

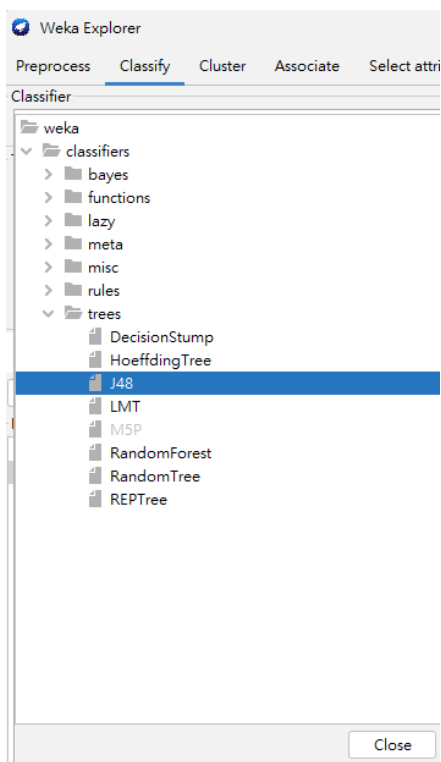
決策樹在 Weka 的範例

1. 藥物決策樹

開啟 Weka 選擇 Explorer



先放入資料選擇 Classify > tree > J48 > start



Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Classifier: Choose RandomTree -K 0 -M 1.0 -V 0.001 -S 1

Test options

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

More options...

(Nom) medicine 以藥物做分類

Start Stop

Result list (right-click for options)

- 14:46:42 - trees.J48
- 14:46:50 - trees.REPTree

Classifier output

Number of Leaves : 4

Size of the tree : 6

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	12	100	%
Incorrectly Classified Instances	0	0	%
Kappa statistic	1		
Mean absolute error	0		
Root mean squared error	0		
Relative absolute error	0	%	
Root relative squared error	0	%	
Total Number of Instances	12		

12 筆資料 12 筆都正確
沒有錯誤筆數

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	A
1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	B
Weighted Avg.	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	

=== Confusion Matrix ===

```

a b  <-- classified as
6 0 | a = A
0 6 | b = B

```

Status OK Log x0

執行結果按右鍵，選擇 Visualize tree，瀏覽分類樹結果

Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visual

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) medicine

Start Stop

Result list (right-click for options)

- 15:44:45 - trees.J48

Classifier output

Number of Leaves : 4

Size of the tree : 6

Time taken to build model:

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances

Incorrectly Classified Instances

Kappa statistic

View in main window

View in separate window

Save result buffer

Delete result buffer(s)

