2019 ECT 作業六

一、 請用 python 依照步驟對 BreastCancer.csv 進行 KNN 及 KMeans 分析，過程中對所有重要程式步驟進行截圖並加以說明，越詳盡越好。(80%)

首先，引入需要使用的套件與讀取 csv 檔。

```
import pandas as pd
#讀取CSV檔案
data = pd.read_csv('BreastCancer.csv')
```

(a) 將 radius_mean 及 area_mean 切為 feature，diagnosis 切為 target

```
Target = data['diagnosis']

#將屬性合併
#變成list
feature=list(zip(data['radius_mean'],data['area_mean']))

import numpy as np
#轉成array
features=np.asarray(feature)

from sklearn import preprocessing
#轉換屬性型態
#將屬性轉為數字label
le = preprocessing.LabelEncoder()
#將 diagnosis 轉為數字 diagnosis
Target=le.fit_transform(Target)
```

(b) 切分資料及與訓練集，設 test_size=0.34，random_state=5

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(feature, Target, test_size=0.34, r
```

(c) 用 KNeighborsClassifier 進行 KNN 分類，n_neighbors 設為 6

```
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors=6)
neigh.fit(X_train, y_train)
y_result = neigh.predict(X_test)
```

(d) 用 metrics 算出此模型對於測試集的準確度

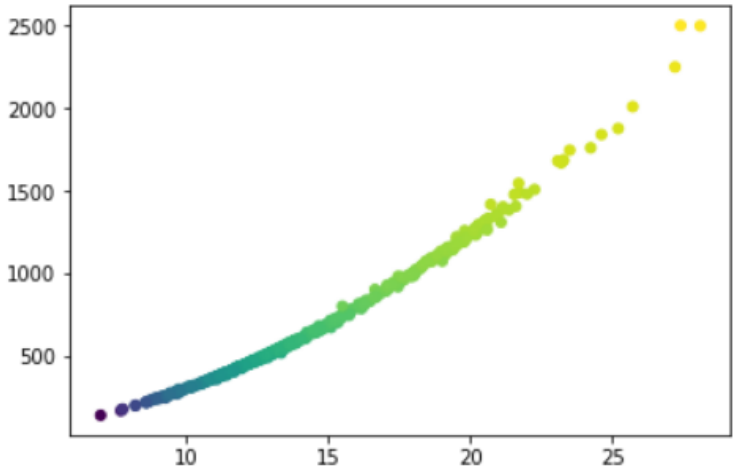
```
from sklearn.metrics import accuracy_score
score = accuracy_score(y_test, y_result)
print("準確率為 :  %f" % score)
```

```
準確率為 :  0.917526
```

(e) 運用 matplotlib 中的 scatter 圖，x 軸設為 radius_mean，y 軸設為 area_mean，c 設為 label，印出測試分類圖形。

```
import matplotlib.pyplot as plt
X = data[['radius_mean','area_mean']].values
label = np.arctan2(X[:, 1], X[:, 0])
plt.scatter(X[:, 0], X[:, 1], s=25, c=label, alpha=1)
```

```
<matplotlib.collections.PathCollection at 0x243ecd333c8>
```



(f) 用 cluster.Means 設 n_clusters=2

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=2)
```

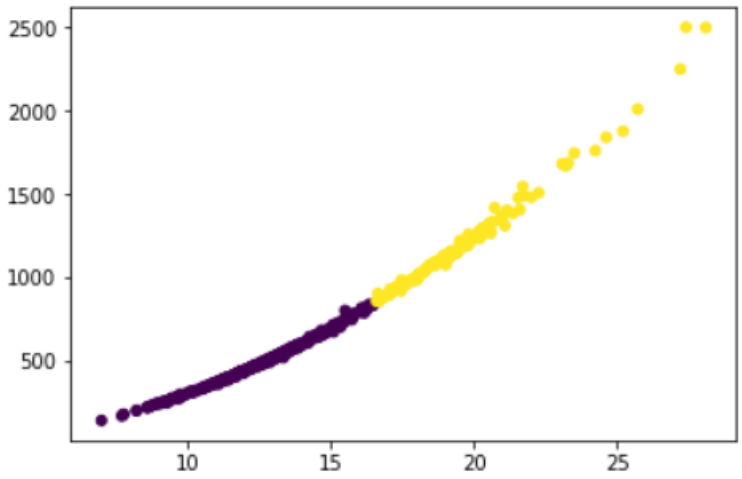
(g) 用 fit_predict 對切分的 feature 進行預測

```
# 使用 K-Means 演算法
clusters = kmeans.fit_predict(X)
```

