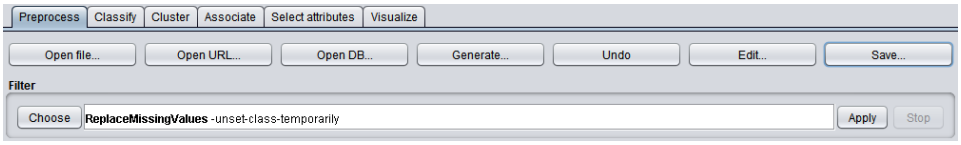
2019 ECT 作業八

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

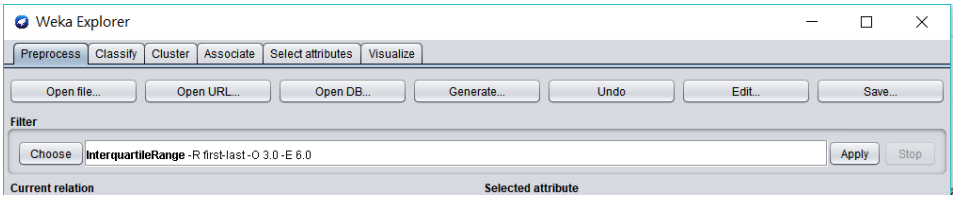
(a). Replace Missing Value，需列出補上的值為何(10%)



i. 在Preprocess的Filter選取「weka/filters/unsupervised/attribute/ReplaceMissingValues」

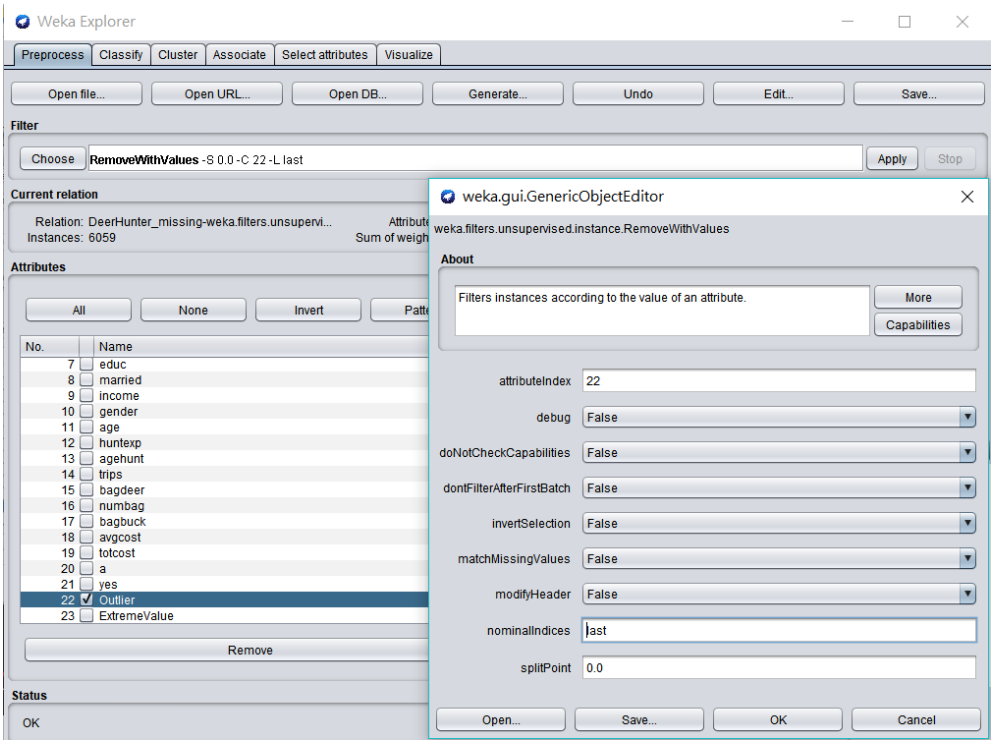
ii. 點選「Apply」，可將MissingValue值變更為MeanValue

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



i. 在Preprocess的Filter選取「weka/filters/unsupervised/attribute/InterquartileRange」

ii. 點選「Apply」，會多出兩個Attribute(Outlier與ExtremeValue)



