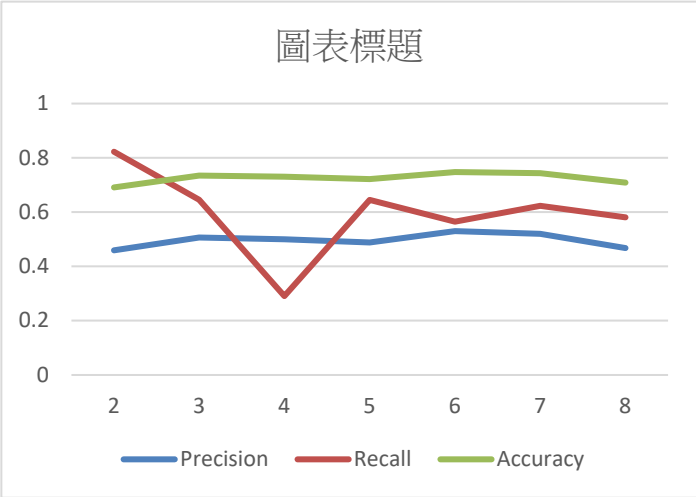


1. Precision, Recall, Accuracy

樹深限制：3

右圖為樹深2~8分別計算得到的Precision, Recall, Accuracy，下圖為依據右圖繪製的圖表
討論:

- A. Precision一直在Accuracy之下：
- 有太多沒死亡的腳色被判斷為死亡，決策樹參考的特徵可能有問題，需要再做特徵篩選。
- B. Recall在樹深=4時，劇烈的下降，但是在樹深=5時再度回升：
- 樹深=4時新納入的特徵應該有巨大的問題，若剔除此批特徵後再做一次決策樹也許能提升整體分數。
- C. 決策樹(深度3)時採用的7個特徵中，有4個是家族姓氏([X8~X29])，可見家族陣營很大程度決定了腳色生存與否。



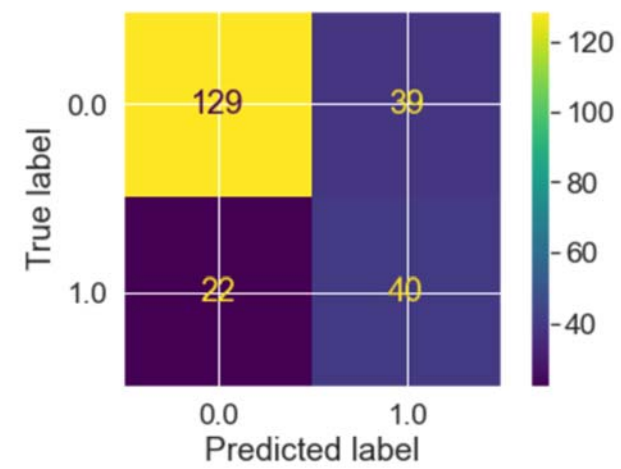
```
precision_2_depth= 0.4594594594594595
recall_2_depth= 0.8225806451612904
accuracy_2_depth= 0.691304347826087
-----
precision_3_depth= 0.5063291139240507
recall_3_depth= 0.6451612903225806
accuracy_3_depth= 0.7347826086956522
-----
precision_4_depth= 0.5
recall_4_depth= 0.2903225806451613
accuracy_4_depth= 0.7304347826086957
-----
precision_5_depth= 0.4878048780487805
recall_5_depth= 0.6451612903225806
accuracy_5_depth= 0.7217391304347827
-----
precision_6_depth= 0.5303030303030303
recall_6_depth= 0.5645161290322581
accuracy_6_depth= 0.7478260869565218
-----
precision_7_depth= 0.52
recall_7_depth= 0.6290322580645161
accuracy_7_depth= 0.7434782608695653
-----
precision_8_depth= 0.4675324675324675
recall_8_depth= 0.5806451612903226
accuracy_8_depth= 0.7086956521739131
-----
```

2. Confusion Matrix 的結果截圖

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import plot_confusion_matrix
confusion_matrix(y_test, Y_prediction)
```

```
[58]: array([[129, 39],
        [ 22, 40]], dtype=int64)
```

```
[59]: # 叫出圖式化confusion_matrix，確認 TP、TN、FN、FP
plot_confusion_matrix(clf, X_test, y_test)
plt.show()
```



3. 決策樹的圖

