

## Project 2\_Report

P76074389 蔡雨芝

(Coding with Python 3.6.5, macOS Mojave 10.14.1)

執行方式：python3 HW2.py (must in 'code' folder)

### Original Dataset: (Kaggle)Titanic: Machine Learning from Disaster

- 利用 Titanic Dataset 原有欄位內容更改 Survived / not Survived(Died)的結果。
- Original Data Columns: PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked
- 9 Features:
  - Survived: {'0(Died)': 0, '1(Survived)': 1}
  - Pclass: {1, 2, 3}
  - Sex: {'female': 0, 'male': 1}
  - Age: {'Age<=16':0, '16<Age<=32':1, '32<Age<=48':2, '48<Age<=64':3, 'Age>64': 4}
  - Fare: {'Fare<=7.91':0, '7.91<Fare<=14.454':1, '14.454<Fare<=31':2, 'Fare>31': 3}
  - Embarked: {'S':0, 'C':1, 'Q':2}
  - FamilySize = 'SibSp'+ 'Parch'+1
  - IsAlone: {'FamilySize>1':0, 'FamilySize=1':1}
  - Title: {"Mr":1, "Master":2, "Mrs":3, "Miss":4, "Rare":5}

### 1.Design Absolutely Right Rules

- Define died rules (else survived):

	PClass	Sex	Age	Fare	Embarked
■ (1)				0	
■ (2)	3	1			0
■ (3)				2	0
■ (4)	2	0	1		

(1) Fare=0

(2) PClass=3, Sex=1, Embarked=0

(3) Fare=2, Embarked=0

(4) PClass=2, Sex=0, Age=1

- Data 數量:

■ Training Data(891): Survived(368) / Died(523)

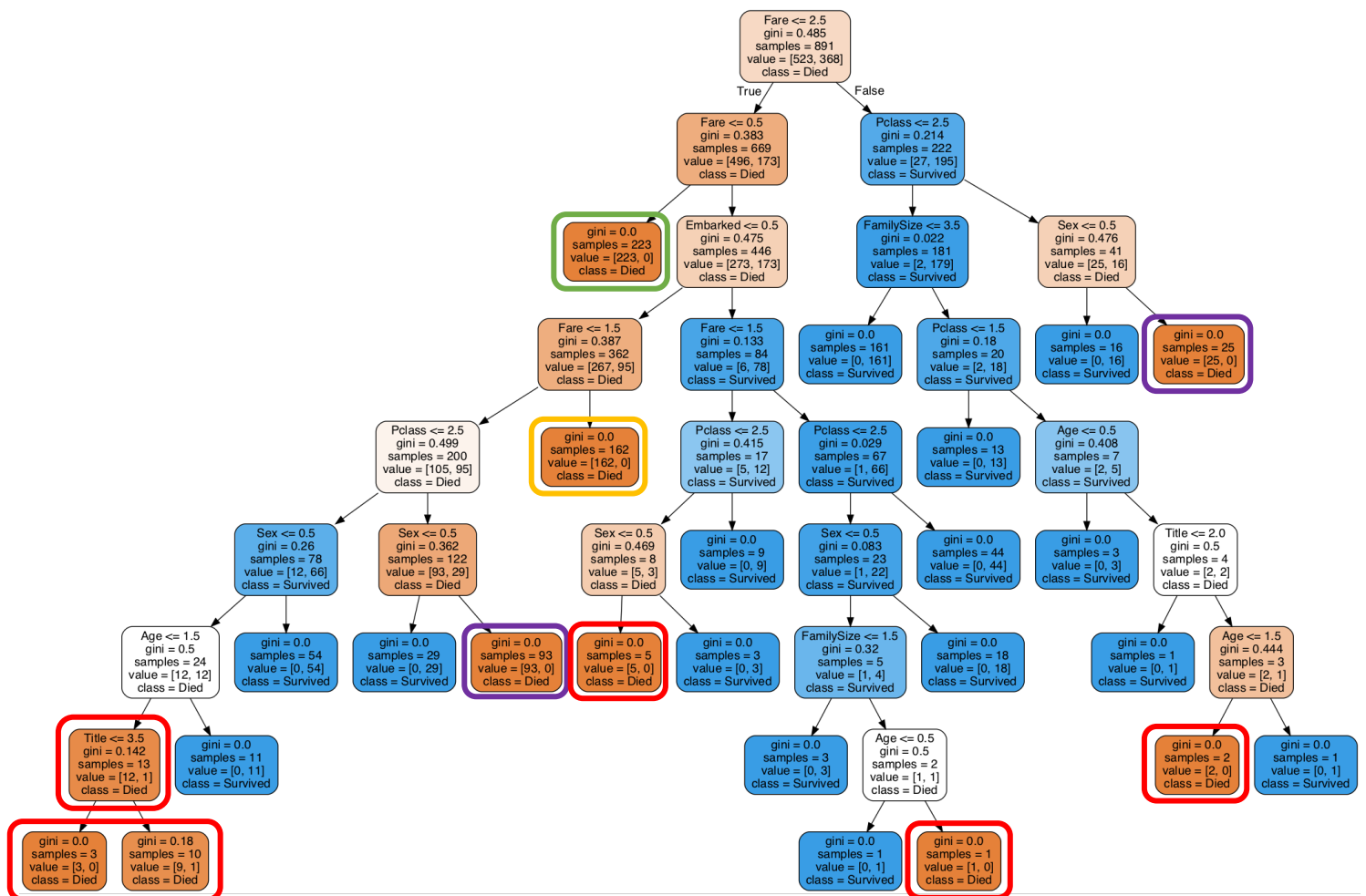
■ Test Data(418): Survived(185) / Died(233)