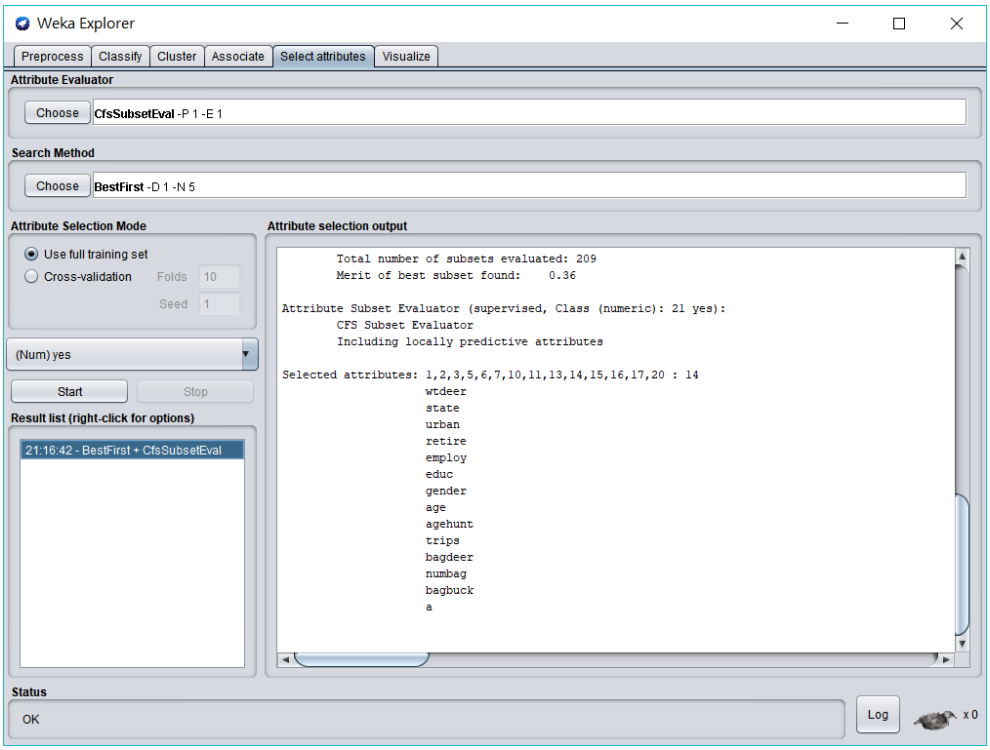
iii. 在Preprocess的Filter選取「weka/filters/unsupervised/instance/RemoveWithValues」

