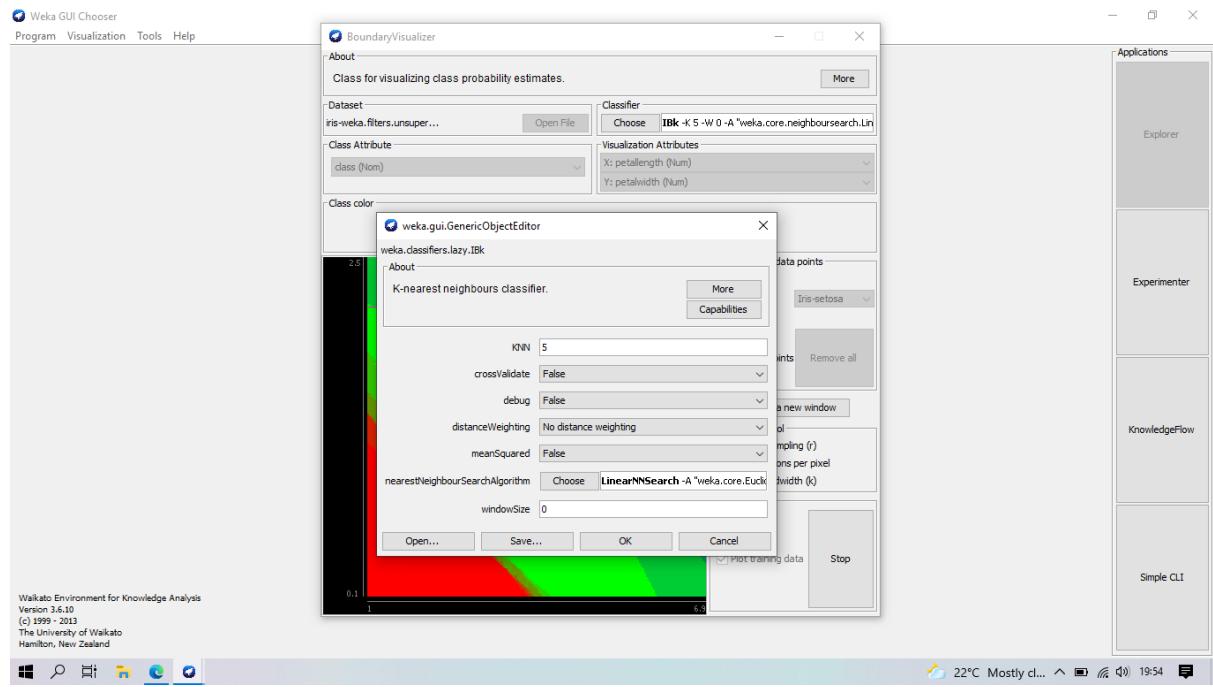
Gunakan classifier OneR. Berikut hasil visualisasinya:



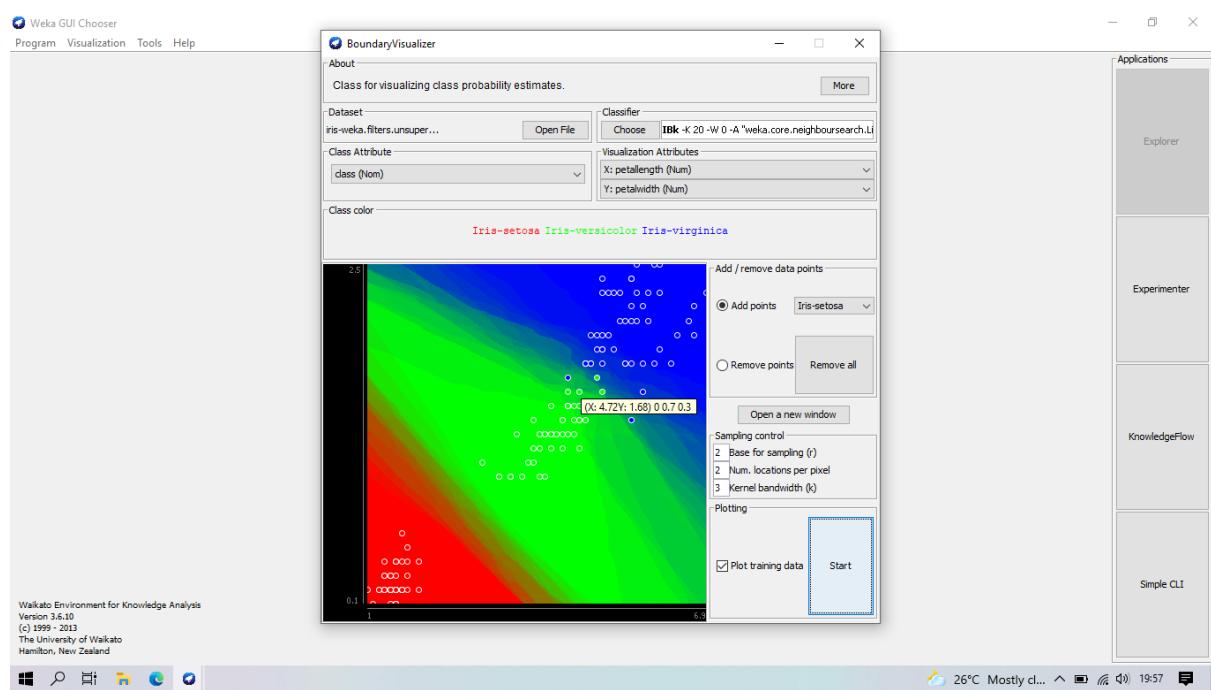
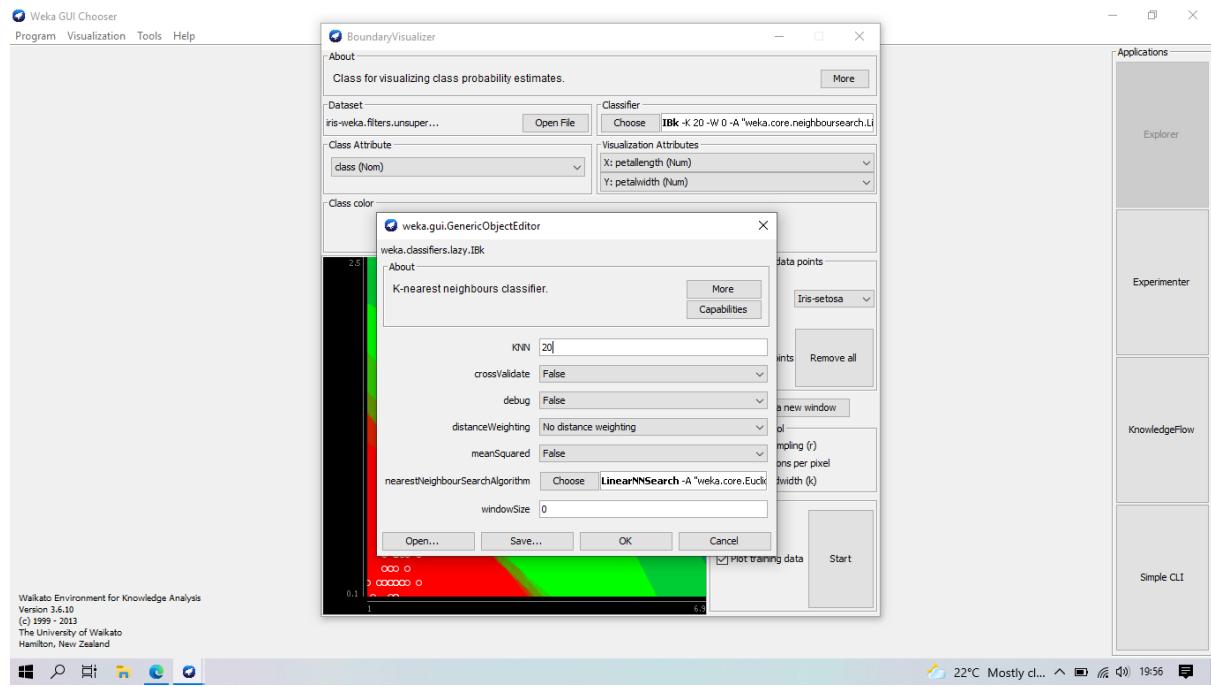
Gunakan classifier IBk. Berikut hasil visualisasinya:



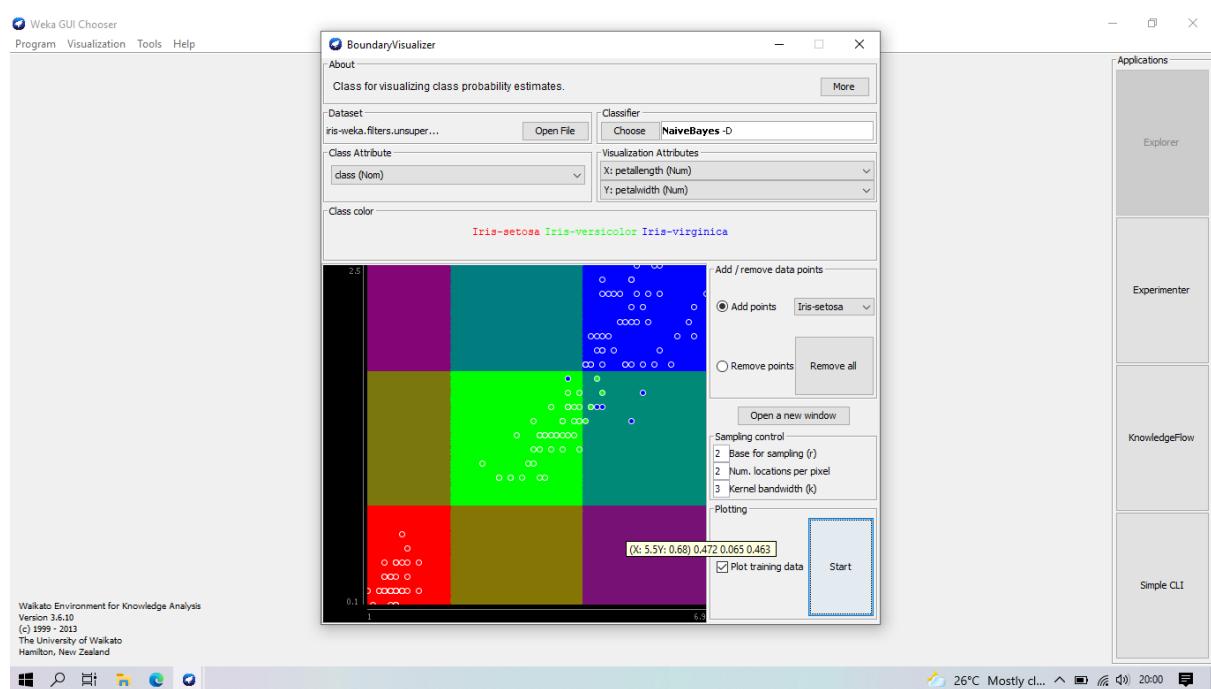
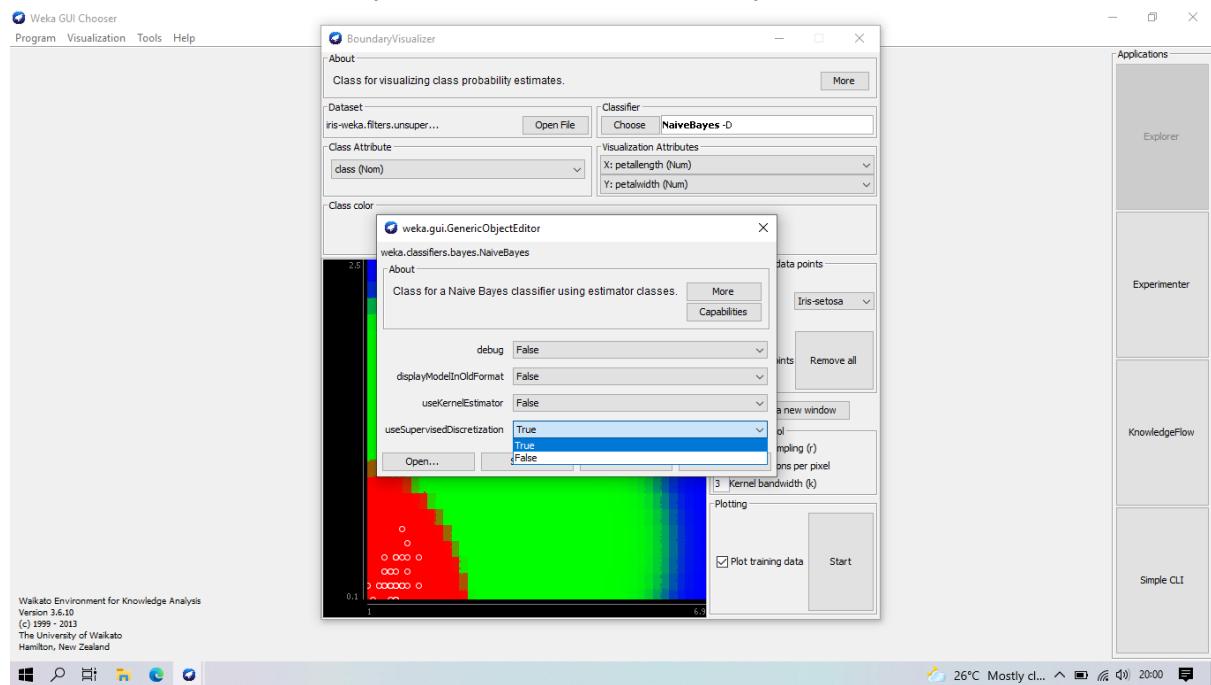
Ubah nilai KNN menjadi 5. Setelah nilai KNN berubah menjadi 5, ada daerah yang transisi antara merah dan hijau, atau warna lainnya. Berikut hasil visualisasinya:



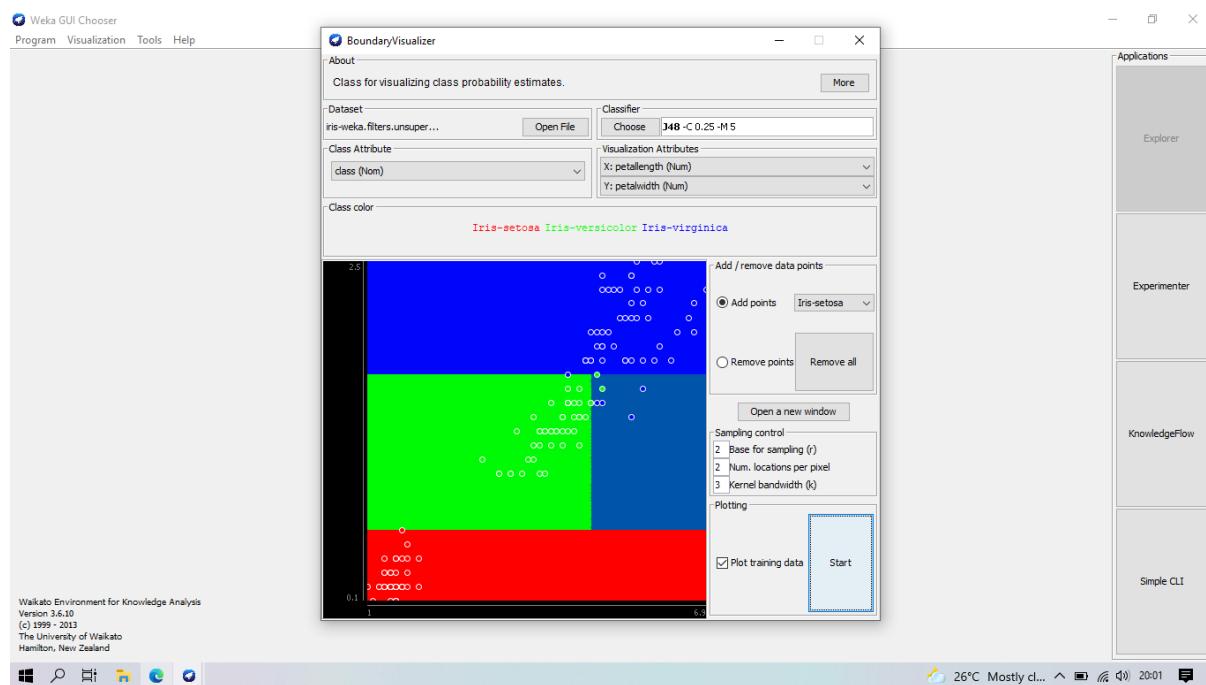
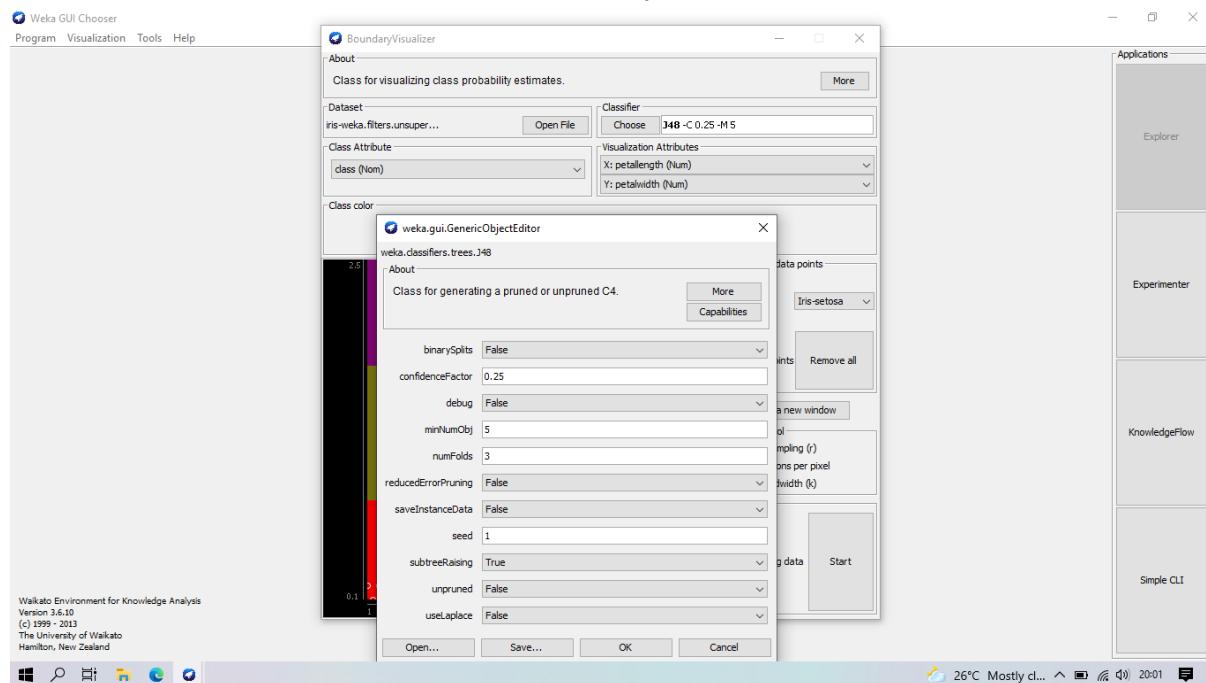
Ubah nilai KNN menjadi 20. Setelah nilai KNN berubah menjadi 20, daerah yang transisi antara merah dan hijau, atau warna lainnya menjadi lebih lebar. Berikut hasil visualisasinya:



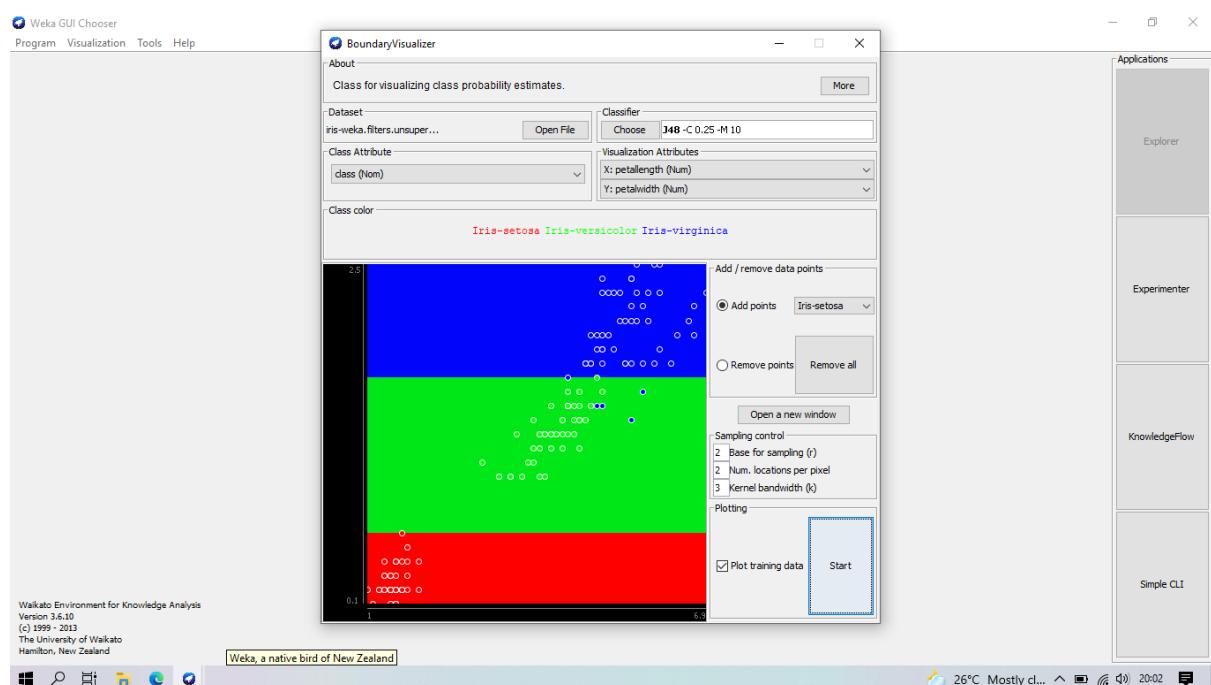
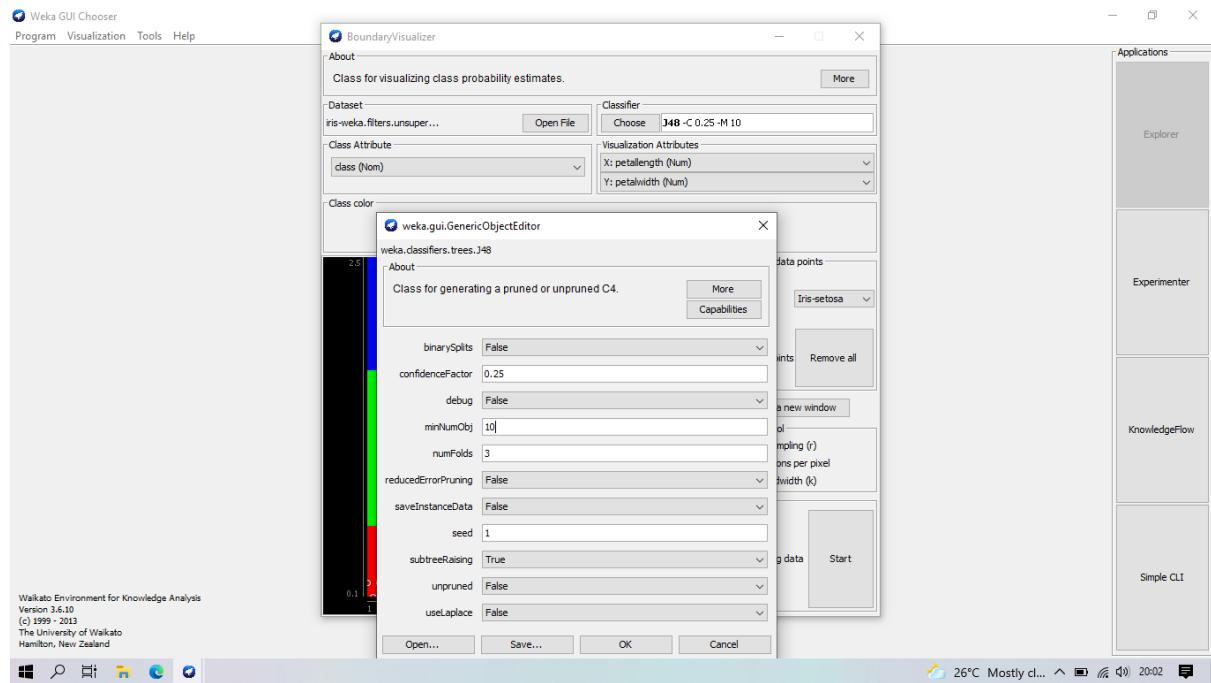
Gunakan Classifier NaiveBayes. Berikut hasil visualisasinya:



Gunakan Classifier J48. Berikut hasil visualisasinya:

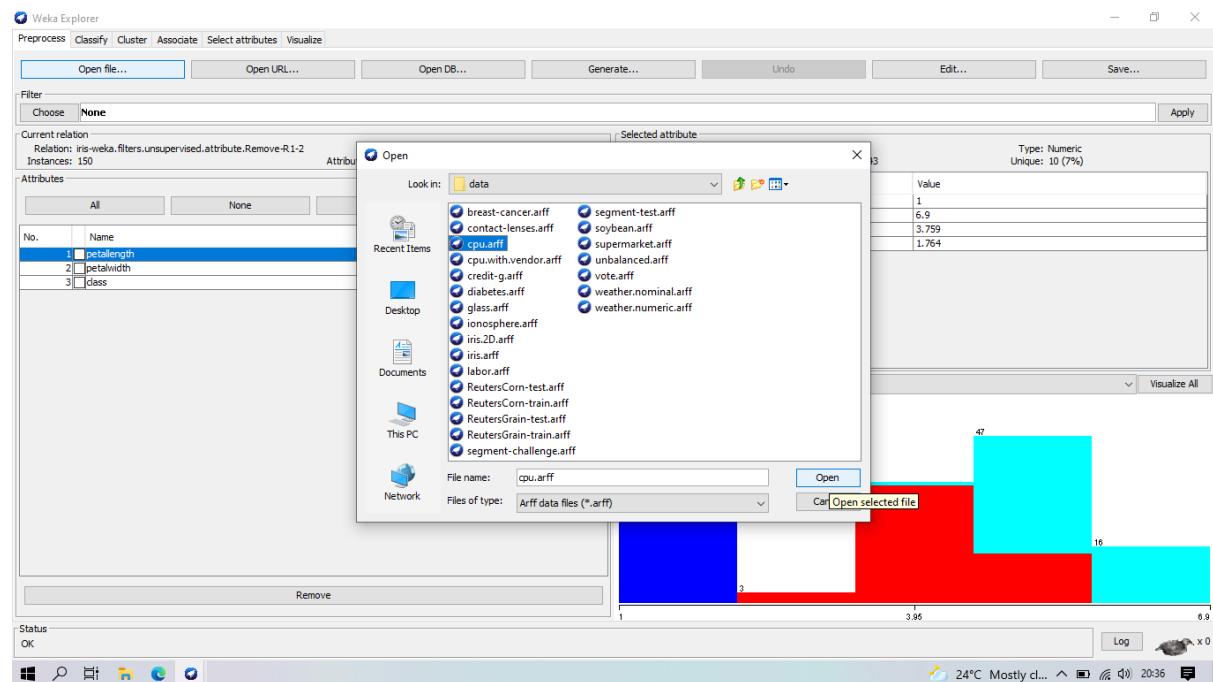


Klik classifier J48 untuk melakukan konfigurasi. Atur minNumObj menjadi 10. Berikut hasil visualisasinya:

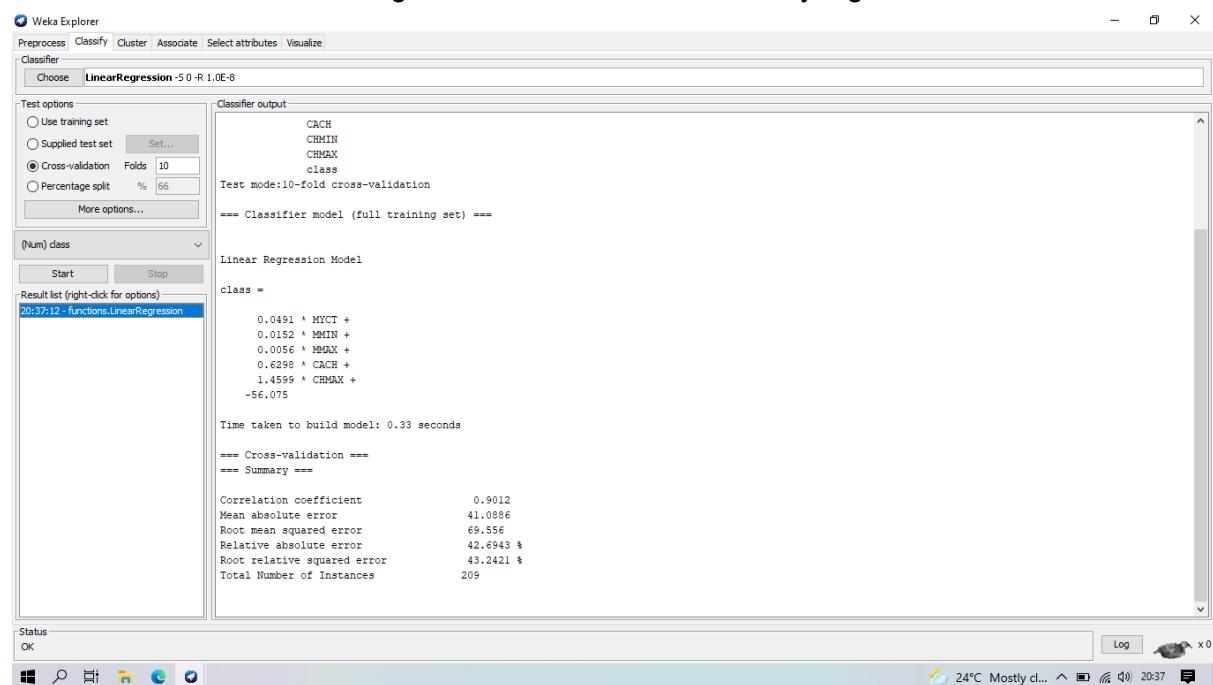


4.2 Linear Regression

Buka dataset CPU.



Gunakan Classifier Linear Regression. Berikut rincian model yang dihasilkan.



