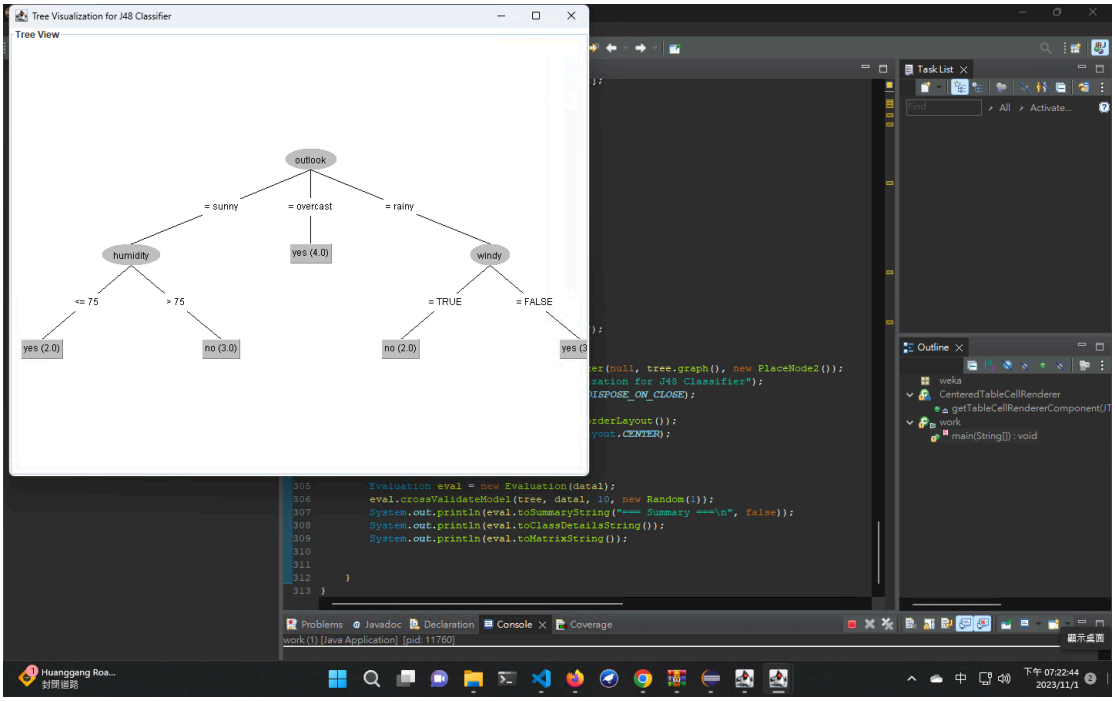