Load model

Save model

Re-evaluate model on current test set

Re-apply this model's configuration

Visualize classifier errors

Visualize tree

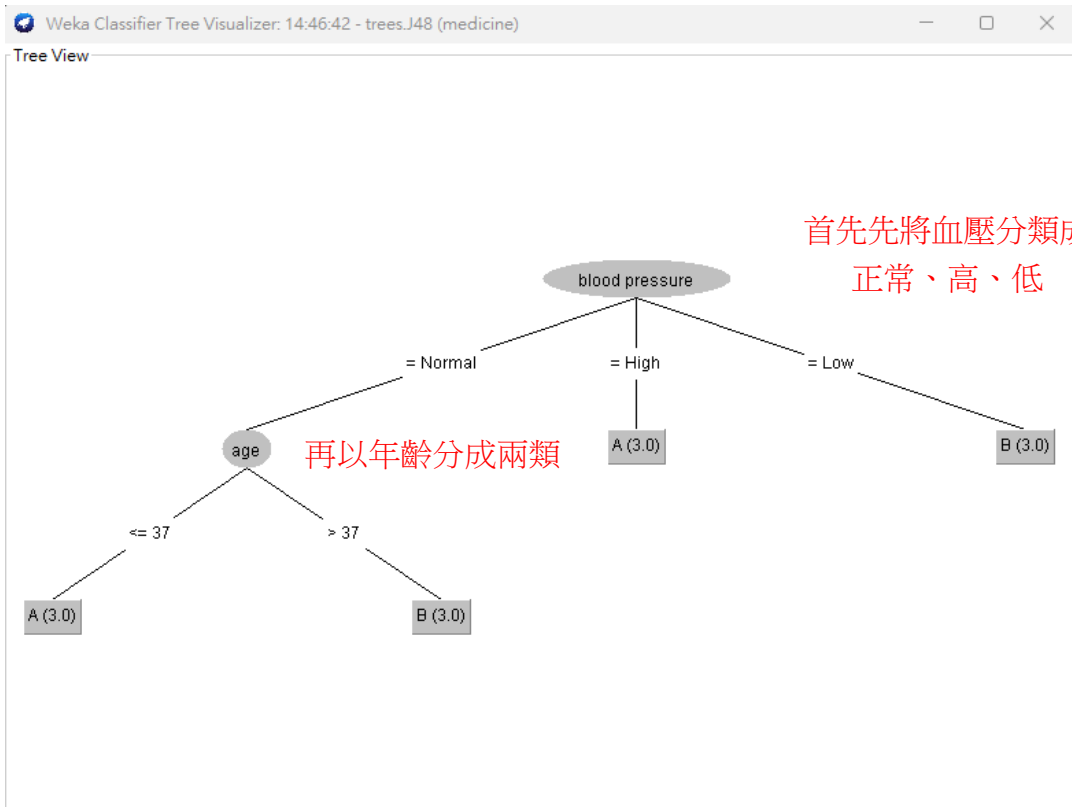
Visualize margin curve

Visualize threshold curve >

Cost/Benefit analysis >

Visualize cost curve >

分類數



2. 鳶尾花決策樹

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)

- 15:08:25 - rules.ZeroR
- 15:08:39 - trees.J48

Classifier output

Size of the tree : 9

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	144	96	% 150 筆資料 144 筆正確
Incorrectly Classified Instances	6	4	% 150 筆資料 6 筆錯誤
Kappa statistic	0.94		
Mean absolute error	0.035		
Root mean squared error	0.1586		
Relative absolute error	7.8705 %		
Root relative squared error	33.6353 %		
Total Number of Instances	150		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.980	0.000	1.000	0.980	0.990	0.985	0.990	0.987	Iris-seto
	0.940	0.030	0.940	0.940	0.940	0.910	0.952	0.880	Iris-vers
	0.960	0.030	0.941	0.960	0.950	0.925	0.961	0.905	Iris-virg
Weighted Avg.	0.960	0.020	0.960	0.960	0.960	0.940	0.968	0.924	

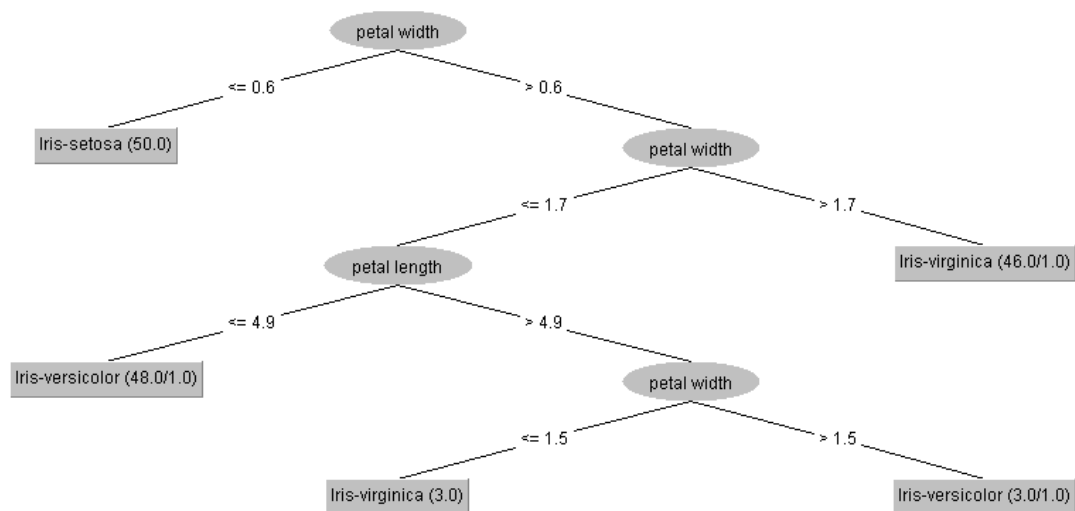
=== Confusion Matrix ===

```

a b c <-- classified as
49 1 0 | a = Iris-setosa
0 47 3 | b = Iris-versicolor
0 2 48 | c = Iris-virginica
  