Berikut adalah classifier M5P, merupakan bentuk pohon dari model regresi. Berikut hasil dari penggunaan classifier M5P.

```

Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier
Choose M5P-M 4.0

Test options
 Use training set
 Supplied test set Set...
 Cross-validation Folds 10
 Percentage split % 66
More options...

Classifier output
LM num: 5
class =
-0.4882 * MYCT
+ 0.0218 * MMIN
+ 0.003 * MMAX
+ 0.3865 * CACH
- 1.3252 * CHMIN
+ 3.3671 * CHMAX
- 51.8474

Number of Rules : 5

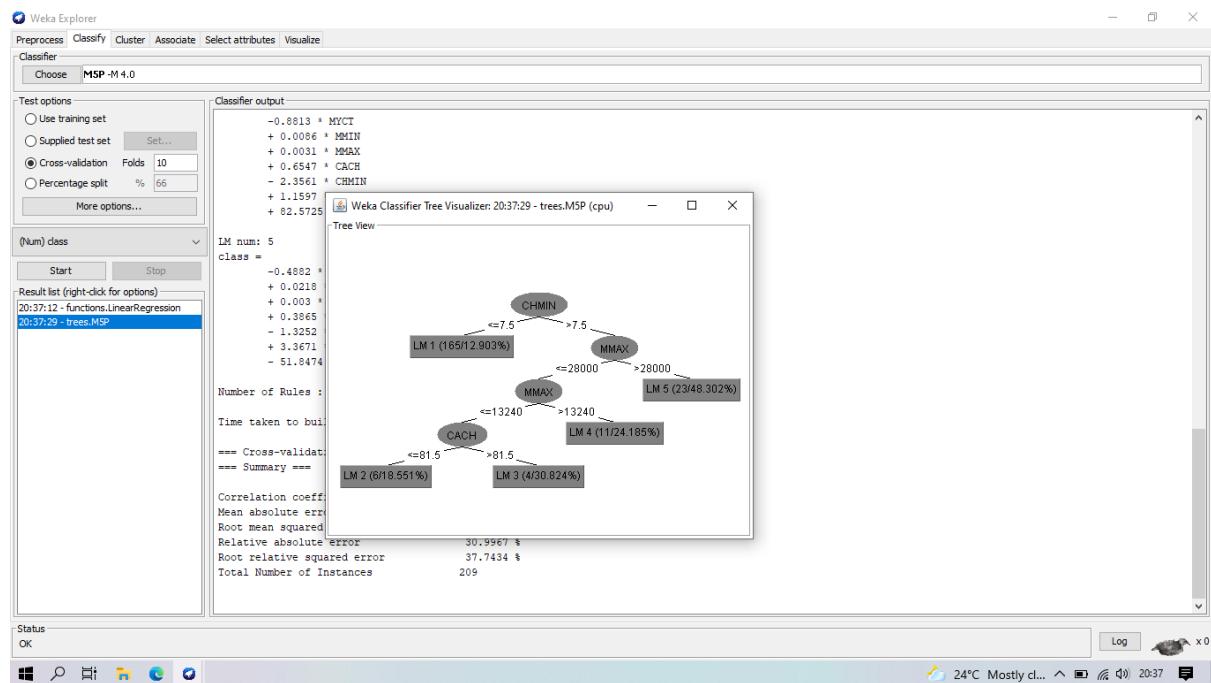
Time taken to build model: 0.12 seconds

==== Cross-validation ====
==== Summary ===

Correlation coefficient 0.9274
Mean absolute error 29.8309
Root mean squared error 60.7112
Relative absolute error 30.9967 %
Root relative squared error 37.7434 %
Total Number of Instances 209

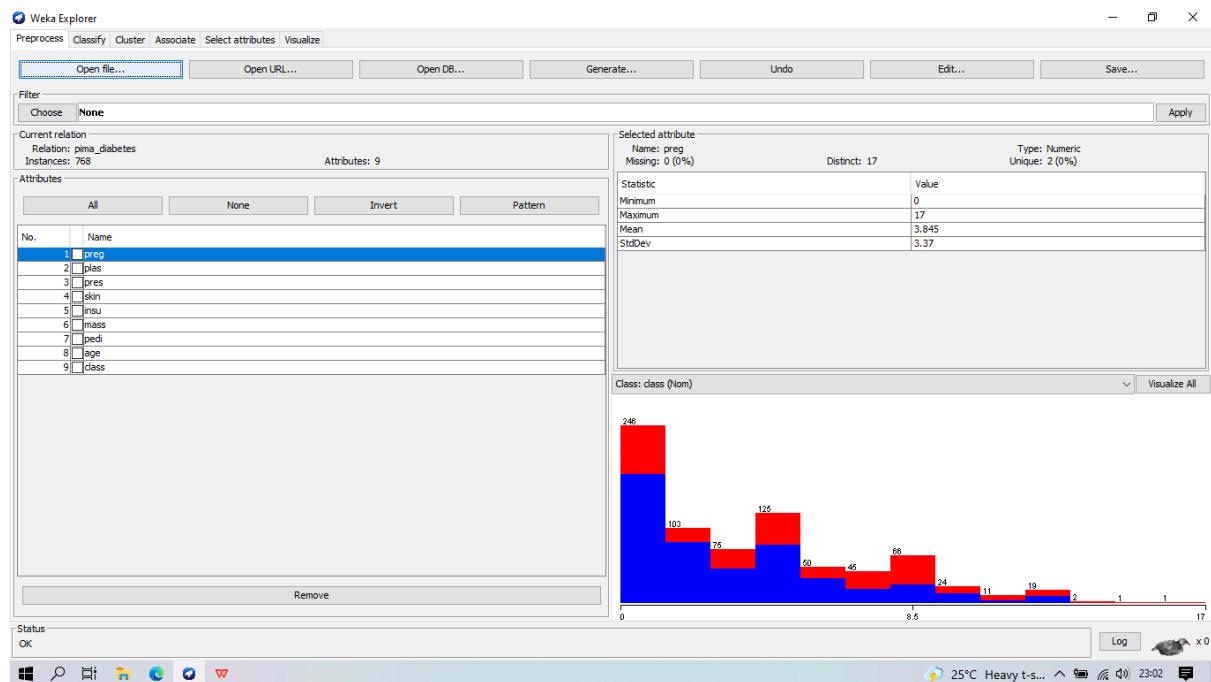
```

Berikut adalah visualisasi pohnnya.