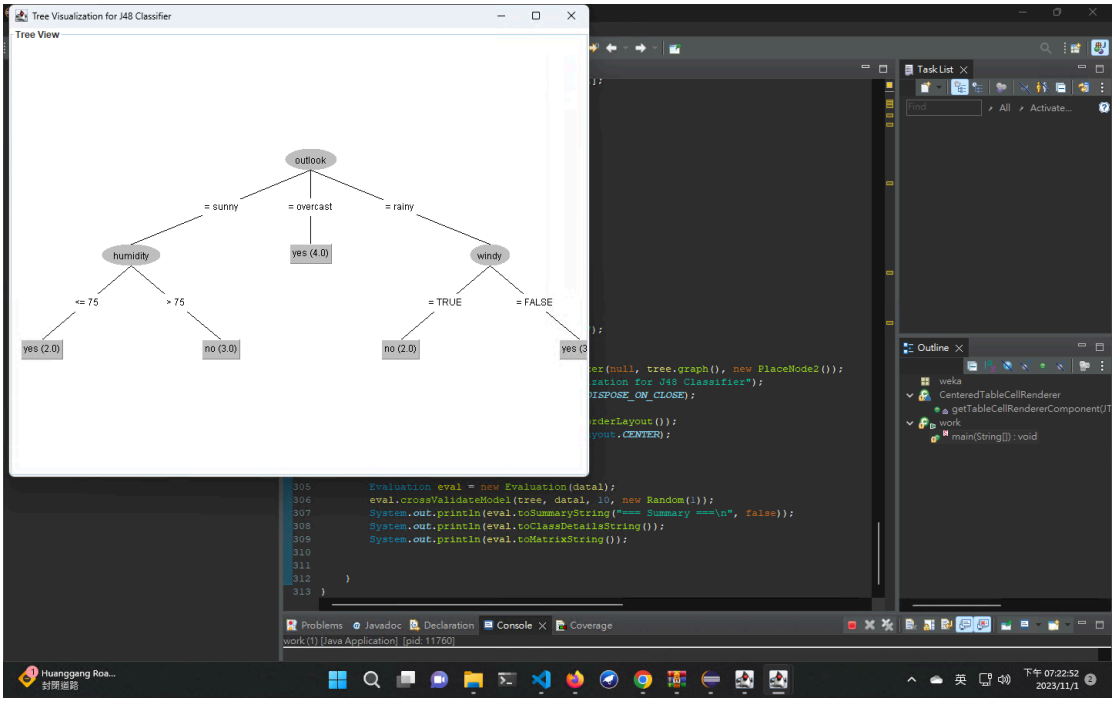
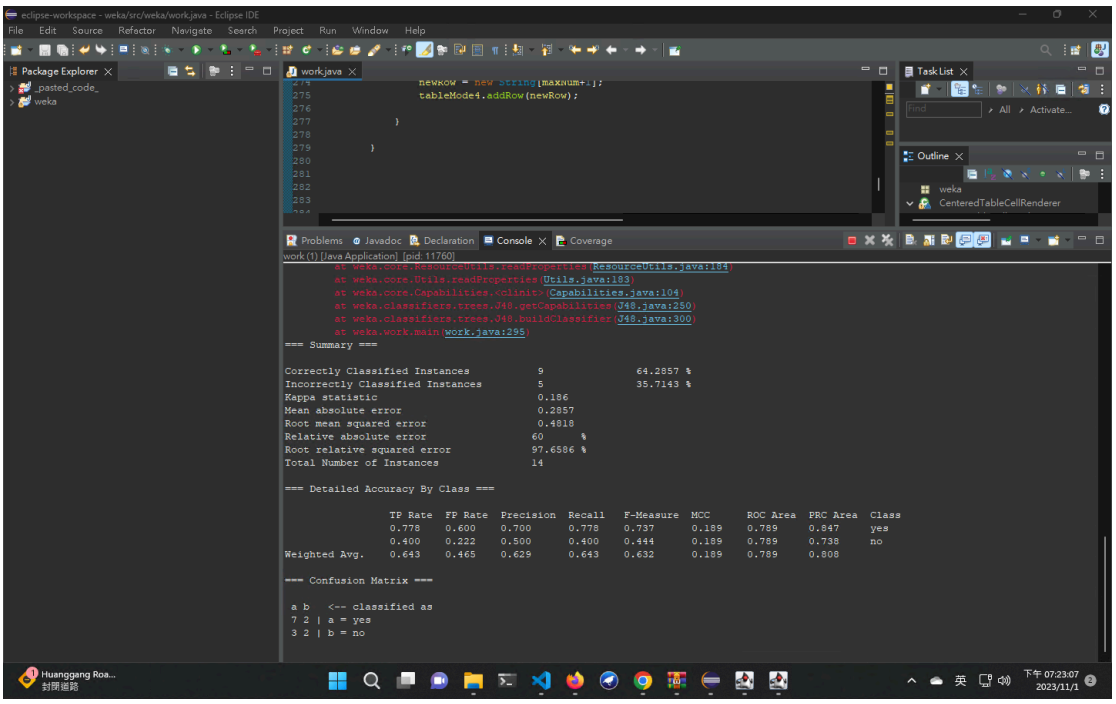