```

Status: OK

Log x0



```

▶ from sklearn import tree
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

# 讀入鳶尾花資料
iris = load_iris()
iris_data = iris.data
iris_label = iris.target

# 切分訓練與測試資料
train_data, test_data, train_label, test_label = train_test_split(iris_data, iris_label, test_size = 0.3)

# 建立分類器
clf = tree.DecisionTreeClassifier()
iris_clf = clf.fit(train_data, train_label)

print("訓練資料正確率:", clf.score(train_data, train_label)) # 訓練資料正確率
print("測試資料正確率:", clf.score(test_data, test_label)) # 測試資料正確率

predict=clf.predict(test_data) # 預測
print("標準答案:", test_label) # 標準答案
print("預測答案:", predict) # 預測答案

```

```

⇒ 訓練資料正確率: 1.0
測試資料正確率: 0.9777777777777777
標準答案: [0 0 2 1 2 2 2 1 0 2 2 0 1 1 2 0 1 0 2 1 2 0 0 1 1 0 1 2 1 0 2 1 2 0 1 1 1
1 2 2 0 0 0 0 0]
預測答案: [0 0 2 1 2 2 2 1 0 2 2 0 1 1 2 0 1 0 2 1 2 0 0 1 1 0 1 2 1 0 2 1 2 0 1 1 2
1 2 2 0 0 0 0 0]

```

```

[2] import graphviz
dot_data = tree.export_graphviz(clf, out_file=None)
graph = graphviz.Source(dot_data)
graph.render("iris") #將結果存成pdf檔案

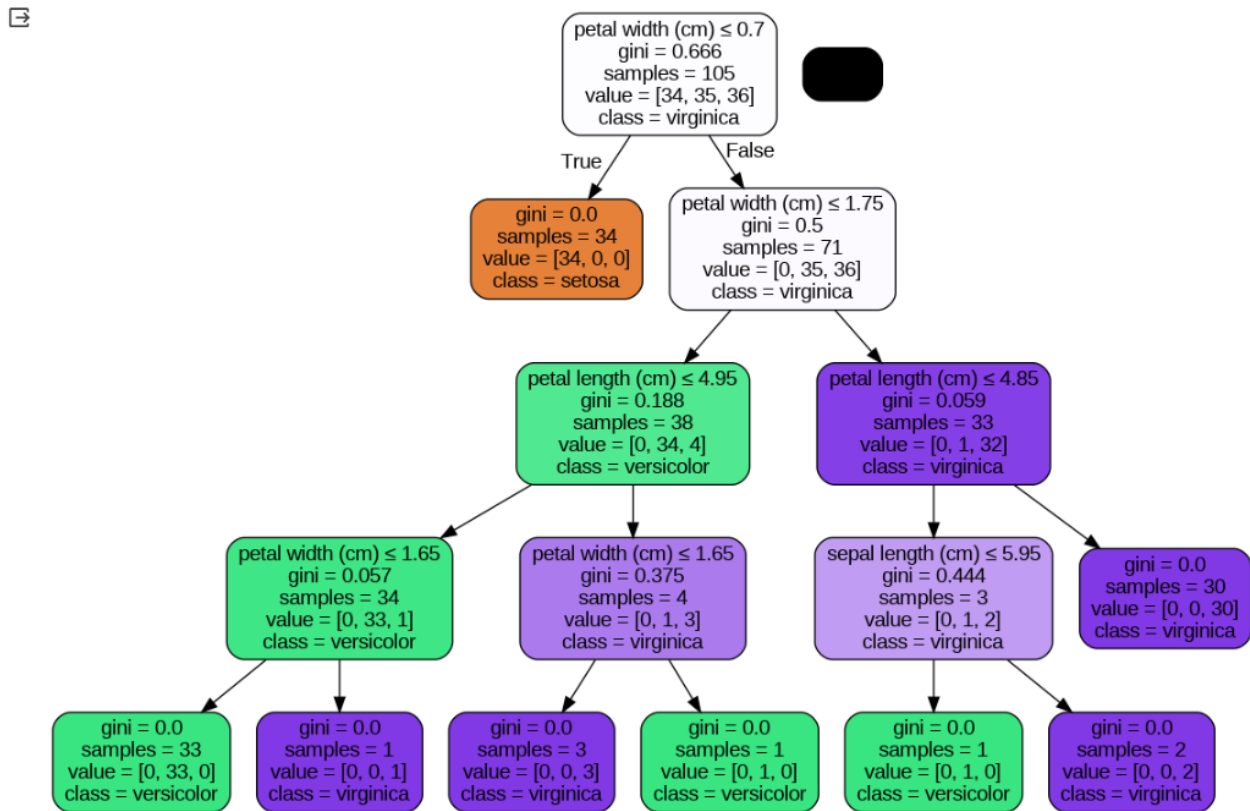
```

