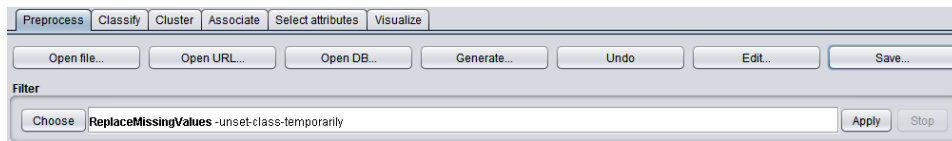


2019 ECT 作業八

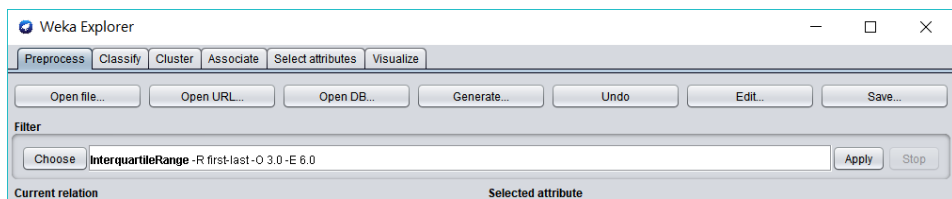
一、 利用Weka 對DeerHunter.arff 進行前處理，依序完成以下步驟及問題：

(a). Replace Missing Value, 需列出補上的值為何(10%)

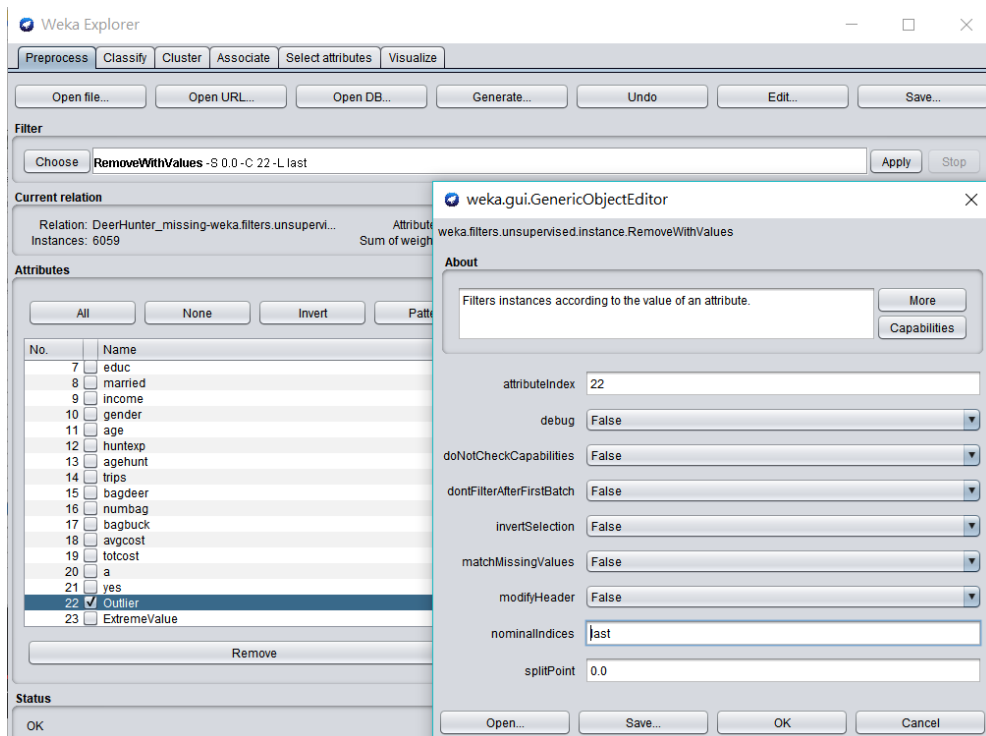


- 在Preprocess的Filter選取「weka/filters/unsupervised/attribute/ReplaceMissingValues」
- 點選「Apply」，可將MissingValue值變更為MeanValue