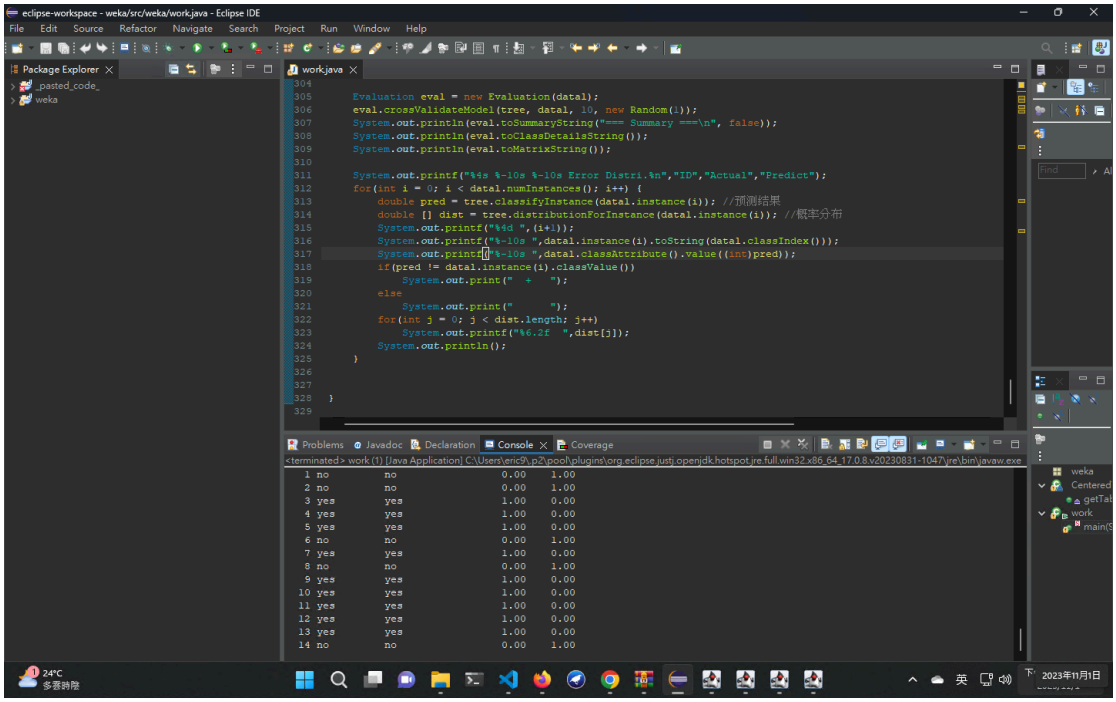
建立決策數與訓練資料