(h) 運用 matplotlib 中的 scatter 圖，x 軸設為 radius_mean，y 軸設為 area_mean，c 設為分群結果，印出分類圖形。

```
plt.scatter(X[:, 0], X[:, 1], s=25, c=clusters, alpha=1)
```

```
<matplotlib.collections.PathCollection at 0x243ec73d470>
```

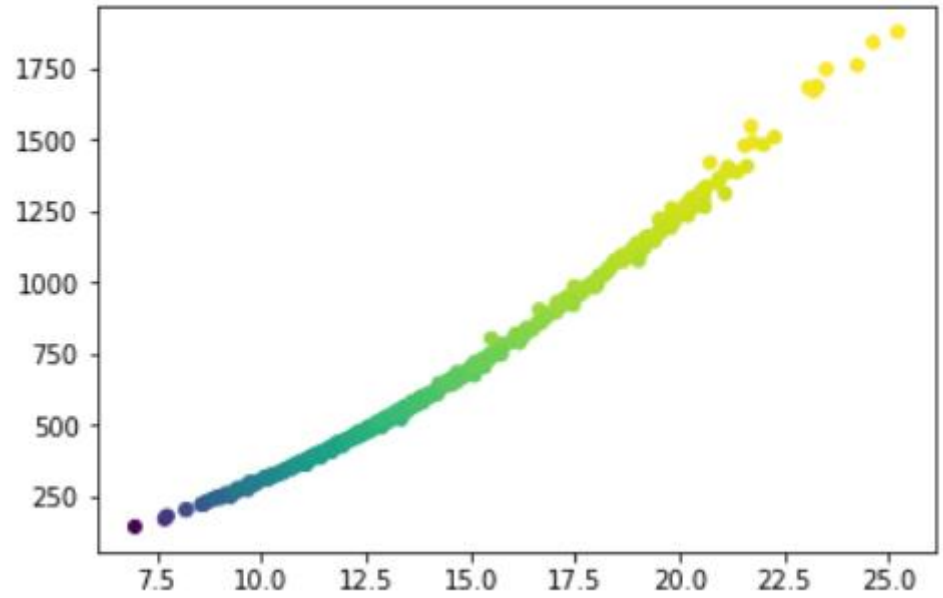


(i) 移除 area_mean 中大於 2000 的資料

```
data = data[data.area_mean <= 2000]
```

(j) 重複上述 a~e 的動作，同時回答問題：在此案例中移除與分布較遠的資料是否有達到更好的效果?

