ECT_HW4

2019

第一大題

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%)

knn

In [1]:
```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn import metrics
import matplotlib.pyplot as plt
```

In [2]:
```
data = pd.read_csv('BreastCancer.csv')
```

In [3]:
```
feature = data.loc[:,['radius_mean','area_mean']]
label=data.iloc[:,1]
```

In [4]:
```
le = preprocessing.LabelEncoder()
encodedlabel = le.fit_transform(label)
```

In [5]:
```
X_train, X_test, y_train, y_test = train_test_split(feature,encodedlabel, test_size=0.34,random_state = 5)
```

In [6]:
```
knn = KNeighborsClassifier(n_neighbors=6)

# Train the model using the training sets
knn.fit(X_train, y_train)

predict = knn.predict(X_test)
```

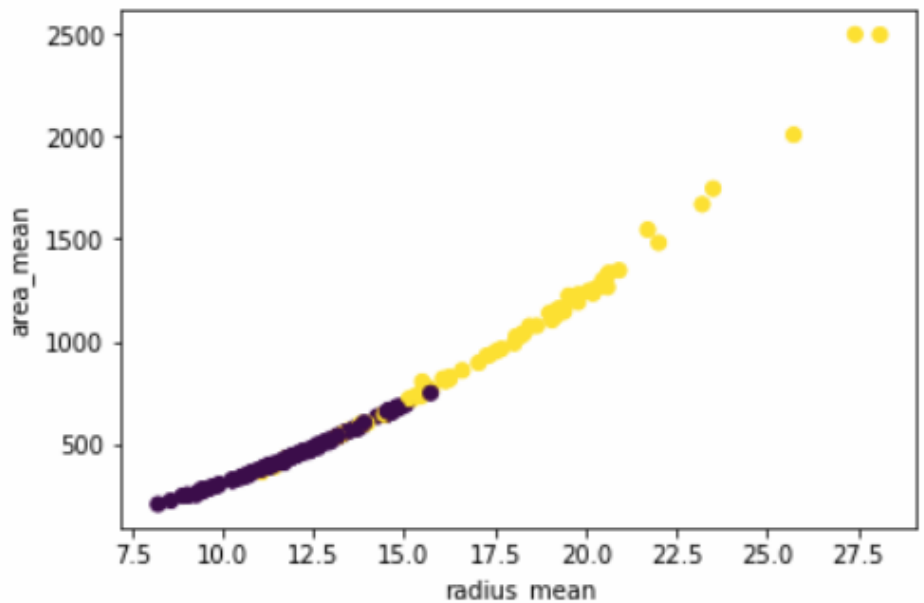
In [7]:
```
accuracy = metrics.accuracy_score(predict, y_test)
```

In [8]:
```
accuracy
```

Out[8]:
```
0.9175257731958762
```

In [11]:
```
plt.scatter(X_test.values[:,0],X_test.values[:,1],c=y_test)
plt.xlabel('radius_mean')
plt.ylabel('area_mean')
```