(b). Outlier Detection& Remove (10%)



- 在Preprocess的Filter選取「weka/filters/unsupervised/attribute/InterquartileRange」
- 點選「Apply」，會多出兩個Attribute(Outlier與ExtremeValue)

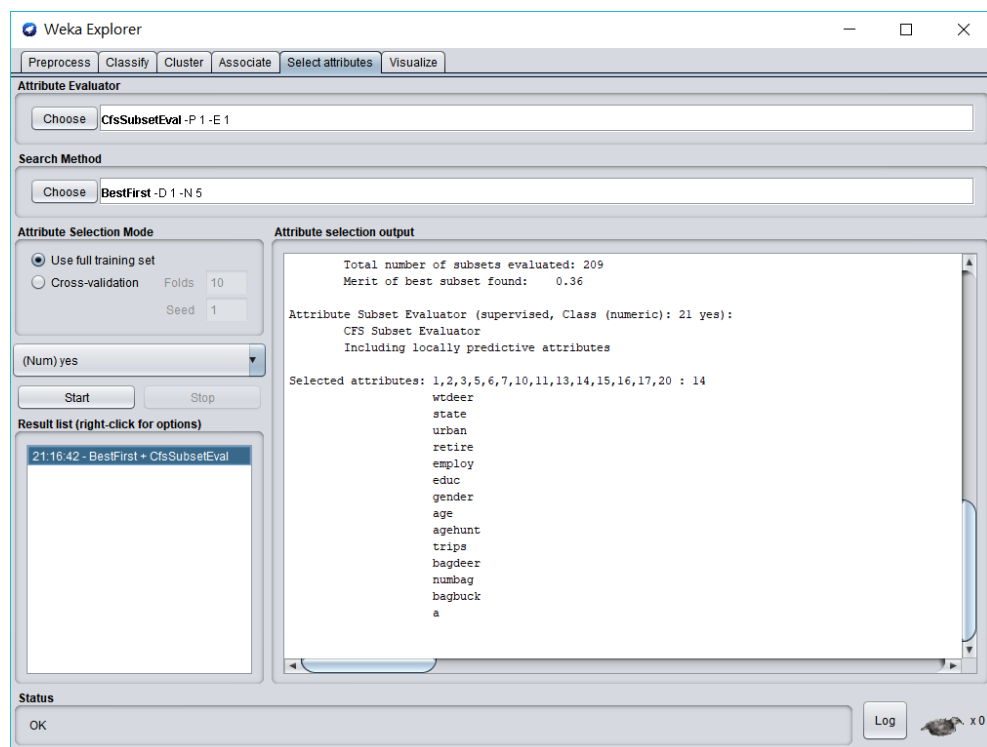


- iii. 在Preprocess的Filter選取「weka/filters/unsupervised/instance/RemoveWithValues」
- iv. 在參數調整內，將「attributeIndex」設為「22」(Outlier的index編號)，「nominalIndices」設為「last」(Outlier為yes的instance)，並點選「OK」。

(此步驟的目的在於判別若為Outlier的instance，則刪除)