## 2. Compare with Decision Tree:

- Decision tree is generated with 'graphviz'.  
(使用安裝至系統的python3.6會有問題，後來改用Anaconda3 5.3方能順利編譯完成)
- Max\_depth 藉由 Cross Validation 的 Accuracy 結果來協助決定：
  - 利用KFold，split=10次的方式進行cross validation
  - max\_depth從1開始測試到max\_attributes(features的數量)

Max_depth	Avg Accuracy
1	0.775618
2	0.785693
3	0.884407
4	0.884407
5	0.940574
6	0.971948
7	0.987678
8	0.988801

- 因此 max\_depth 設定為 8，產生 decision tree:



- 上圖利用顏色區分對應的Rules：■=(1)，■=(2)，■=(3)，■=(4)。

- **Accuracy(Decision Tree): 0.9952153110047847**

- 討論：

- Absolutely Rules希望不要設計得太複雜（怕decision tree太複雜不易觀察），因為一開始設計得比較複雜(7 Rules)後發現整個decision tree很雜不易觀察，後來修改成4條Rules，方便觀察討論，也刻意不用同樣的features（共用5種features，每個rule使用1 ~ 3種不等），希望可以測試效果。
- 以training data來說，除了左下角第二個block有1人被錯誤分類之外，基本上都是可以正確分類的。左下角三個block的部分的錯誤分類可能原因為：Rule4設定的Age限制為'Age=1'，然而左下角倒數第二排的Block卻是'Age=0' or 'Age=1' 都可能存在的狀況。
- 因為decision tree的設計，在split時，best partition的情形可能與我建立的rules不同，所以才有同一個rule被分散在多個branches的狀況，其中以Rule4最為明顯（被分散在4個branches），Rule2則是被分散在2個branches。
- Predict的accuracy分數很高，可能是因為features都經過前處理分類，所以每個feature大致上類別也不多(2 ~ 5種，FamilySize除外)，且rules沒有設計得太複雜。經實驗，若rules增加至7條，Accuracy 分數大約會降至0.9左右（一開始設計7rules時的情形）。

### 3.Compare with SVM:

- Kernel: RBF

- SVC(C=1.0, cache\_size=200, class\_weight=None, coef0=0.0, decision\_function\_shape='ovr', degree=3, gamma='auto', kernel='rbf', max\_iter=-1, probability=False, random\_state=None, shrinking=True, tol=0.001, verbose=False)

- SVM Cross Validation:

```
fit_time: [0.01348495 0.011904 0.01039577 0.01253772 0.01304007
           0.01208425 0.01102209 0.01150608 0.01073003 0.01133227]
score_time: [0.00122595 0.00103498 0.00108624 0.0010891 0.00117493
             0.00098109 0.00141478 0.0011282 0.00097895 0.00118399]
test_score: [0.9 0.94444444 0.82222222 0.93258427 0.88764045
             0.93258427 0.88764045 0.87640449 0.86363636 0.92045455]
train_score: [0.92759051 0.92134831 0.91510612 0.92394015 0.93391521
              0.90773067 0.93017456 0.92394015 0.93399751 0.92777086]
```

- **Accuracy(SVM): 0.9282296650717703**

- 討論：

- SVM的accuracy較低，可能因為我並沒有特別測試、優化SVM的參數，不像decision tree有經過cross validation來修改max\_depth。

- SVM的cross validation accuracy結果相較於max\_depth=8的decision tree而言也算較低，甚至有到0.86（灰底標示）的情形，可能因此也造成最終的accuracy(SVM)較低。

#### 4.Compare with Neural Network (MLPClassifier) :

- MLPClassifier(solver='lbfgs', alpha=1e-5, hidden\_layer\_sizes=(10, 2), random\_state=1)
- NN Cross Validation:  
fit\_time:[0.10598779 0.11074567 0.108814    0.12226796 0.10256696  
          0.10102415 0.10120201 0.10838509 0.10668683 0.10283685]  
score\_time: [0.00036001 0.00064421 0.00036216 0.00105596 0.00034094  
             0.00033879 0.0003562    0.00056005 0.00036788 0.0004859 ]  
test\_score: [0.93333333 0.98888889 1.            1.            0.96629213  
             0.98876404 0.98876404 0.98876404 0.98863636 0.98863636]  
train\_score: [0.99625468 0.99500624 0.99750312 0.99376559 0.99750623  
              0.99376559 0.99501247 0.99376559 0.99003736 0.99377335]  
avg\_test\_score = 0.9832079189999998  
avg\_train\_score = 0.994639022
- Accuracy(Neural Network): 0.992822966507177
- 討論：
  - NN的accuracy較SVM高，可能是因為hidden\_layer\_sizes參數有經過測試、調整。
  - NN的cross validation accuracy結果相較於max\_depth=8的decision tree的average accuracy而言算高，然而predict的結果並沒有比decision tree好。