Out[11]:
```
Text(0, 0.5, 'area_mean')
```

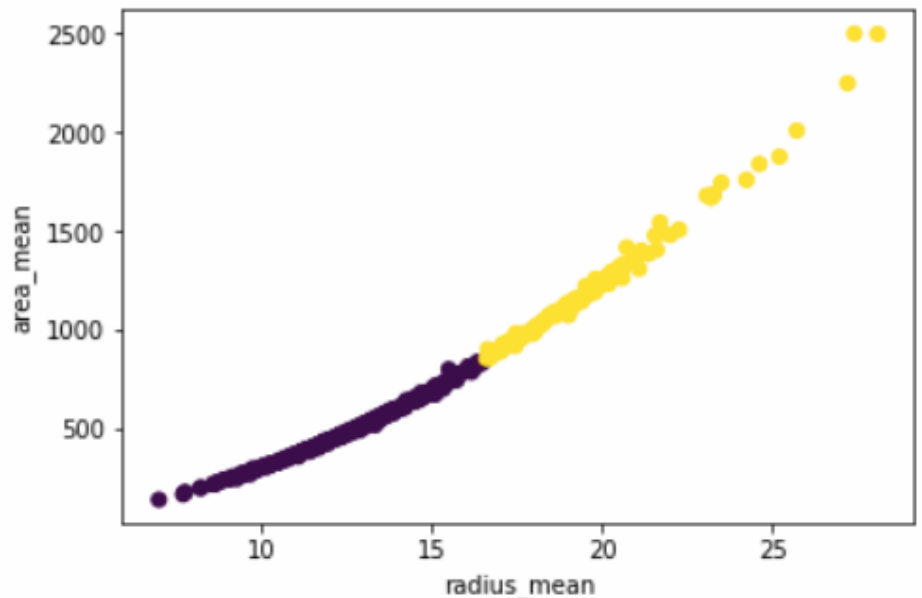


Kmeans

In [151]:
```
from sklearn import cluster
```

In [160]:
```
km = cluster.KMeans(n_clusters=2)  #K=2群
y_pred = km.fit_predict(feature)
```

In [161]:
```
plt.xlabel('radius_mean')
plt.ylabel('area_mean')
plt.scatter(feature.values[:,0],feature.values[:,1],c=y_pred) #C是第三維度 已顏色做維度
plt.show()
```



In [170]:
```
newData = data[data.area_mean<2000]
```

In [171]:
```
newfeature = newData.loc[:,['radius_mean','area_mean']]
newlabel=newData.iloc[:,1]
```

In [172]:
```
le = preprocessing.LabelEncoder()
encodedlabel = le.fit_transform(newlabel)
```

In [173]:
```
X_train, X_test, y_train, y_test = train_test_split(newfeature,encodedlabel, test_size=0.34,random_state = 5)
```

In [174]:
```
knn = KNeighborsClassifier(n_neighbors=6)

# Train the model using the training sets
knn.fit(X_train, y_train)

predict = knn.predict(X_test)
```

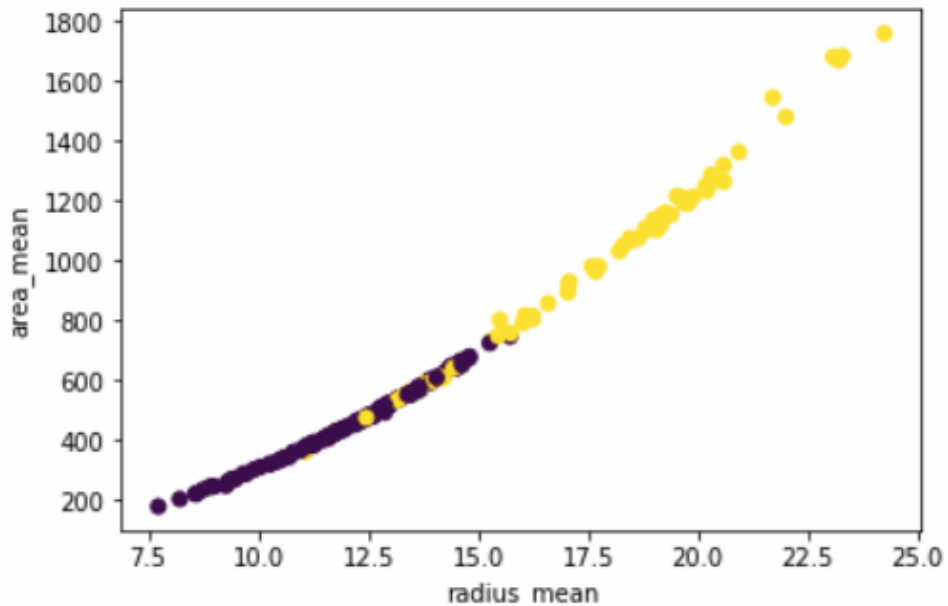
In [175]:
```
accuracy = metrics.accuracy_score(predict, y_test)
```