- v. 點選「Apply」，完成。

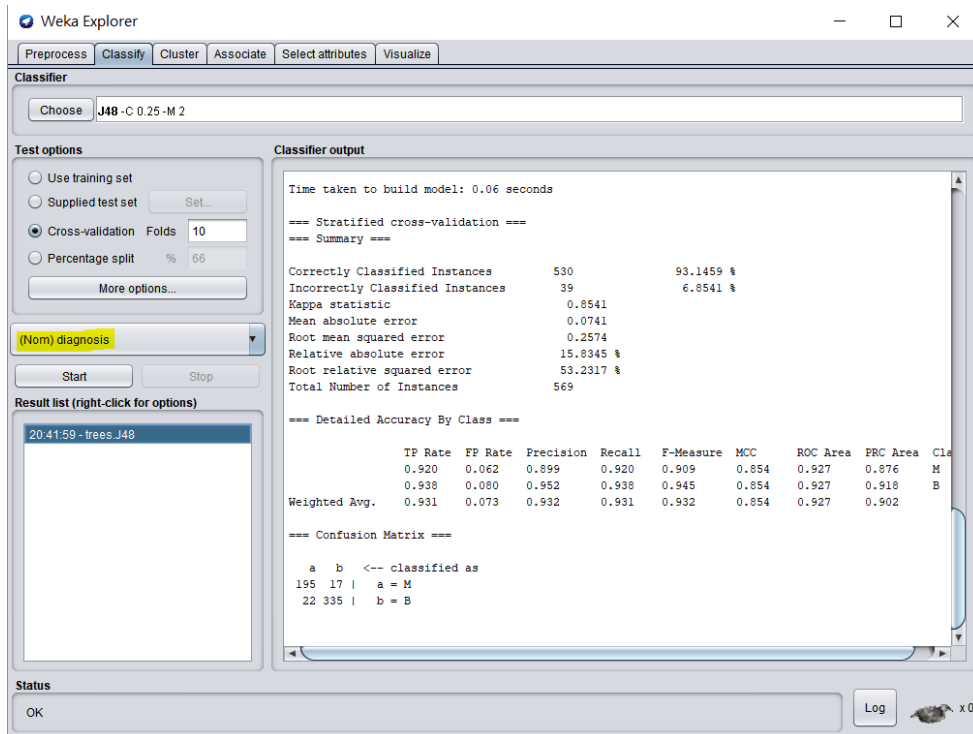
(c). Attribute Selection, 請篩選出適合的屬性(10%)



- i. 在Select attributes的Attribute Evaluator選取「weka/attributeSelection/CfsSubsetEval」、在Search Method選取「weka/attributeSelection/BestFirst」
- ii. 將屬性設為「(Num)yes」
- iii. 點選「Start」
- iv. 可由「Attribute selection output」得適合的屬性為「wtdeer」、「state」、「urban」、「retire」、「employ」、「educ」、「gender」、「age」、「agehunt」、「trips」、「bagdeer」、「numbag」、「bagbuck」、「a」，共14個

二、 請用weka對BreastCancer.csv對目標diagnosis進行Ensemble learning並與未使用的結果進行比較(請列出重要過程及適當說明)：

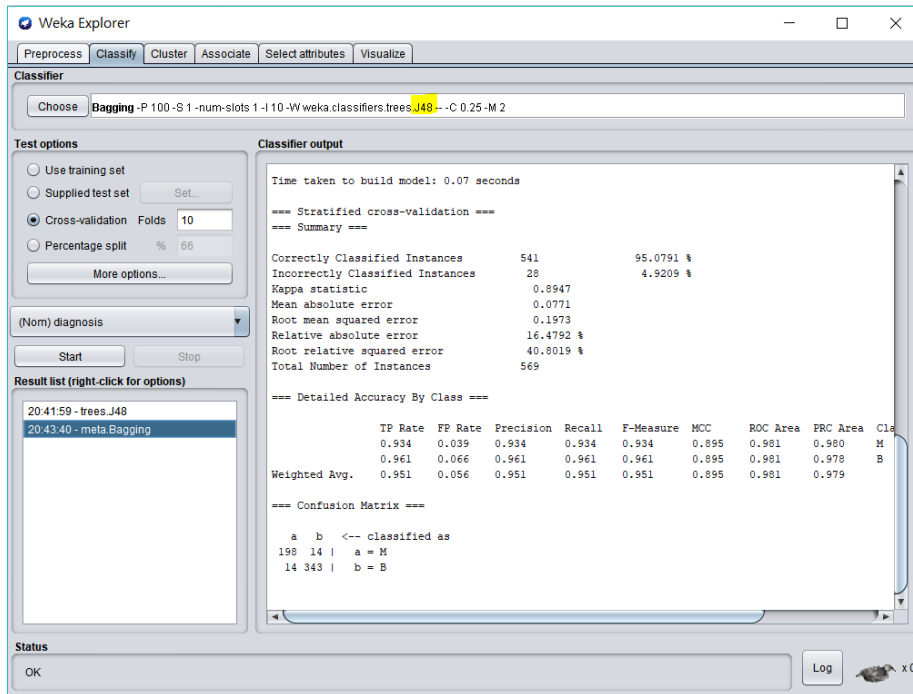
- (a). 以10 Folds cross-validation進行J48分類(5%)