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」

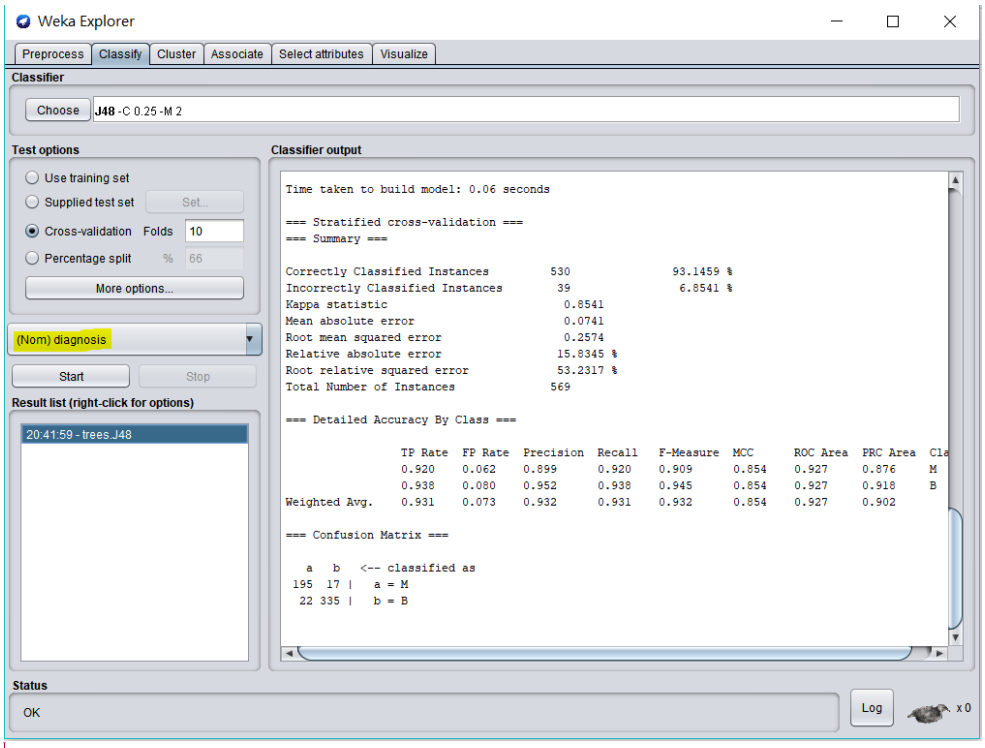
ii. 將屬性設為「(Num)yes」

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)：

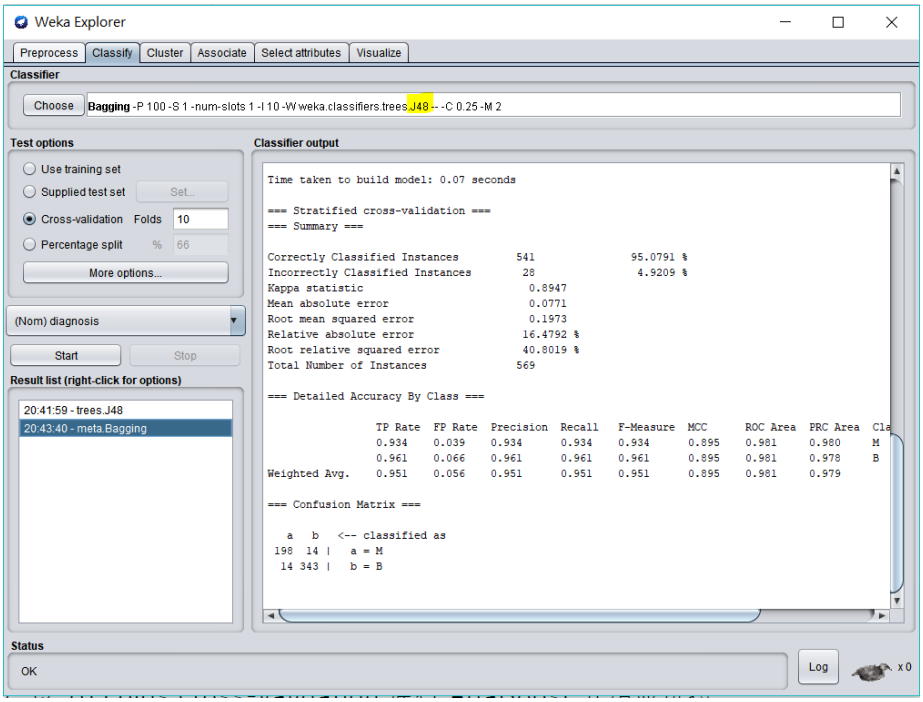
(a). 以10 Folds cross-validation進行J48分類(5%)



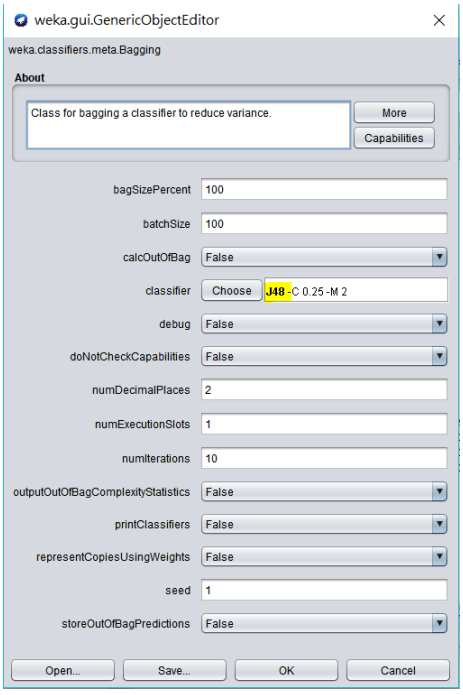
已註解 [柏丞1]:

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」



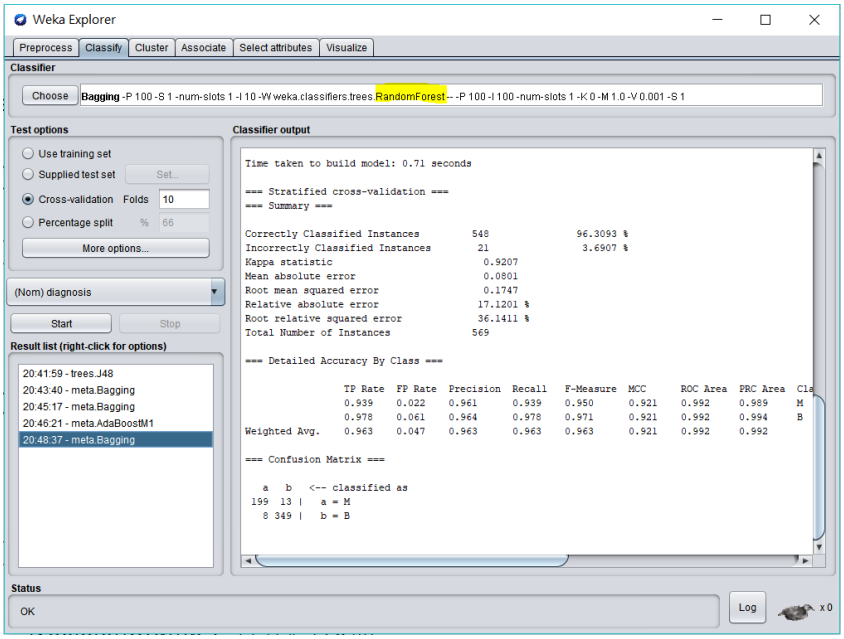
ii. 在參數調整的地方，將classifier設為「J48」

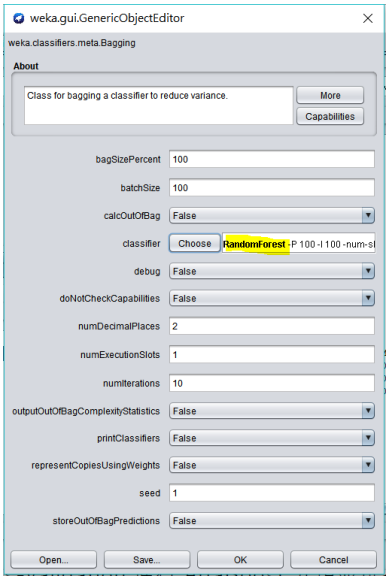
iii. Test options 使用 Cross-validation，並設定Folds為10

iv. 選取屬性「(Nom)diagnosis」

v. 點選「Start」

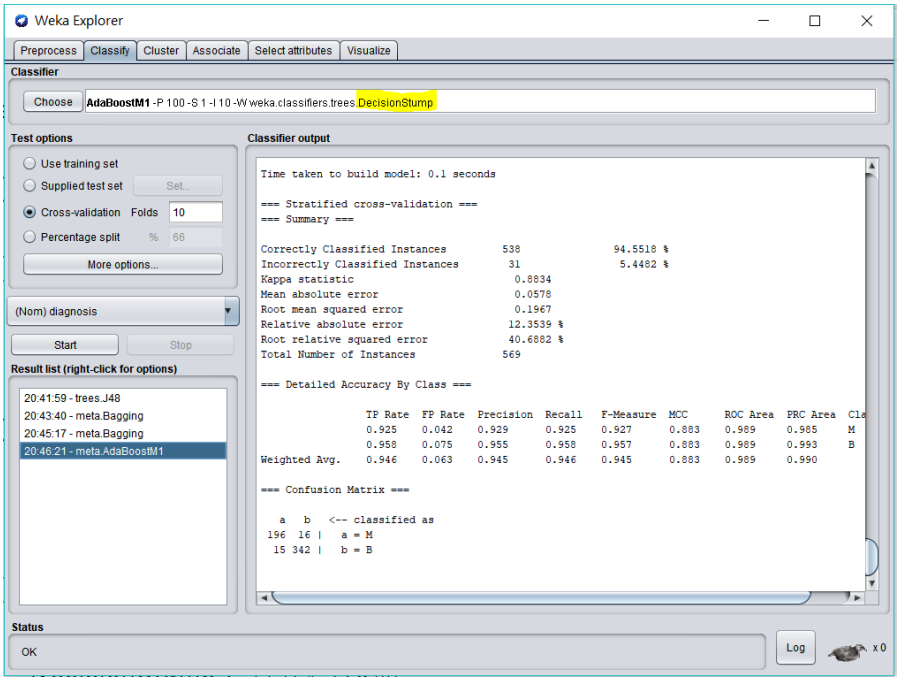
(c). 以10 Folds cross-validation進行Bagging分類並選擇Randomforest進行分類(10%)