Tree Visualization for J48 Classifier
Tree View
outlook
= sunny
= overcast
= rainy
humidity
yes (4.0)
windy
<= 75
> 75
= TRUE
= FALSE
yes (2.0)
no (3.0)
no (2.0)
Task List
Outline
Problems
Javadoc
Declaration
Console
Coverage

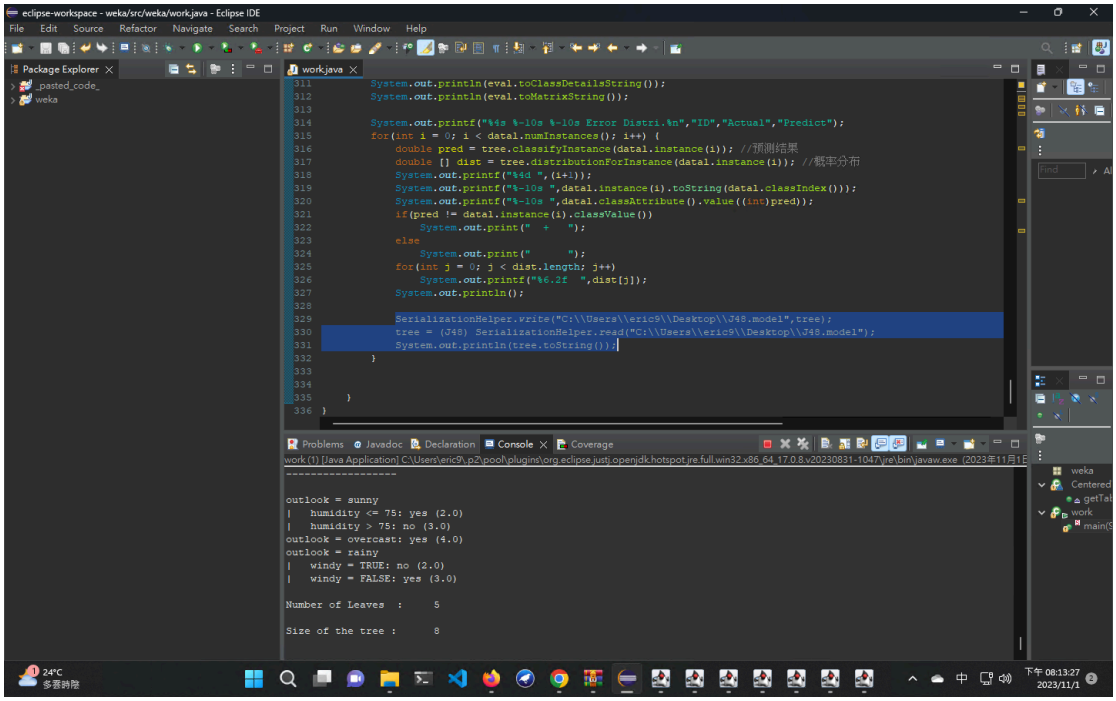

Tree Visualization for J48 Classifier
Tree View
outlook
= sunny
= overcast
= rainy
humidity
yes (4.0)
windy
<= 75
> 75
= TRUE
= FALSE
yes (2.0)
no (3.0)
no (2.0)
Task List
Outline
Problems
Javadoc
Declaration
Console
Coverage


eclipse-workspace - weka/src/weka/work.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
_pasted_code_
weka
work.java
Task List
Outline
Problems Javadoc Declaration Console Coverage
work (1) [Java Application] [pid: 11760]
at weka.core.ResourceUtils.readProperties(ResourceUtils.java:184)
at weka.core.Utils.readProperties(Utils.java:183)
at weka.core.Capabilities.<clinit>(Capabilities.java:104)
at weka.classifiers.trees.J48.getCapabilities(J48.java:250)
at weka.classifiers.trees.J48.buildClassifier(J48.java:300)
at weka.work.main(work.java:295)
=== Summary ===
Correctly Classified Instances 9 64.2857 %
Incorrectly Classified Instances 5 35.7143 %
Kappa statistic 0.186
Mean absolute error 0.2857
Root mean squared error 0.4818
Relative absolute error 60 %
Root relative squared error 97.6586 %
Total Number of Instances 14
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.778 0.600 0.700 0.778 0.737 0.189 0.789 0.847 yes
0.400 0.222 0.500 0.400 0.444 0.189 0.789 0.738 no
Weighted Avg. 0.643 0.465 0.629 0.643 0.632 0.189 0.789 0.808
=== Confusion Matrix ===
a b <-- classified as
7 2 | a = yes
3 2 | b = no

印出每筆預測


eclipse-workspace - weka/src/weka/work.java - Eclipse IDE
Evaluation eval = new Evaluation(datal);
eval.crossValidateModel(tree, datal, 10, new Random(1));
System.out.println(eval.toSummaryString("=== Summary ===\n", false));
System.out.println(eval.toClassDetailsString());
System.out.println(eval.toMatrixString());
System.out.printf("%4s %-10s %-10s Error Distri.%n","ID","Actual","Predict");
for(int i = 0; i < datal.numInstances(); i++) {
double pred = tree.classifyInstance(datal.instance(i)); //預測結果
double [] dist = tree.distributionForInstance(datal.instance(i)); //概率分布
System.out.printf("%4d ",(i+1));
System.out.printf("%-10s ",datal.instance(i).toString(datal.classIndex()));
System.out.printf("%-10s ",datal.classAttribute().value((int)pred));
if(pred != datal.instance(i).classValue())
System.out.print(" + ");
else
System.out.print(" ");
for(int j = 0; j < dist.length; j++)
System.out.printf("%6.2f ",dist[j]);
System.out.println();
}
}
1 no no 0.00 1.00
2 no no 0.00 1.00
3 yes yes 1.00 0.00
4 yes yes 1.00 0.00
5 yes yes 1.00 0.00
6 no no 0.00 1.00
7 yes yes 1.00 0.00
8 no no 0.00 1.00
9 yes yes 1.00 0.00
10 yes yes 1.00 0.00
11 yes yes 1.00 0.00
12 yes yes 1.00 0.00
13 yes yes 1.00 0.00
14 no no 0.00 1.00

儲存模型



NaiveBayes;