'iris.pdf'

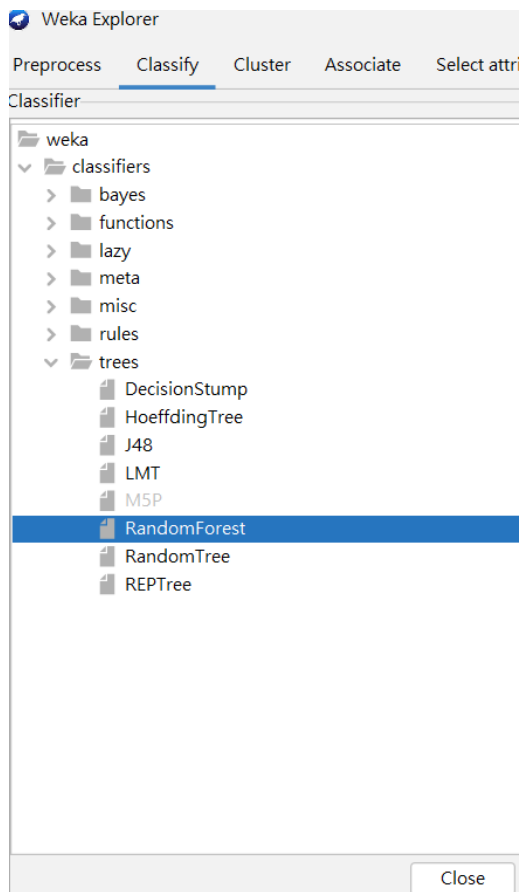
```

import pydotplus
from IPython.display import Image
dot_data = trees.export_graphviz(clf, out_file=None,
                                feature_names=iris.feature_names,
                                class_names=iris.target_names,
                                filled=True, rounded=True,
                                special_characters=True)
graph = pydotplus.graph_from_dot_data(dot_data)
Image(graph.create_png()) #將結果用圖形化呈現

```



使用 Random Forest 測試資料



完全使用測試資料 資料會 100%正確

Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Classifier: Choose RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

- ☒ Use training set
- ☐ Supplied test set
- ☐ Cross-validation Folds 10
- ☐ Percentage split % 66

More options...

(Nom) class

Start Stop

Result list (right-click for options)

- 14:05:07 - trees.J48
- 14:30:45 - trees.RandomForest

Classifier output

Time taken to build model: 0.05 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances	150	100	%
Incorrectly Classified Instances	0	0	%
Kappa statistic	1		
Mean absolute error	0.0156		
Root mean squared error	0.0628		
Relative absolute error	3.52	%	
Root relative squared error	13.3147	%	
Total Number of Instances	150		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Iris-seto
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Iris-vers
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Iris-virg
Weighted Avg.	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	

=== Confusion Matrix ===

```
a b c <-- classified as
50 0 0 | a = Iris-setosa
0 50 0 | b = Iris-versicolor
0 0 50 | c = Iris-virginica
```

Status OK Log x0

使用 80%的測試資料 會有 1 比錯誤資料

Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Classifier: Choose RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options

- ☐ Use training set
- ☐ Supplied test set
- ☐ Cross-validation Folds 10
- ☒ Percentage split % 80

More options...

(Nom) class

Start Stop

Result list (right-click for options)

- 14:05:07 - trees.J48
- 14:30:45 - trees.RandomForest
- 14:32:45 - trees.RandomForest

Classifier output

Time taken to build model: 0.03 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances	29	96.6667	%
Incorrectly Classified Instances	1	3.3333	%
Kappa statistic	0.9497		
Mean absolute error	0.0304		
Root mean squared error	0.1132		
Relative absolute error	6.8444	%	
Root relative squared error	24.001	%	
Total Number of Instances	30		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Iris-seto
	1.000	0.050	0.909	1.000	0.952	0.929	1.000	1.000	Iris-vers
	0.889	0.000	1.000	0.889	0.941	0.921	1.000	1.000	Iris-virg
Weighted Avg.	0.967	0.017	0.970	0.967	0.966	0.953	1.000	1.000	

=== Confusion Matrix ===

```
a b c <-- classified as
11 0 0 | a = Iris-setosa
0 10 0 | b = Iris-versicolor
0 1 8 | c = Iris-virginica
```

