In [12]:

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

In [13]:

```
# 讀取資料
from sklearn.datasets import load_breast_cancer
data = load_breast_cancer(return_X_y=False)
X, y = data.data, data.target
```

In [14]:

```
# 由以下資料可知 y=0代表惡性腫瘤，y=1代表良性腫瘤
data.target_names
```

Out[14]:

```
array(['malignant', 'benign'], dtype='<U9')
```

In [39]:

```
import pandas as pd
# 儲存模型
metrics = pd.DataFrame({'mean_accuracy':[],'mean_precision':[],'mean_recall':[]}).T
```

In [53]:

```
?LogisticRegression
```

Model1：一般線性模型

In [44]:

```
import sklearn
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.metrics import confusion_matrix
rskf = RepeatedStratifiedKFold(n_splits=10, n_repeats = 2)
accuracy = []
precision = []
recall = []
# Fit model
clf = LogisticRegression(penalty='l2', solver='sag', max_iter=200)
for train_index, test_index in rskf.split(X, y):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
    clf = clf.fit(X_train,y_train)

    # test
    cm = confusion_matrix(clf.predict(X_test),y_test)
    # 劃出confusion matrix(因為篇幅關係沒有畫出來，避免同學看得混亂)
    # sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', annot_kws={'size':50})
    # plt.title('confusion matrix')
    # plt.ylabel('Actual')
    # plt.xlabel('Predicted')
    # plt.show()

    # 臨床實驗中 1 代表陽性，但此題因為0的位置代表惡性腫瘤，所以取 0 的位置當作陽性
    # 這邊如果使用sklearn,會誤判 recall與 precision
    # 所以手動定義
    TP, FN = cm[0,0], cm[0,1]
    FP, TN = cm[1,0], cm[1,1]
    accuracy.append(clf.score(X_test,y_test))
    precision.append(TP/(TP+FP))
    recall.append(TP/(TP+FN))

print('mean accuracy:{:.2f}'.format(np.mean(accuracy)))
print('mean precision:{:.2f}'.format(np.mean(precision)))
print('mean recall:{:.2f}'.format(np.mean(recall)))
# 存下來
metrics['Model1']=[np.mean(accuracy),np.mean(precision),np.mean(recall)]
metrics
```

```
mean accuracy:0.92
mean precision:0.83
mean recall:0.95
```

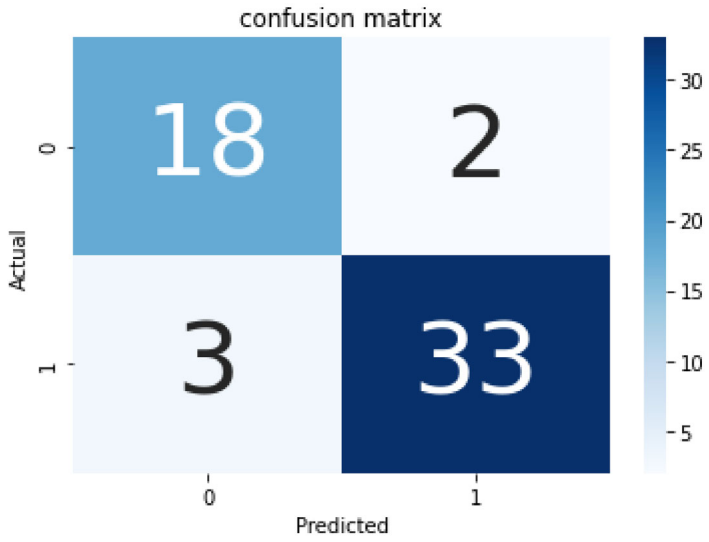
Out[44]:

	Model1
mean_accuracy	0.918296
mean_precision	0.829654
mean_recall	0.946747

In [16]:

```
# 畫一次confusion matrix (示範)
print("最後一次evaluation 的 confusion matric")
print('accuracy:{:.2f}'.format(accuracy[-1]))
print('precision:{:.2f}'.format(precision[-1]))
print('recall:{:.2f}'.format(recall[-1]))
cm = confusion_matrix(clf.predict(X_test),y_test)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', annot_kws={'size':50})
plt.title('confusion matrix')
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.show()
```

```
最後一次evaluation 的 confusion matric
accuracy:0.91
precision:0.86
recall:0.90
```



Model2 : Normalize

In [17]:

```
from sklearn.preprocessing import normalize
norm_X = normalize(X)
```

In [18]:

```
norm_X.shape
```

Out[18]:

```
(569, 30)
```

In [19]:

```
# Fit model
clf = LogisticRegression(penalty='l2', solver='sag', max_iter=200)
```

In [45]:

```
rskf = RepeatedStratifiedKFold(n_splits=10, n_repeats = 2)
accuracy = []
precision = []
recall = []
for train_index, test_index in rskf.split(norm_X, y):
    X_train, X_test = norm_X[train_index], norm_X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    clf = clf.fit(X_train,y_train)

    # test
    cm = confusion_matrix(clf.predict(X_test),y_test)
    # 劃出confusion matrix(因為篇幅關係沒有畫出來，避免同學看得混亂)
    # sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', annot_kws={'size':50})
    # plt.title('confusion matrix')
    # plt.ylabel('Actual')
    # plt.xlabel('Predicted')
    # plt.show()

    # 臨床實驗中 1 代表陽性，但此題因為0的位置代表惡性腫瘤，所以取 0 的位置當作陽性
    TP, FN = cm[0,0], cm[0,1]
    FP, TN = cm[1,0], cm[1,1]
    accuracy.append(clf.score(X_test,y_test))
    precision.append(TP/(TP+FP))
    recall.append(TP/(TP+FN))

print('mean accuracy for model2:{:.2f} '.format(np.mean(accuracy)))
print('mean precision for model2:{:.2f}'.format(np.mean(precision)))
print('mean recall for model2:{:.2f}'.format(np.mean(recall)))
# 存下來
metrics['Model2']=[np.mean(accuracy),np.mean(precision),np.mean(recall)]
metrics
```

```
mean accuracy for model2:0.79
mean precision for model2:0.44
mean recall for model2:0.99
```