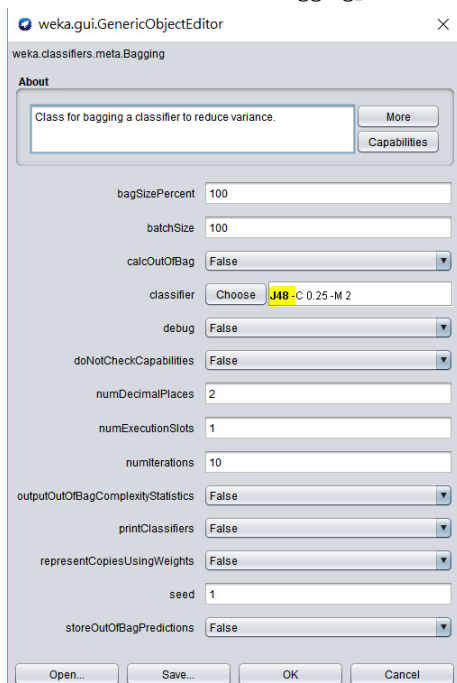
已註解 [柏丞1]:

- i. 在Classifier中選取「J48」
- ii. Test options 使用 Cross-validation, 並設定Folds為10
- iii. 選取屬性「(Nom)diagnosis」
- iv. 點選「Start」

(b). 以10 Folds cross-validation進行Bagging分類並選擇J48 classifier進行分類(10%)

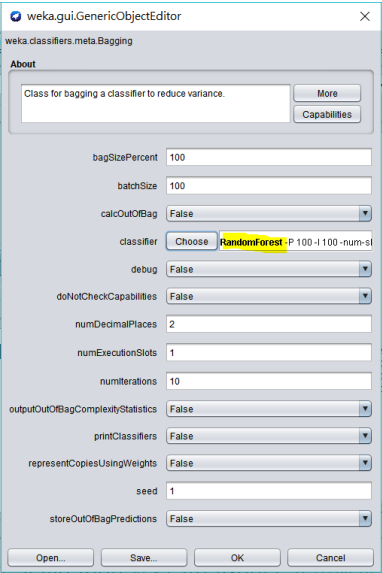
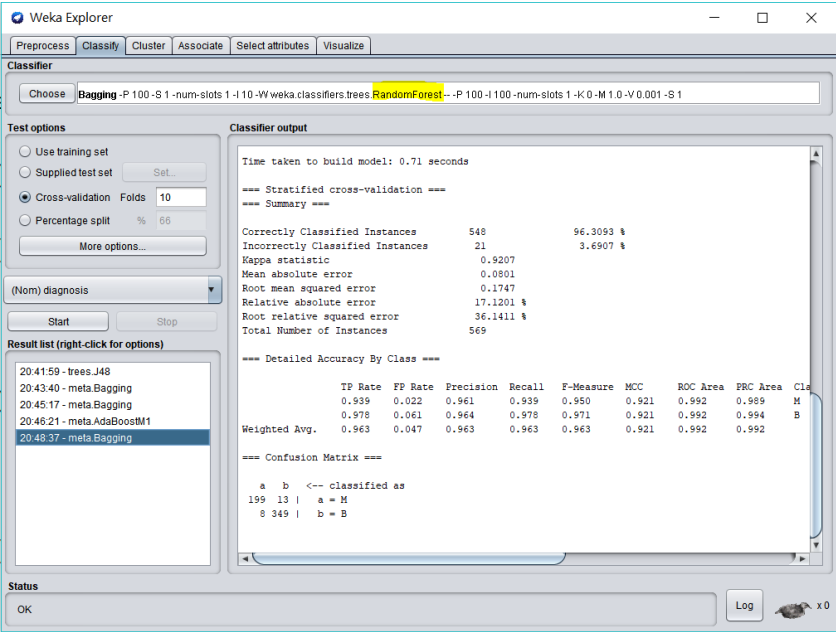