準確率為 ： 0.886010



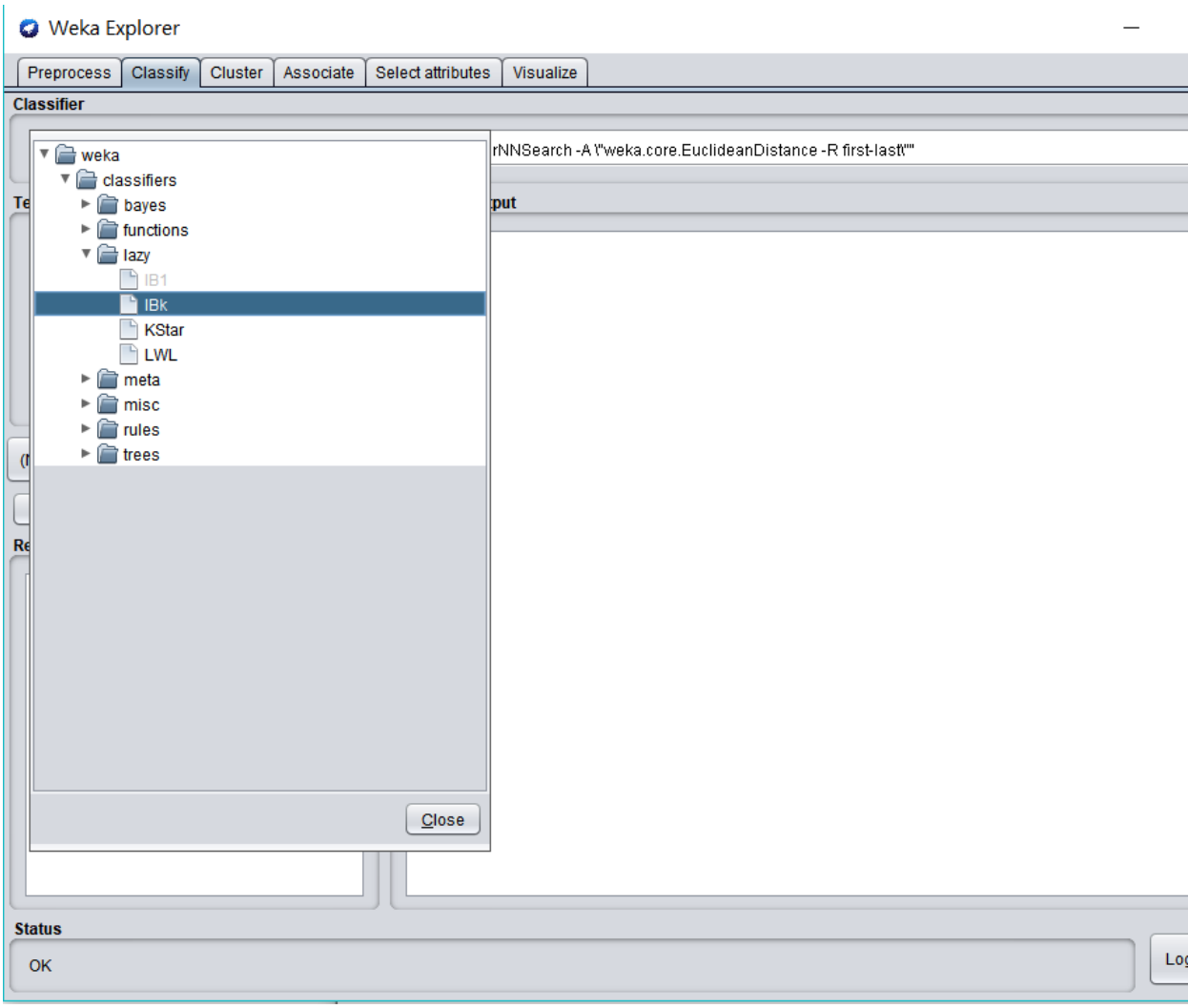
並沒有達到更好的效果。準確率從 0.9175 下降至 0.886

二、 請用 weka 對 BreastCancer.csv，進行 IBK(knn) k 設為 6 及 simplekMeans 進行分析，Percentage split 設為 66%，截圖並附上過程及準確率。(20%)