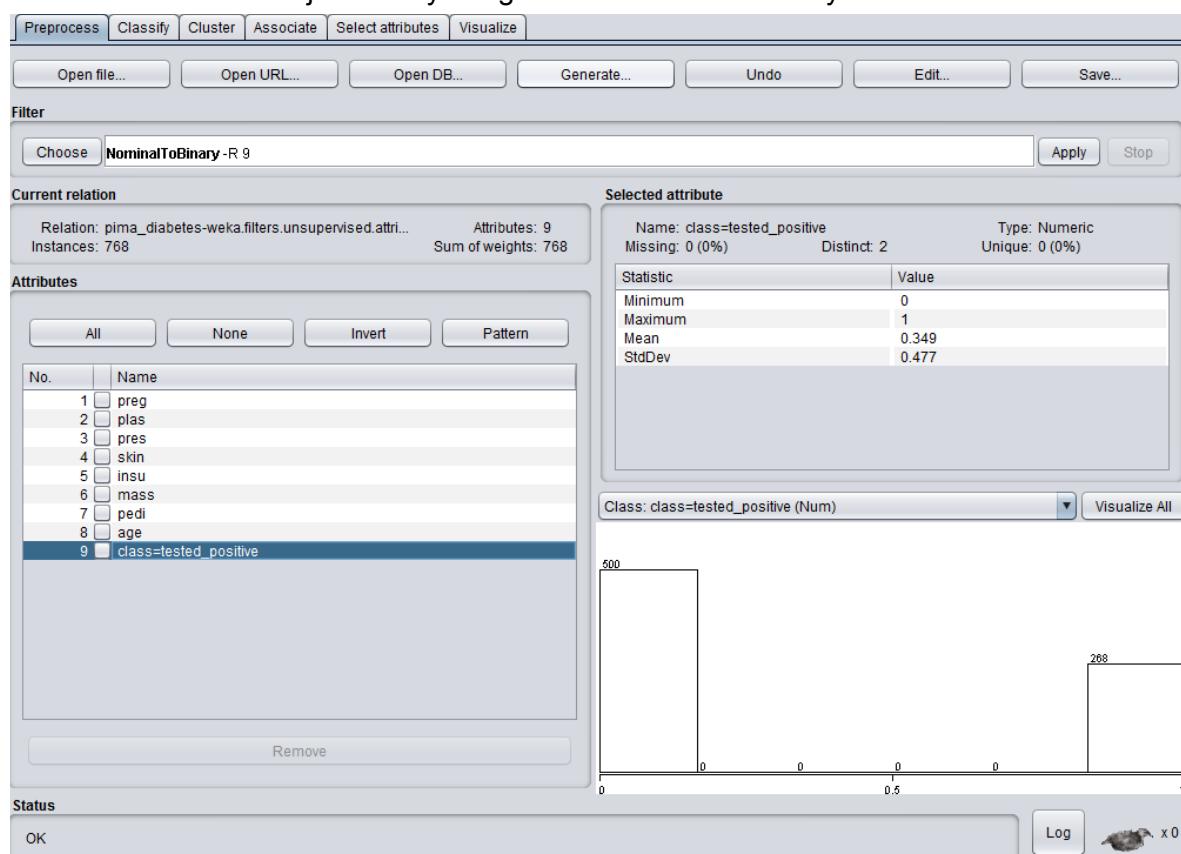


4.3 Classification by Regression

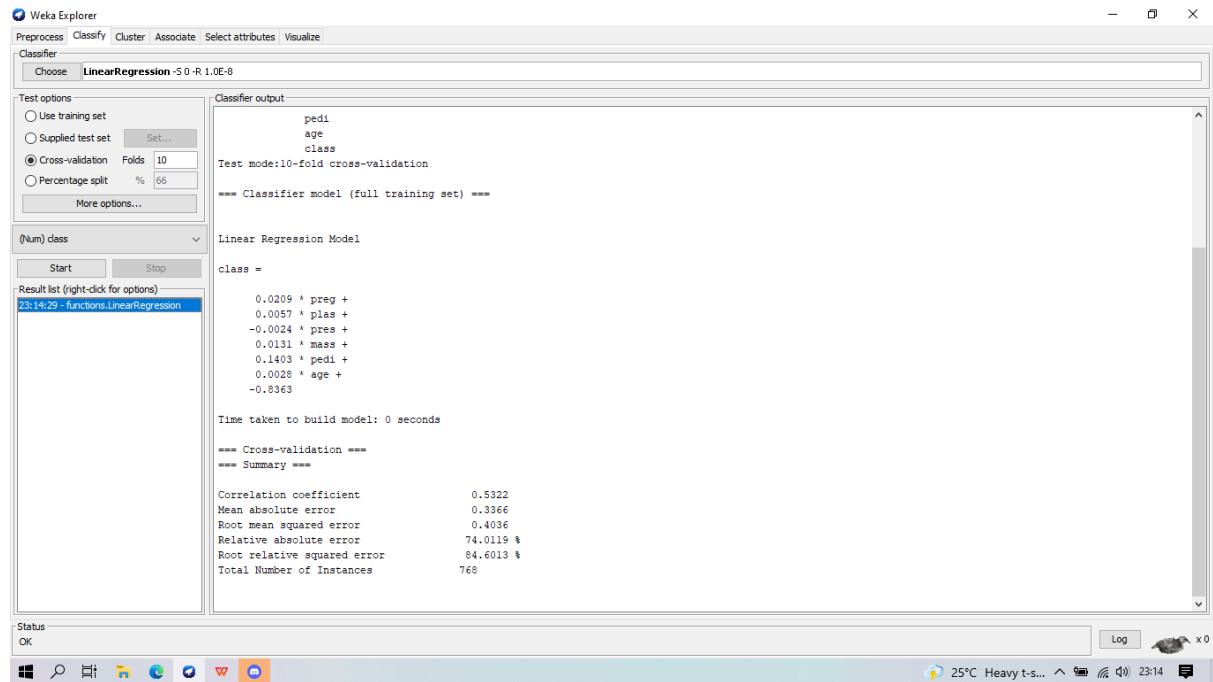
Buka dataset diabetes



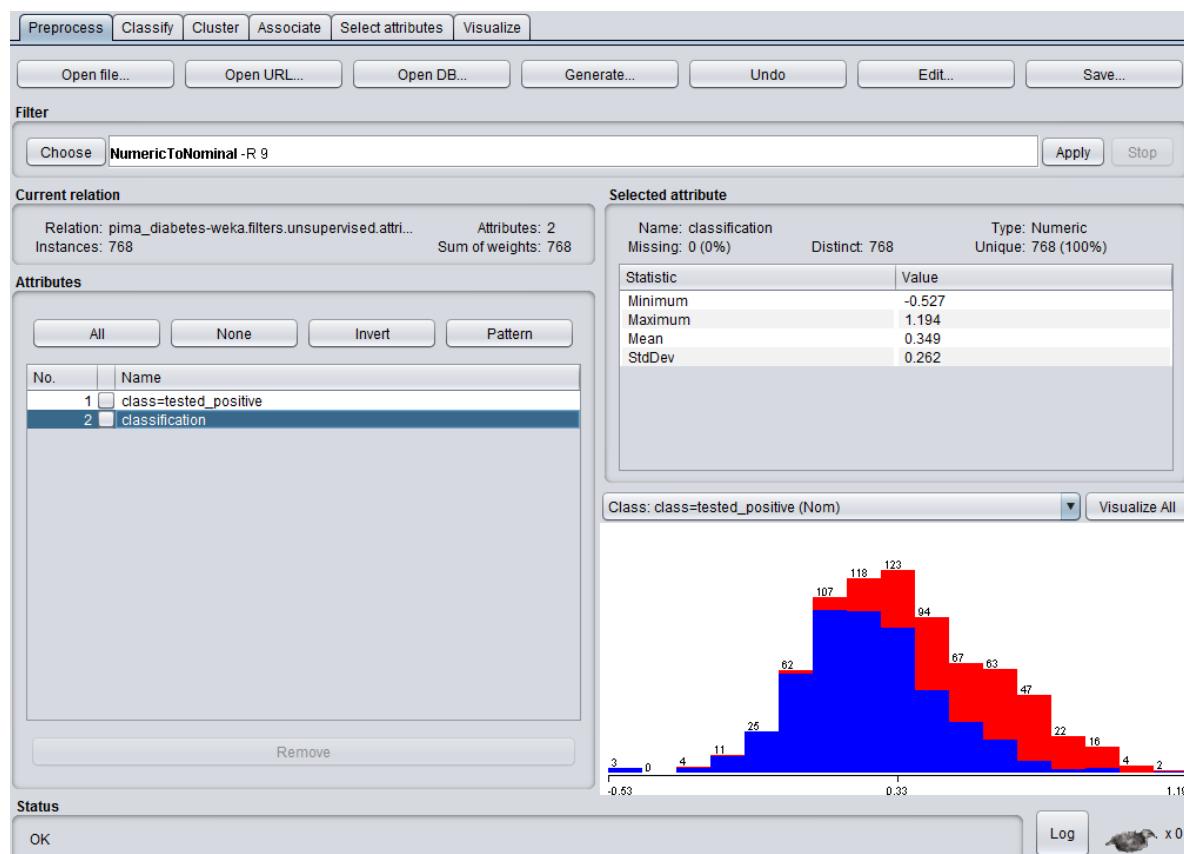
Ubah atribut kelas menjadi binary dengan filter NominalToBinary.



Lakukan Classifier menggunakan Linear Regression



Modifikasi dataset dengan ditambah atribut classification yang berasal dari filer addClassification dengan classifier Linear Regression dan parameter outputClassification=True, pengembalian atribut class menjadi nominal, dan menghapus atribut selain kedua atribut tersebut.



Lakukan classifier OneR dengan parameter minBucketSize=100.

4.4 Logistic Regression

Buka Dataset Diabetes. Kemudian gunakan classifier NaiveBayes dengan metode 90% percentage split.

Gunakan classifier ZeroR dengan metode 90% percentage split.

The screenshot shows the Weka interface with the 'Classify' tab selected. In the 'Classifier' panel, 'ZeroR' is chosen. The 'Test options' section has 'Percentage split' selected at 90%. The 'Classifier output' panel displays a table of predictions for a test split:

inst#	actual	predicted	error	prediction
1	1:tested_negative	1:tested_negative		0.648
2	2:tested_positive	1:tested_negative	+	0.648
3	1:tested_negative	1:tested_negative		0.648
4	1:tested_negative	1:tested_negative		0.648
5	2:tested_positive	1:tested_negative	+	0.648
6	1:tested_negative	1:tested_negative		0.648
7	2:tested_positive	1:tested_negative	+	0.648
8	1:tested_negative	1:tested_negative		0.648
9	1:tested_negative	1:tested_negative		0.648
10	1:tested_negative	1:tested_negative		0.648
11	1:tested_negative	1:tested_negative		0.648
12	1:tested_negative	1:tested_negative		0.648
13	1:tested_negative	1:tested_negative		0.648
14	2:tested_positive	1:tested_negative	+	0.648
15	1:tested_negative	1:tested_negative		0.648
16	1:tested_negative	1:tested_negative		0.648
17	1:tested_negative	1:tested_negative		0.648
18	2:tested_positive	1:tested_negative	+	0.648
19	2:tested_positive	1:tested_negative	+	0.648
20	1:tested_negative	1:tested_negative		0.648
21	1:tested_negative	1:tested_negative		0.648
22	2:tested_positive	1:tested_negative	+	0.648
23	1:tested_negative	1:tested_negative		0.648
24	1:tested_negative	1:tested_negative		0.648
25	1:tested_negative	1:tested_negative		0.648
26	1:tested_negative	1:tested_negative		0.648

Gunakan classifier J48 dengan metode 90% percentage split.