In [176]:
```
accuracy
```

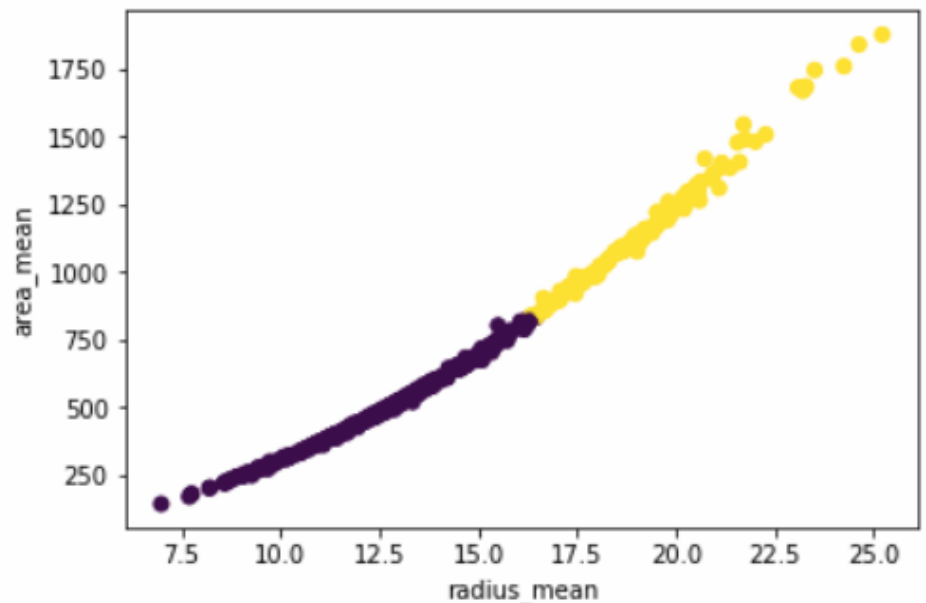
Out[176]: 0.8860103626943006

In [177]:
```
plt.scatter(X_test.values[:,0],X_test.values[:,1],c=y_test)
plt.xlabel('radius_mean')
plt.ylabel('area_mean')
```

Out[177]: Text(0, 0.5, 'area_mean')