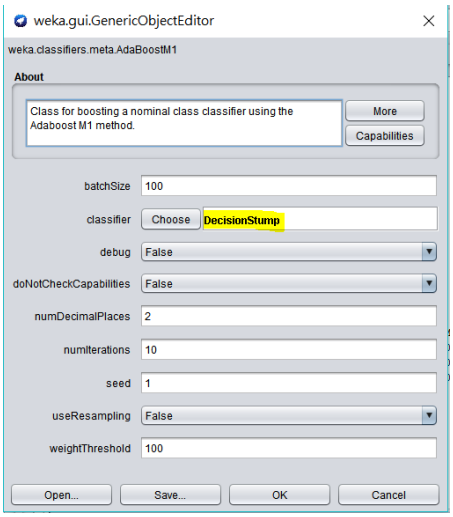




i. 在參數調整的地方，將classifier設為「RandomForest」
ii. Test options 使用 Cross-validation，並設定Folds為10
iii. 選取屬性「(Nom)diagnosis」
iv. 點選「Start」

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





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)：

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

#讀取CSV檔案
data = pd.read_csv('BreastCancer.csv')

feature = data.iloc[:,2:32]
target = data['diagnosis']

# 切分訓練與測試資料
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%)

```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
from sklearn import metrics

# 建立 DecisionTree 模型
clf = DecisionTreeClassifier()

# 進行 Ensemble learning
tree_scores = cross_val_score(estimator=clf, X=feature, y=target, cv=10)

# 不使用 Ensemble learning
BreastCancer_clf = clf.fit(train_X, train_y)
# 預測
test_y_predicted = BreastCancer_clf.predict(test_X)
# 績效
accuracy = metrics.accuracy_score(test_y, test_y_predicted)

print('使用Ensemble Learning 準確度為：', tree_scores.mean())
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%)

```
from sklearn.ensemble import BaggingClassifier

# 建立 bagging 模型
bag = BaggingClassifier(n_estimators = 10)

# 進行 Ensemble learning
bag_scores = cross_val_score(estimator=bag, X=feature, y=target, cv=10, n_jobs=4)

# 不使用 Ensemble learning
BreastCancer_bag = bag.fit(train_X, train_y)
# 預測
test_y_predicted = BreastCancer_bag.predict(test_X)
# 績效
accuracy = metrics.accuracy_score(test_y, test_y_predicted)

print('使用Ensemble Learning 準確度為：', bag_scores.mean())
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%)

```
from sklearn.ensemble import RandomForestClassifier

# 建立 RandomForest 模型
forest = RandomForestClassifier(n_estimators = 10)

# 進行 Ensemble learning
forest_scores = cross_val_score(estimator=forest, X=feature, y=target, cv=10, n_jobs=4)

# 不使用 Ensemble learning
BreastCancer_forest = forest.fit(train_X, train_y)
# 預測
test_y_predicted = BreastCancer_forest.predict(test_X)
# 績效
accuracy = metrics.accuracy_score(test_y, test_y_predicted)

print('使用Ensemble Learning 準確度為：', forest_scores.mean())
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%)

```python
from sklearn.ensemble import AdaBoostClassifier

# 建立 AdaBoost 模型
boost = AdaBoostClassifier(n_estimators = 10)

# 進行 Ensemble learning
boost_scores = cross_val_score(estimator=forest, X=feature, y=target, cv=10, n_jobs=4)

# 不使用 Ensemble learning
BreastCancer_boost = boost.fit(train_X, train_y)
# 預測
test_y_predicted = BreastCancer_boost.predict(test_X)
# 績效
accuracy = metrics.accuracy_score(test_y, test_y_predicted)

print('使用Ensemble Learning 準確度為：', boost_scores.mean())
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。