Status OK Log x0

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

# 讀入鳶尾花資料
iris = load_iris()
iris_data = iris.data
iris_label = iris.target

# 切分訓練與測試資料
train_data, test_data, train_label, test_label = train_test_split(iris_data, iris_label, test_size = 0.3)

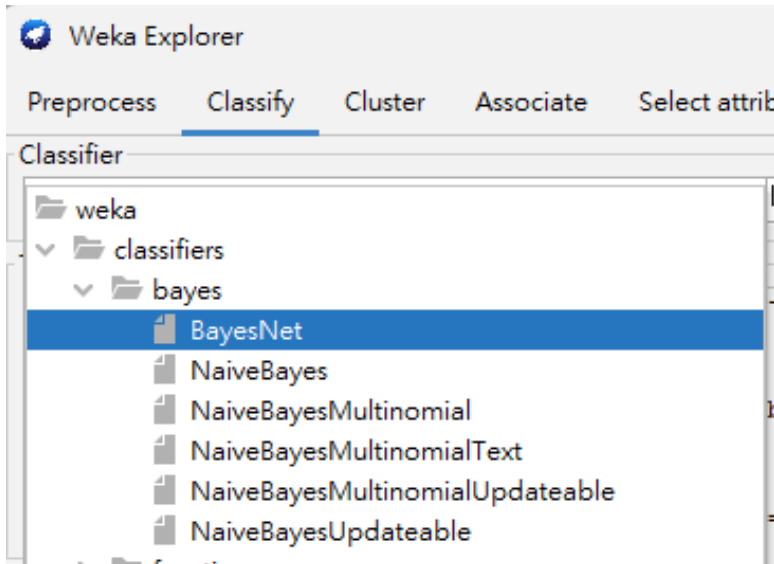
# 建立分類器
clf = RandomForestClassifier(n_estimators=30, max_depth=4)
iris_clf = clf.fit(train_data, train_label)
print("訓練資料正確率:", clf.score(train_data, train_label)) # 訓練資料正確率
print("測試資料正確率:", clf.score(test_data, test_label)) # 測試資料正確率

predict=clf.predict(test_data) # 預測
print("標準答案:", test_label) # 標準答案
print("預測答案:", predict) # 預測答案
```