The screenshot shows the Weka interface with the 'Classify' tab selected. In the 'Classifier' panel, 'J48 - C 0.25 - M 2' is chosen. The 'Test options' section has 'Percentage split' selected at 90%. The 'Classifier output' panel displays a table of predictions for a test split:

inst#	actual	predicted	error	prediction
1	1:tested_negative	1:tested_negative		0.982
2	2:tested_positive	2:tested_positive		0.635
3	1:tested_negative	2:tested_positive	+	0.635
4	1:tested_negative	1:tested_negative		0.867
5	2:tested_positive	1:tested_negative	+	0.9
6	1:tested_negative	1:tested_negative		0.867
7	2:tested_positive	2:tested_positive		0.881
8	1:tested_negative	1:tested_negative		0.867
9	1:tested_negative	1:tested_negative		0.699
10	1:tested_negative	1:tested_negative		0.78
11	1:tested_negative	1:tested_negative		0.867
12	1:tested_negative	1:tested_negative		0.699
13	1:tested_negative	1:tested_negative		0.867
14	2:tested_positive	1:tested_negative	+	0.699
15	1:tested_negative	1:tested_negative		0.867
16	1:tested_negative	1:tested_negative		0.982
17	1:tested_negative	1:tested_negative		0.867
18	2:tested_positive	2:tested_positive		0.635
19	2:tested_positive	1:tested_negative	+	0.867
20	1:tested_negative	2:tested_positive	+	0.833
21	1:tested_negative	1:tested_negative		1
22	2:tested_positive	2:tested_positive		0.635
23	1:tested_negative	1:tested_negative		0.78
24	1:tested_negative	1:tested_negative		0.982
25	1:tested_negative	1:tested_negative		0.699
26	1:tested_negative	2:tested_positive	+	0.635

Gunakan classifier Logistic dengan metode 10 fold cross-validation

Classifier

Choose **Logistic -R 1.0E-8 -M -1 -num-decimal-places 4**

Test options

Use training set
 Supplied test set Set...
 Cross-validation Folds 10
 Percentage split % 90
More options...

(Nom) class Start Stop

Result list (right-click for options)

- 21:29:09 - trees.M5P
- 21:45:31 - functions.LinearRegression
- 21:55:58 - rules.OneR
- 21:56:17 - rules.OneR
- 21:56:43 - rules.OneR
- 21:59:20 - rules.OneR
- 21:59:44 - rules.OneR
- 22:01:50 - rules.OneR
- 22:02:45 - rules.OneR
- 22:12:10 - bayes.NaiveBayes
- 22:13:44 - rules.ZeroR
- 22:16:43 - trees.J48
- 22:21:24 - functions.Logistic

Classifier output

```
75 2:tested_positive 1:tested_negative + 0.653
76 2:tested_positive 1:tested_negative + 0.841

==== Stratified cross-validation ====
==== Summary ===

Correctly Classified Instances      593          77.2135 %
Incorrectly Classified Instances   175          22.7865 %
Kappa statistic                   0.4734
Mean absolute error               0.3094
Root mean squared error           0.3954
Relative absolute error            68.0818 %
Root relative squared error       82.9651 %
Total Number of Instances         768

==== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall   F-Measure  MCC    ROC Area  PRC Area  Cla
          0,880    0,429    0,793     0,880    0,834     0,480   0,832    0,892    tes
          0,571    0,120    0,718     0,571    0,636     0,480   0,832    0,715    tes
Weighted Avg.   0,772    0,321    0,767     0,772    0,765     0,480   0,832    0,831

==== Confusion Matrix ===

      a   b  <-- classified as
440  60 |  a = tested_negative
115 153 |  b = tested_positive
```

Status OK Log 

4.5 Support Vector Machines

Gunakan parameter pada SMO (Sequential Minimal Optimization) untuk mengimplementasi support vector machines.

weka.classifiers.functions.SMO

About

Implements John Platt's sequential minimal optimization algorithm for training a support vector classifier.

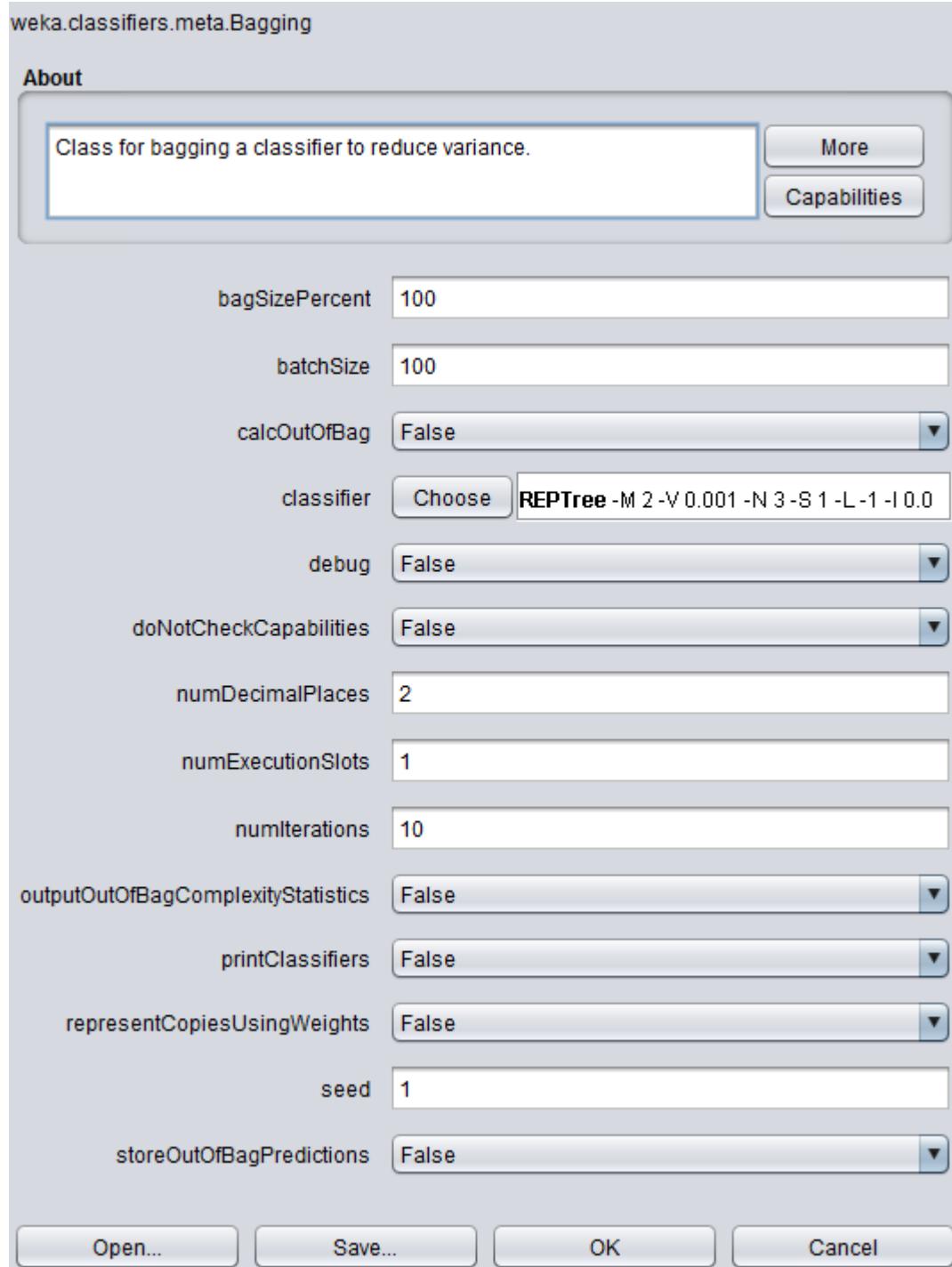
More Capabilities

batchSize	100
buildCalibrationModels	False
c	1.0
calibrator	Choose Logistic -R 1.0E-8 -M -1 -num-decimal-places 4
checksTurnedOff	False
debug	False
doNotCheckCapabilities	False
epsilon	1.0E-12
filterType	Normalize training data
kernel	Choose PolyKernel -E 1.0 -C 250007
numDecimalPlaces	2
numFolds	-1
randomSeed	1
toleranceParameter	0.001

Open... Save... OK Cancel

4.6 Ensemble Learning

Gunakan parameter pada classifier Bagging yang merupakan training set di-sample sesuai batchSize dengan replacement yang kemudian dilakukan iterasi sejumlah numIterations dengan classifier tertentu.



Gunakan parameter pada classifier Random Forest yang merupakan kebalikan dari Bagging yang merandom training set, tetapi me-random classifier.

weka.classifiers.trees.RandomForest

About

Class for constructing a forest of random trees.