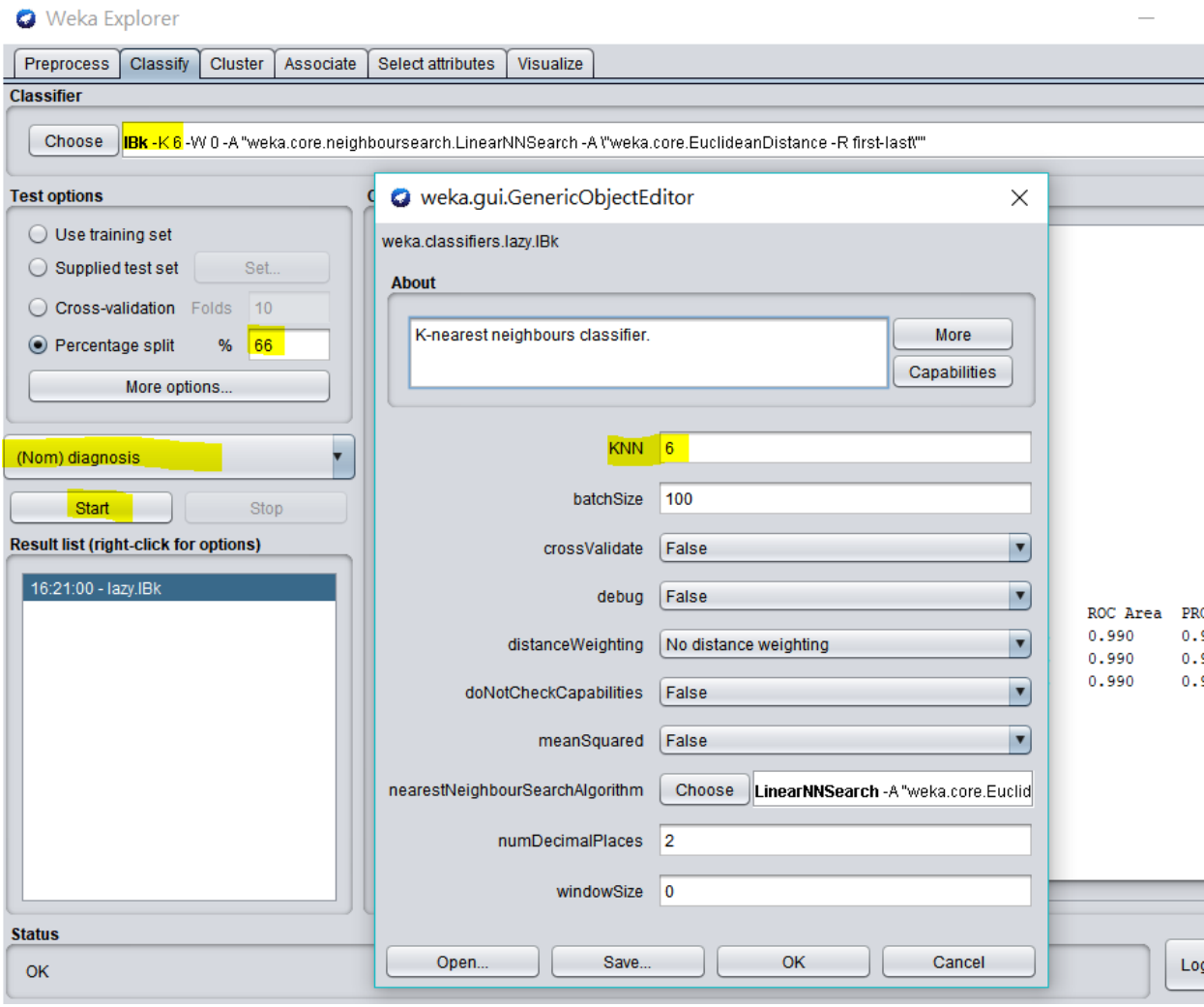
首先在 Weka 中開啟 BreastCancer.csv。

IBK(knn) 分析

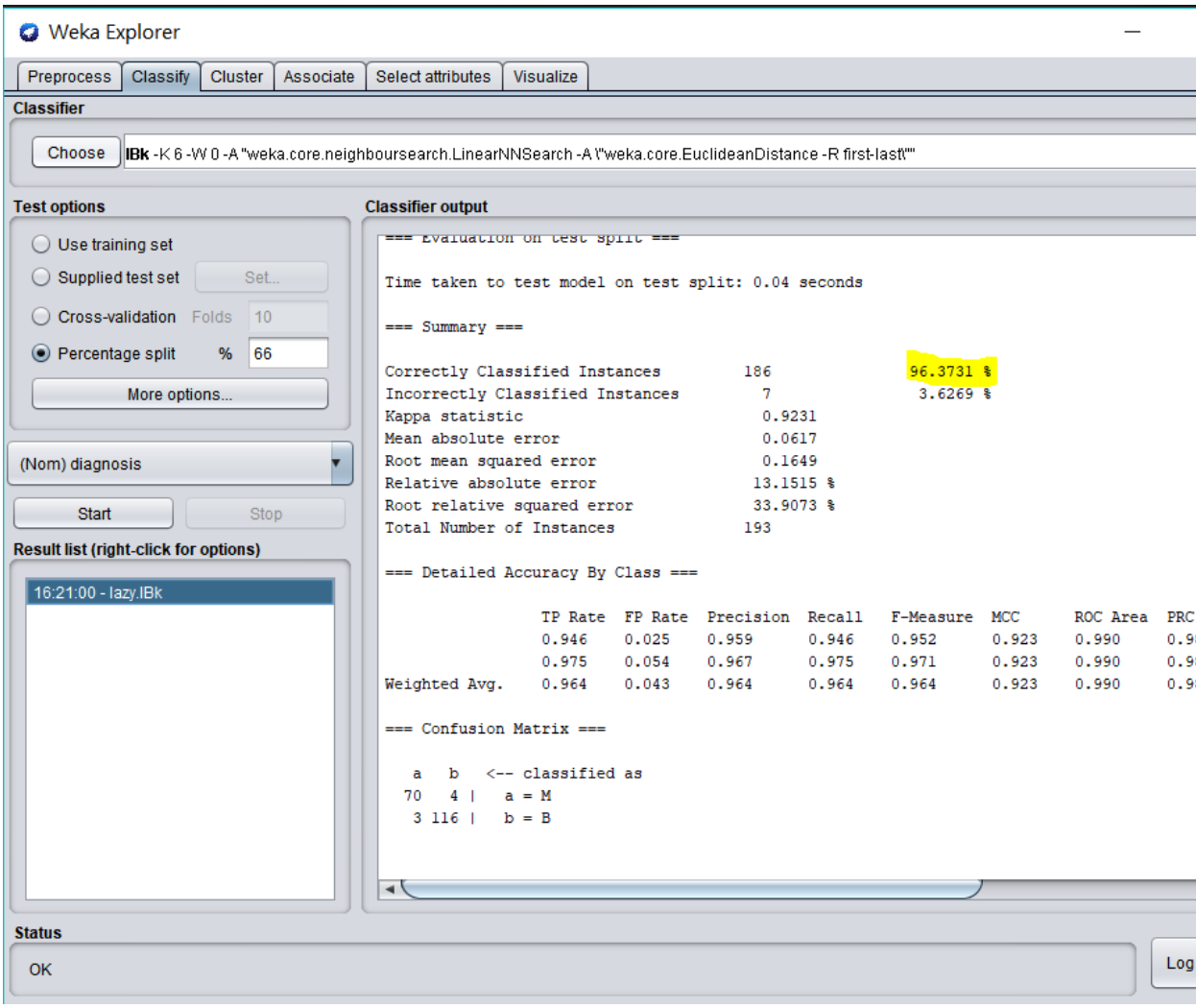
i. 在 Classify 面板中，在 Classifier 選擇「weka / classifiers / lazy/ IBK」。



ii. 點選上方的參數設定，將「KNN」設置成 6，在「Test options」中選取「Percentage split」，並設定為 66%；選擇預測「(Nom)diagnosis」，並點選「Start」。