```
➡ 訓練資料正確率: 1.0
測試資料正確率: 0.9333333333333333
標準答案: [2 1 2 2 1 2 1 0 0 0 0 1 0 2 0 0 2 0 0 2 2 2 0 2 1 0 1 2 1 1 2 1 2 1 0 2 1
 1 0 0 1 2 0 1 1]
預測答案: [2 1 2 2 1 1 2 0 0 0 0 2 0 2 0 0 2 0 0 2 2 2 0 2 1 0 1 2 1 1 2 1 2 1 0 2 1
 1 0 0 1 2 0 1 1]
```

Naive Bayesian Classifier 貝氏分類器

1. 開啟開啟 Iris.csv
2. 選擇 Classify -> bayes BayesNet



3. 測試出結果正確率為 92.6667%

The screenshot shows the Weka Explorer interface with the 'Classifier' tab selected. The 'Classifier' list shows 'BayesNet' selected. The 'Test options' section shows 'Cross-validation' selected with 'Folds' set to 10. The 'Classifier output' section displays the results of the cross-validation, including the accuracy of 92.6667%.

Classifier output

LogScore AIC: -497.39347511858665

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	139	92.6667 %
Incorrectly Classified Instances	11	7.3333 %
Kappa statistic	0.89	
Mean absolute error	0.0454	
Root mean squared error	0.1828	
Relative absolute error	10.2111 %	
Root relative squared error	38.7793 %	
Total Number of Instances	150	

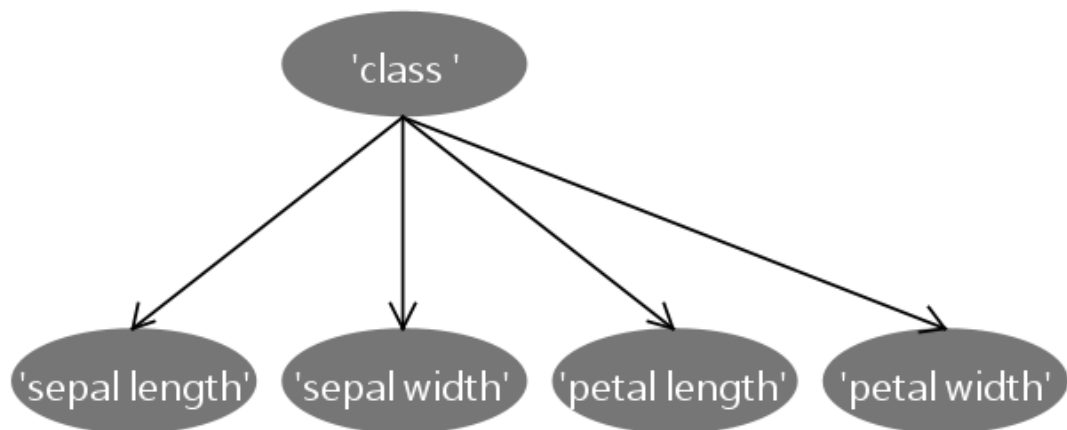
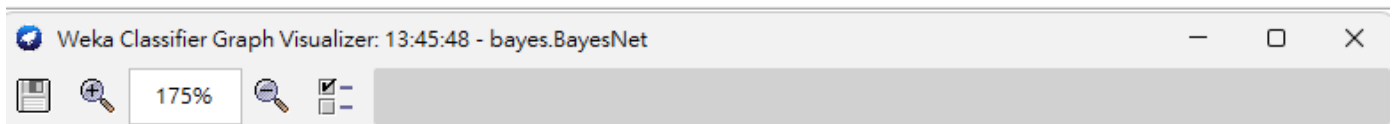
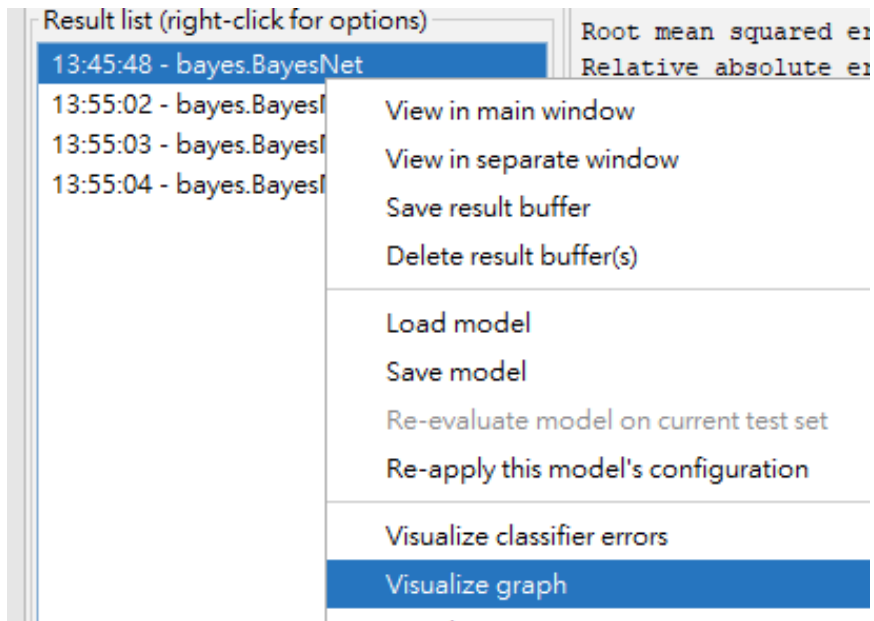
=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Iris-seto
	0.880	0.050	0.898	0.880	0.889	0.834	0.971	0.906	Iris-vers
	0.900	0.060	0.882	0.900	0.891	0.836	0.970	0.919	Iris-virg
Weighted Avg.	0.927	0.037	0.927	0.927	0.927	0.890	0.980	0.942	

=== Confusion Matrix ===

```
a b c <-- classified as
50 0 0 | a = Iris-setosa
0 44 6 | b = Iris-versicolor
0 5 45 | c = Iris-virginica
```


4. 查看圖形結果 執行結果右鍵 -> Visualize graph



```

from sklearn import datasets
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split

iris=datasets.load_iris()
x=iris.data
y=iris.target
train_data, test_data, train_label, test_label = train_test_split(x, y, test_size=0.3, random_state=0)
clf=GaussianNB()
clf.fit(train_data, train_label)
print("訓練資料正確率:", clf.score(train_data, train_label)) # 訓練資料正確率
print("測試資料正確率:", clf.score(test_data, test_label)) # 測試資料正確率
predict=clf.predict(test_data)
print("標準答案:", test_label) # 標準答案
print("預測答案:", predict) # 預測答案

```

```

➡ 訓練資料正確率: 0.9428571428571428
測試資料正確率: 1.0
標準答案: [2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0
1 1 1 2 0 2 0 0]
預測答案: [2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0
1 1 1 2 0 2 0 0]

```