Out[45]:

	Model1	Model2
mean_accuracy	0.918296	0.789975
mean_precision	0.829654	0.441450
mean_recall	0.946747	0.990909

Model3 : Standardize

In [26]:

```
from sklearn.preprocessing import StandardScaler
std = StandardScaler()
std_X = std.fit_transform(X)
```

In [18]:

```
std_X.shape
```

Out[18]:

```
(569, 30)
```

In [28]:

```
# Fit model
clf = LogisticRegression(penalty='l2', solver='sag', max_iter=200)
```

In [46]:

```
rskf = RepeatedStratifiedKFold(n_splits=10, n_repeats = 2)
accuracy = []
precision = []
recall = []
for train_index, test_index in rskf.split(std_X, y):
    X_train, X_test = std_X[train_index], std_X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    clf = clf.fit(X_train,y_train)

    # test
    cm = confusion_matrix(clf.predict(X_test),y_test)
    # 劃出confusion matrix(因為篇幅關係沒有畫出來，避免同學看得混亂)
    # sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', annot_kws={'size':50})
    # plt.title('confusion matrix')
    # plt.ylabel('Actual')
    # plt.xlabel('Predicted')
    # plt.show()

    # 臨床實驗中 1 代表陽性，但此題因為0的位置代表惡性腫瘤，所以取 0 的位置當作陽性
    TP, FN = cm[0,0], cm[0,1]
    FP, TN = cm[1,0], cm[1,1]
    accuracy.append(clf.score(X_test,y_test))
    precision.append(TP/(TP+FP))
    recall.append(TP/(TP+FN))

print('mean accuracy:{:.2f}'.format(np.mean(accuracy)))
print('mean precision:{:.2f}'.format(np.mean(precision)))
print('mean recall:{:.2f}'.format(np.mean(recall)))
metrics['Model3']=[np.mean(accuracy),np.mean(precision),np.mean(recall)]
metrics
```

```
mean accuracy:0.98
mean precision:0.96
mean recall:0.98
```

Out[46]:

	Model1	Model2	Model3
mean_accuracy	0.918296	0.789975	0.976222
mean_precision	0.829654	0.441450	0.960173
mean_recall	0.946747	0.990909	0.977706

Model4：取 $log(x)$ 來嘗試

In [48]:

```
log_X = np.log(X+1e-6) #因為log(0)會出錯，所以加入極小值使資料恆正
log_X.shape
```

Out[48]:

```
(569, 30)
```

In [49]:

```
# Fit model
clf = LogisticRegression(penalty='l2', solver='sag', max_iter=200)
```

In [50]:

```
rskf = RepeatedStratifiedKFold(n_splits=10, n_repeats = 2)
accuracy = []
precision = []
recall = []
for train_index, test_index in rskf.split(log_X, y):
    X_train, X_test = log_X[train_index], log_X[test_index]
    y_train, y_test = y[train_index], y[test_index]


    clf = clf.fit(X_train,y_train)

    # test
    cm = confusion_matrix(clf.predict(X_test),y_test)
    # 劃出confusion matrix(因為篇幅關係沒有畫出來，避免同學看得混亂)
    # sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', annot_kws={'size':50})
    # plt.title('confusion matrix')
    # plt.ylabel('Actual')
    # plt.xlabel('Predicted')
    # plt.show()

    # 臨床實驗中 1 代表陽性，但此題因為0的位置代表惡性腫瘤，所以取 0 的位置當作陽性
    TP, FN = cm[0,0], cm[0,1]
    FP, TN = cm[1,0], cm[1,1]
    accuracy.append(clf.score(X_test,y_test))
    precision.append(TP/(TP+FP))
    recall.append(TP/(TP+FN))

print('mean accuracy:{:.2f}'.format(np.mean(accuracy)))
print('mean precision:{:.2f}'.format(np.mean(precision)))
print('mean recall:{:.2f}'.format(np.mean(recall)))
metrics['Model4']=[np.mean(accuracy),np.mean(precision),np.mean(recall)]
metrics
```

```
mean accuracy:0.96
mean precision:0.93
mean recall:0.96
```

Out[50]:

	Model1	Model2	Model3	Model4
mean_accuracy	0.918296	0.789975	0.976222	0.959571
mean_precision	0.829654	0.441450	0.960173	0.927056
mean_recall	0.946747	0.990909	0.977706	0.964157

結果

1.單從accuracy可看出model3(standardize)結果很好。

2.Model2的recall雖然很高，但precision不好，整體而言不是很好的模型。

3.Model3的各項指數皆高於Model1與Model4，所以我選擇使用Model3(先Standardize data,然後利用LogisticRegression去fit模型)