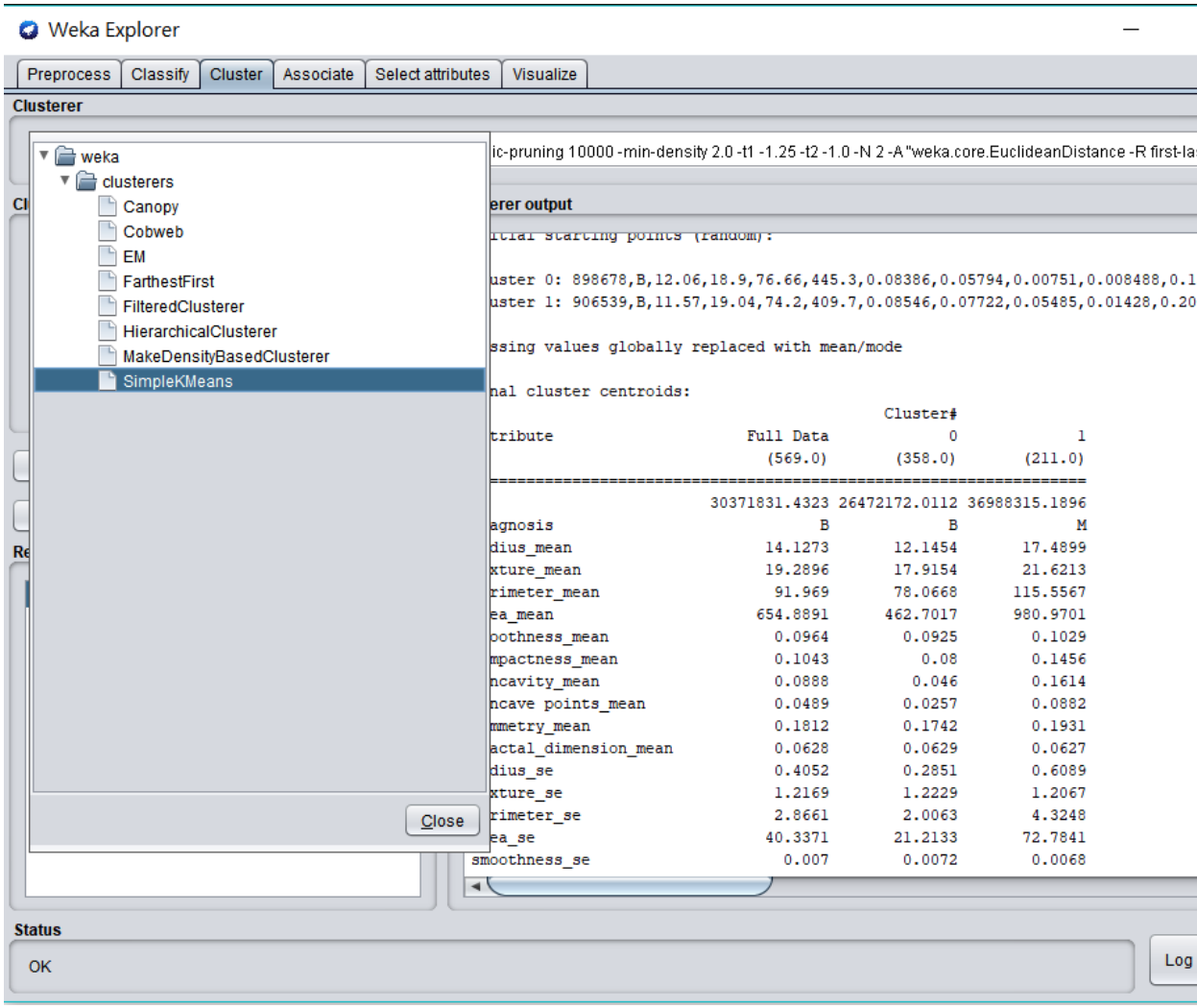


iii. 可得準確度為 96.3731 %



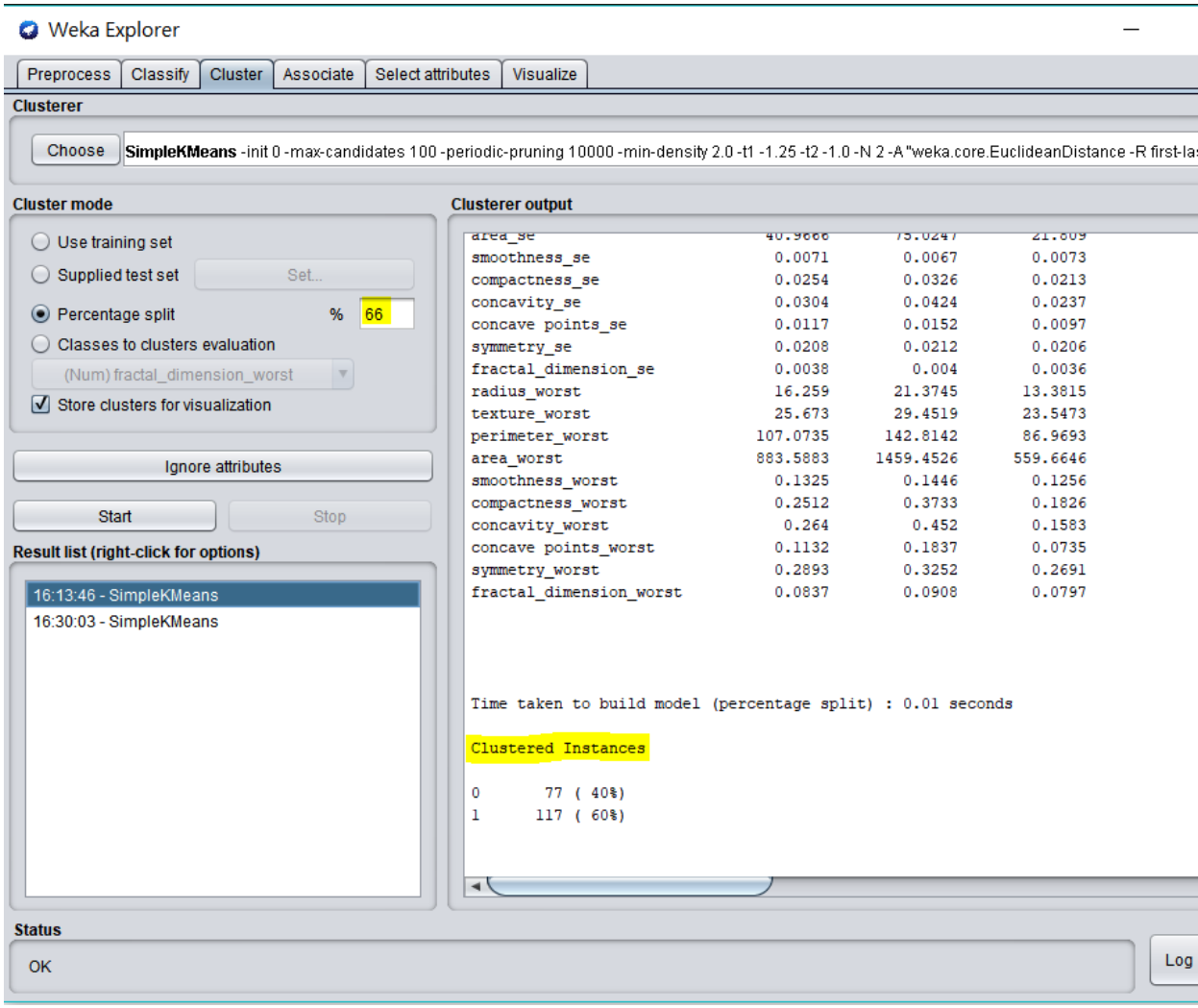
simplekMeans 分析

i. 在 Cluster 面板中，在 Cluster 選擇「weka / clusters/ **simplekMeans**」。



ii. 在「Test options」中選取「Percentage split」，並設定為 66%，並點選「Start」。

可得 Clustered Instance，有 40%的 instance 被歸類到 0，60%的 instance 被歸類到 1。


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-las
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Num) fractal_dimension_worst
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
16:13:46 - SimpleKMeans
16:30:03 - SimpleKMeans
Clusterer output
smoothness_se 0.0071 0.0067 0.0073
compactness_se 0.0254 0.0326 0.0213
concavity_se 0.0304 0.0424 0.0237
concave points_se 0.0117 0.0152 0.0097
symmetry_se 0.0208 0.0212 0.0206
fractal_dimension_se 0.0038 0.004 0.0036
radius_worst 16.259 21.3745 13.3815
texture_worst 25.673 29.4519 23.5473
perimeter_worst 107.0735 142.8142 86.9693
area_worst 883.5883 1459.4526 559.6646
smoothness_worst 0.1325 0.1446 0.1256
compactness_worst 0.2512 0.3733 0.1826
concavity_worst 0.264 0.452 0.1583
concave points_worst 0.1132 0.1837 0.0735
symmetry_worst 0.2893 0.3252 0.2691
fractal_dimension_worst 0.0837 0.0908 0.0797
Time taken to build model (percentage split) : 0.01 seconds
Clustered Instances
0 77 (40%)
1 117 (60%)
Status
OK
Log