i. 在Classifier中選取「Bagging」



- ii. 在參數調整的地方，將classifier設為「J48」
- iii. Test options 使用 Cross-validation，並設定Folds為10
- iv. 選取屬性「(Nom)diagnosis」
- v. 點選「Start」

(c). 以10 Folds cross-validation進行Bagging分類並選擇Randomforest進行分類(10%)



- i. 在參數調整的地方，將classifier設為「RandomForest」
- ii. Test options 使用 Cross-validation，並設定Folds為10
- iii. 選取屬性「(Nom)diagnosis」
- iv. 點選「Start」

(d). 以10 Folds cross-validation進行AdaBoost分類並選擇DecisionStump進行分類(10%)

Weka Explorer

Preprocess

Classify

Cluster

Associate

Select attributes

Visualize

Classifier

Choose

AdaBoostM1 - P 100 - S 1 - I 10 - W weka.classifiers.trees.DecisionStump

Test options

Use training set

Supplied test set

Cross-validation

Folds

10

Percentage split

%

68

More options...

(Nom) diagnosis

Start

Stop

Result list (right-click for options)

20:41:59 - trees.J48

20:43:40 - meta.Bagging

20:45:17 - meta.Bagging

20:46:21 - meta.AdaBoostM1

Classifier output

Time taken to build model: 0.1 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances

539

94.5518 %

Incorrectly Classified Instances

31

5.4482 %

Kappa statistic

0.8834

Mean absolute error

0.0578

Root mean squared error

0.1967

Relative absolute error

12.3539 %

Root relative squared error

40.6882 %

Total Number of Instances

569

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Classification
M	0.925	0.042	0.929	0.925	0.927	0.883	0.989	0.965	M
B	0.958	0.075	0.955	0.958	0.957	0.883	0.989	0.993	B
Weighted Avg.	0.946	0.063	0.945	0.946	0.945	0.883	0.989	0.990	

=== Confusion Matrix ===

a

b

<-- classified as

196

16

|

a = M

15

342

|

b = B

Status

OK

Log

x0

weka.gui.GenericObjectEditor

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

seed

1

useResampling

False

weightThreshold

100

Open...

Save...

OK

Cancel

- i. 在Classifier中選取「AdaBoost」
- ii. 在參數調整的地方，將classifier設為「DecisionStump」
- iii. Test options 使用 Cross-validation，並設定Folds為10
- iv. 選取屬性「(Nom)diagnosis」
- v. 點選「Start」

三、請用python對BreastCancer.csv對目標diagnosis進行Ensemble learning並與未使用的結果進行比較(請列出重要過程及適當說明)：

```
1 import pandas as pd
2 import numpy as np
3 from sklearn.model_selection import train_test_split
4
5 #讀取CSV檔案
6 data = pd.read_csv('BreastCancer.csv')
7
8 feature = data.iloc[:,2:32]
9 target = data['diagnosis']
10
11 # 切分訓練與測試資料
12 train_X, test_X, train_y, test_y = train_test_split(feature, target, test_size = 0.4, random_state=0)
```

- i. 首先將csv檔匯入
- ii. 將資料切割為feature與target
- iii. 切分為訓練資料與測試資料(用於未使用Ensemble learning的部分)