More Capabilities

bagSizePercent	100
batchSize	100
breakTiesRandomly	False
calcOutOfBag	False
computeAttributeImportance	False
debug	False
doNotCheckCapabilities	False
maxDepth	0
numDecimalPlaces	2
numExecutionSlots	1
numFeatures	0
numIterations	100
outputOutOfBagComplexityStatistics	False
printClassifiers	False
seed	1

Open... Save... OK Cancel

Gunakan parameter classifier AdaBoost M1 yang merupakan metode belajar Boost dimana model akan diperbaiki sejumlah numIterations.

weka.classifiers.meta.AdaBoostM1

About

Class for boosting a nominal class classifier using the Adaboost M1 method.

More

Capabilities

batchSize 100

classifier Choose DecisionStump

debug False

doNotCheckCapabilities False

numDecimalPlaces 2

numIterations 10

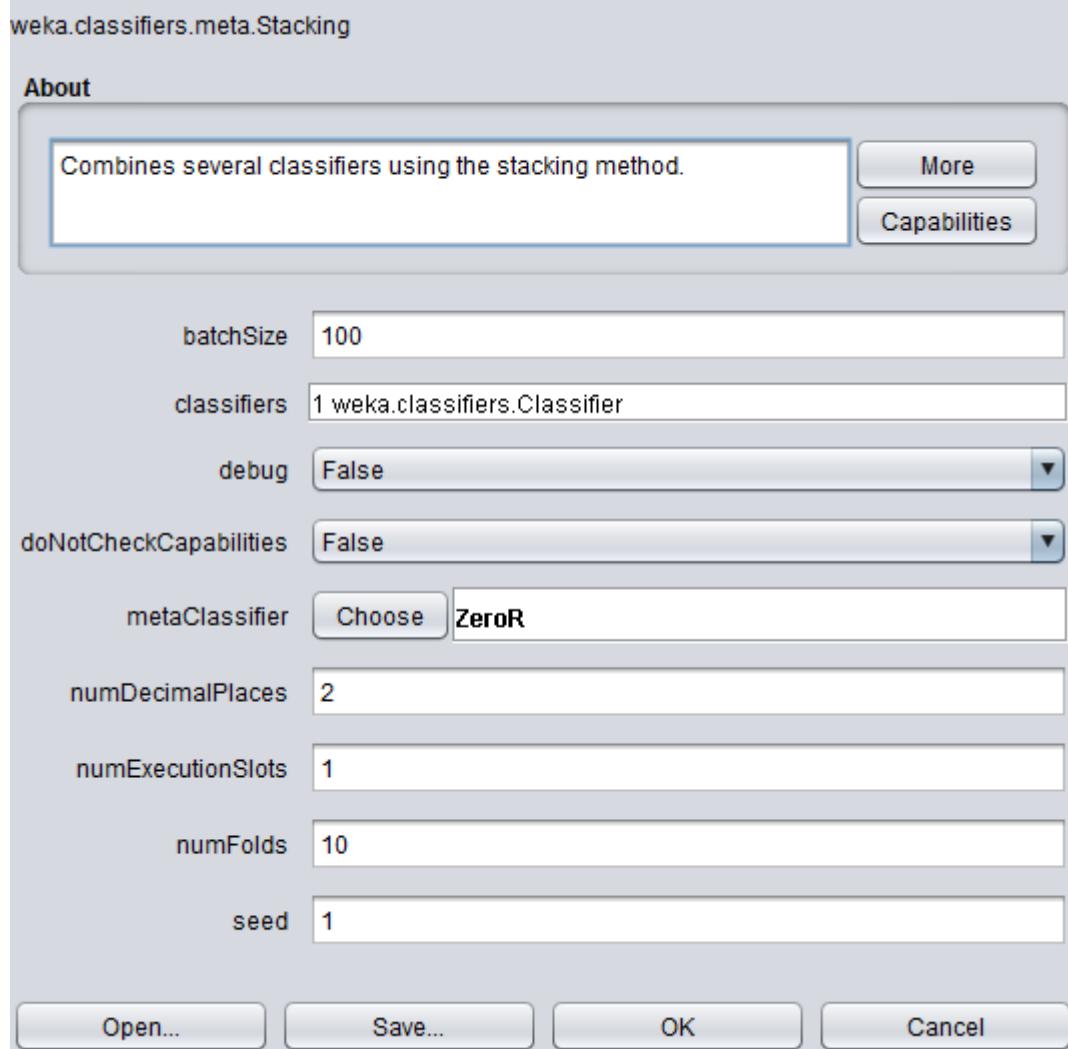
resume False

seed 1

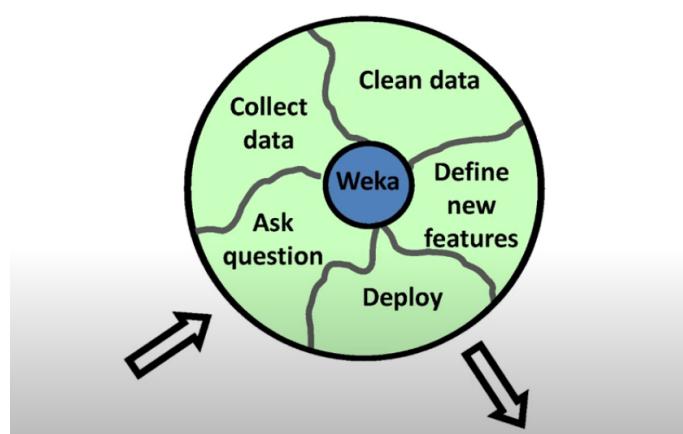
useResampling False

weightThreshold 100

Gunakan parameter classifier Stacking dimana output (model) dari classifier (level-0) akan menjadi input pada metaClassifier (level-1).



5.1 The Data Mining Process



Proses yang dilalui dalam Data Mining secara garis besar ada 5, yaitu:

1. *Ask a question*: Apa yang kita ingin ketahui dari data
2. *Collect data*: mengumpulkan beberapa dari beberapa sumber untuk memperkaya informasi yang ingin didapatkan,

3. *Clean data*: Melakukan pembersihan terhadap data yang dimiliki karena data sebenarnya sangatlah kotor dan tidak beraturan.
4. *Define a new features*: Menentukan fitur baru adalah kunci dalam data mining. Fitur tersebut disebut *feature engineering*.
5. *Deploy the result*: Setelah berhasil melalui semua tahapan yang ada, model prediksi atau informasi yang didapatkan di-deploy agar bisa digunakan.

Dalam proses data mining, WEKA hanya membantu dalam bagian kecil saja, terutama bagian teknikal.

5.2 Pitfalls and Pratfalls

Dalam proses data mining, sering kali ditemui *pitfalls* dan *pratfalls*. *Pitfalls* adalah kesulitan atau bahaya yang tidak terdeteksi dan *pratfalls* adalah aksi yang kurang baik atau ceroboh. Dalam data mining, sangat mudah sekali dilakukan kecurangan pada pemrosesan data, baik secara sadar maupun tidak sadar. Untuk melakukan tes yang *reliable*, gunakanlah data yang benar-benar baru dan belum dipakai sebelumnya.

Dalam proses pembuatan model, seringkali terjadi overfitting. Hal yang dapat dilakukan untuk menghindari hal tersebut adalah dengan tidak memakai training data dalam melakukan testing dan melakukan evaluasi terhadap model yang telah dibuat.

5.3 Data Mining and Ethics

Dalam melakukan data mining ada beberapa hukum privasi yang diperhatikan, di setiap negara/benua memiliki aturan yang tidak semuanya sama, namun secara garis besar memiliki makna yang sama.

Berikut contohnya hukum privasi yang ada di Eropa.

Information privacy laws (in Europe, but not US)

- ❖ A purpose must be stated for any personal information collected
- ❖ Such information must not be disclosed to others without consent
- ❖ Records kept on individuals must be accurate and up to date
- ❖ To ensure accuracy, individuals should be able to review data about themselves
- ❖ Data must be deleted when it is no longer needed for the stated purpose
- ❖ Personal information must not be transmitted to locations where equivalent data protection cannot be assured
- ❖ Some data is too sensitive to be collected, except in extreme circumstances

5.4 Summary

Dari video-video yang sudah dipelajari sebelumnya, dapat disimpulkan bahwa banyak teknik yang bisa dilakukan dalam melakukan data mining. Metode-metode yang ada di dunia tidak satupun cocok ke setiap kondisi data, sehingga perlu dievaluasi setiap algoritma yang berbeda dan performa yang ditunjukkan, dari algoritma yang telah dipakai.