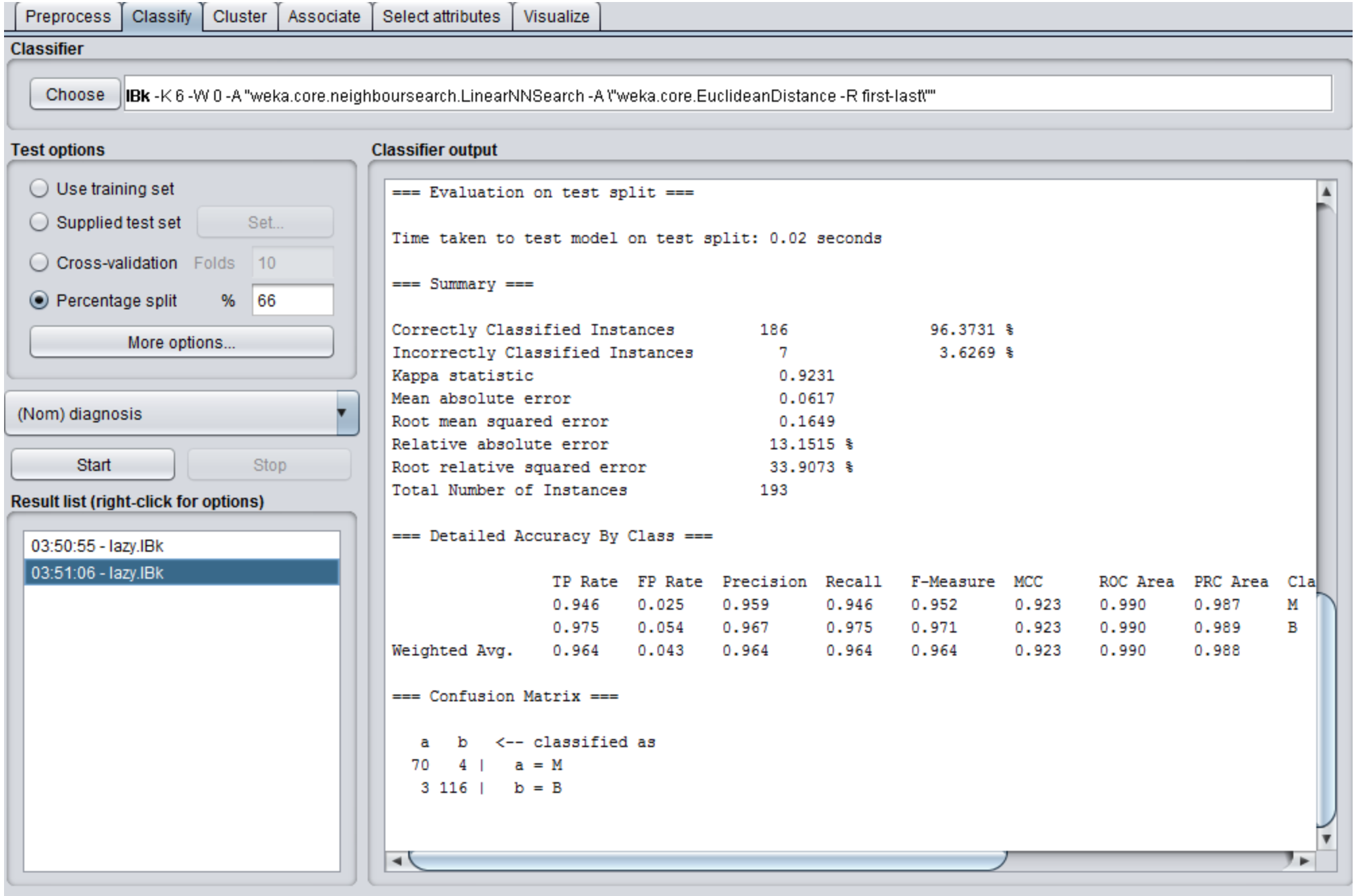


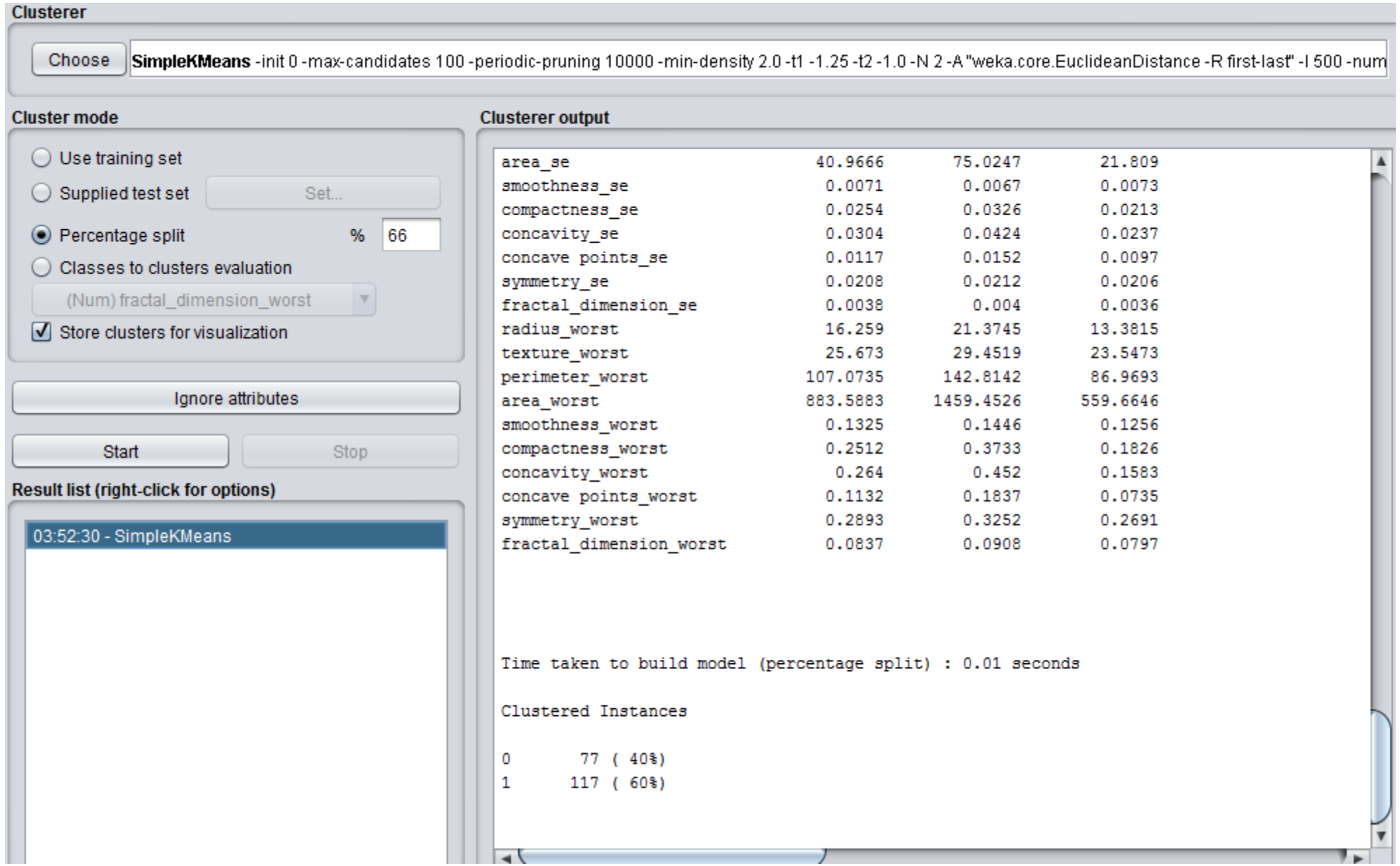
In [178]:
```
km = cluster.KMeans(n_clusters=2)
y_pred = km.fit_predict(newfeature)
plt.xlabel('radius_mean')
plt.ylabel('area_mean')
plt.scatter(newfeature.values[:,0],newfeature.values[:,1],c=y_pred)
plt.show()
```



第二大題

請用 weka 對 Titanic.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%)


Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
IBk -K 6 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) diagnosis
Start
Stop
Result list (right-click for options)
03:50:55 - lazy.IBk
03:51:06 - lazy.IBk
Classifier output
=== Evaluation on test split ===
Time taken to test model on test split: 0.02 seconds
=== Summary ===
Correctly Classified Instances 186 96.3731 %
Incorrectly Classified Instances 7 3.6269 %
Kappa statistic 0.9231
Mean absolute error 0.0617
Root mean squared error 0.1649
Relative absolute error 13.1515 %
Root relative squared error 33.9073 %
Total Number of Instances 193
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Cla
0.946 0.025 0.959 0.946 0.952 0.923 0.990 0.987 M
0.975 0.054 0.967 0.975 0.971 0.923 0.990 0.989 B
Weighted Avg. 0.964 0.043 0.964 0.964 0.964 0.923 0.990 0.988
=== Confusion Matrix ===
a b <-- classified as
70 4 | a = M
3 116 | b = B


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-last" -I 500 -num
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)
03:52:30 - SimpleKMeans
Clusterer output
area_se 40.9666 75.0247 21.809
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%)