(a). 以10 Folds cross-validation進行DecisionTreeClassifier分類(5%)

```
1 from sklearn.tree import DecisionTreeClassifier
2 from sklearn.model_selection import cross_val_score
3 from sklearn import metrics
4
5 # 建立 DecisionTree 模型
6 clf = DecisionTreeClassifier()
7
8 # 進行 Ensemble Learning
9 tree_scores = cross_val_score(estimator=clf, X=feature, y=target, cv=10)
10
11 # 不使用 Ensemble Learning
12 BreastCancer_clf = clf.fit(train_X, train_y)
13 # 預測
14 test_y_predicted = BreastCancer_clf.predict(test_X)
15 # 績效
16 accuracy = metrics.accuracy_score(test_y, test_y_predicted)
17
18 print('使用Ensemble Learning 準確度為：', tree_scores.mean())
19 print('未使用Ensemble Learning 準確度為：', accuracy)
```

使用Ensemble Learning 準確度為： 0.921134733385187
未使用Ensemble Learning 準確度為： 0.9035087719298246

(b). 以10 Folds cross-validation進行BaggingClassifier, n_estimators=10分類(10%)

```
1 from sklearn.ensemble import BaggingClassifier
2
3 # 建立 bagging 模型
4 bag = BaggingClassifier(n_estimators = 10)
5
6 # 進行 Ensemble Learning
7 bag_scores = cross_val_score(estimator=bag, X=feature, y=target, cv=10, n_jobs=4)
8
9 # 不使用 Ensemble Learning
10 BreastCancer_bag = bag.fit(train_X, train_y)
11 # 預測
12 test_y_predicted = BreastCancer_bag.predict(test_X)
13 # 績效
14 accuracy = metrics.accuracy_score(test_y, test_y_predicted)
15
16 print('使用Ensemble Learning 準確度為：', bag_scores.mean())
17 print('未使用Ensemble Learning 準確度為：', accuracy)
```

使用Ensemble Learning 準確度為： 0.9526877538674272
未使用Ensemble Learning 準確度為： 0.9473684210526315

(c). 以10 Folds cross-validation進行RandomForestClassifier分類(10%)

```
1 from sklearn.ensemble import RandomForestClassifier
2
3 # 建立 RandomForest 模型
4 forest = RandomForestClassifier(n_estimators = 10)
5
6 # 進行 Ensemble Learning
7 forest_scores = cross_val_score(estimator=forest, X=feature, y=target, cv=10, n_jobs=4)
8
9 # 不使用 Ensemble Learning
10 BreastCancer_forest = forest.fit(train_X, train_y)
11 # 預測
12 test_y_predicted = BreastCancer_forest.predict(test_X)
13 # 績效
14 accuracy = metrics.accuracy_score(test_y, test_y_predicted)
15
16 print('使用Ensemble Learning 準確度為：', forest_scores.mean())
17 print('未使用Ensemble Learning 準確度為：', accuracy)
```

使用Ensemble Learning 準確度為： 0.9475131795004753
未使用Ensemble Learning 準確度為： 0.9342105263157895

(d). 以10 Folds cross-validation進行AdaBoost,n_estimators=10分類(10%)

```
1 from sklearn.ensemble import AdaBoostClassifier
2
3 # 建立 AdaBoost 模型
4 boost = AdaBoostClassifier(n_estimators = 10)
5
6 # 進行 Ensemble Learning
7 boost_scores = cross_val_score(estimator=boost, X=feature, y=target, cv=10, n_jobs=4)
8
9 # 不使用 Ensemble Learning
10 BreastCancer_boost = boost.fit(train_X, train_y)
11 # 預測
12 test_y_predicted = BreastCancer_boost.predict(test_X)
13 # 績效
14 accuracy = metrics.accuracy_score(test_y, test_y_predicted)
15
16 print('使用Ensemble Learning 準確度為：', boost_scores.mean())
17 print('未使用Ensemble Learning 準確度為：', accuracy)
```

使用Ensemble Learning 準確度為： 0.9578893354074841
未使用Ensemble Learning 準確度為： 0.956140350877193

結論：使用 Ensemble learning 通常會有較佳的準確度，通過組合多個模型，集成學習有助於提高機器學習效果。與單個模型相比，該方法允許產生更好的預測性能。