**Data Mining hw2 F76074402 張圜華**

開發環境: PYTHON

***Dataset***

使用 Kaggle(titanic/train.csv) 進行修改, 訂定新的規則, 使存亡結果不同

存亡規則:

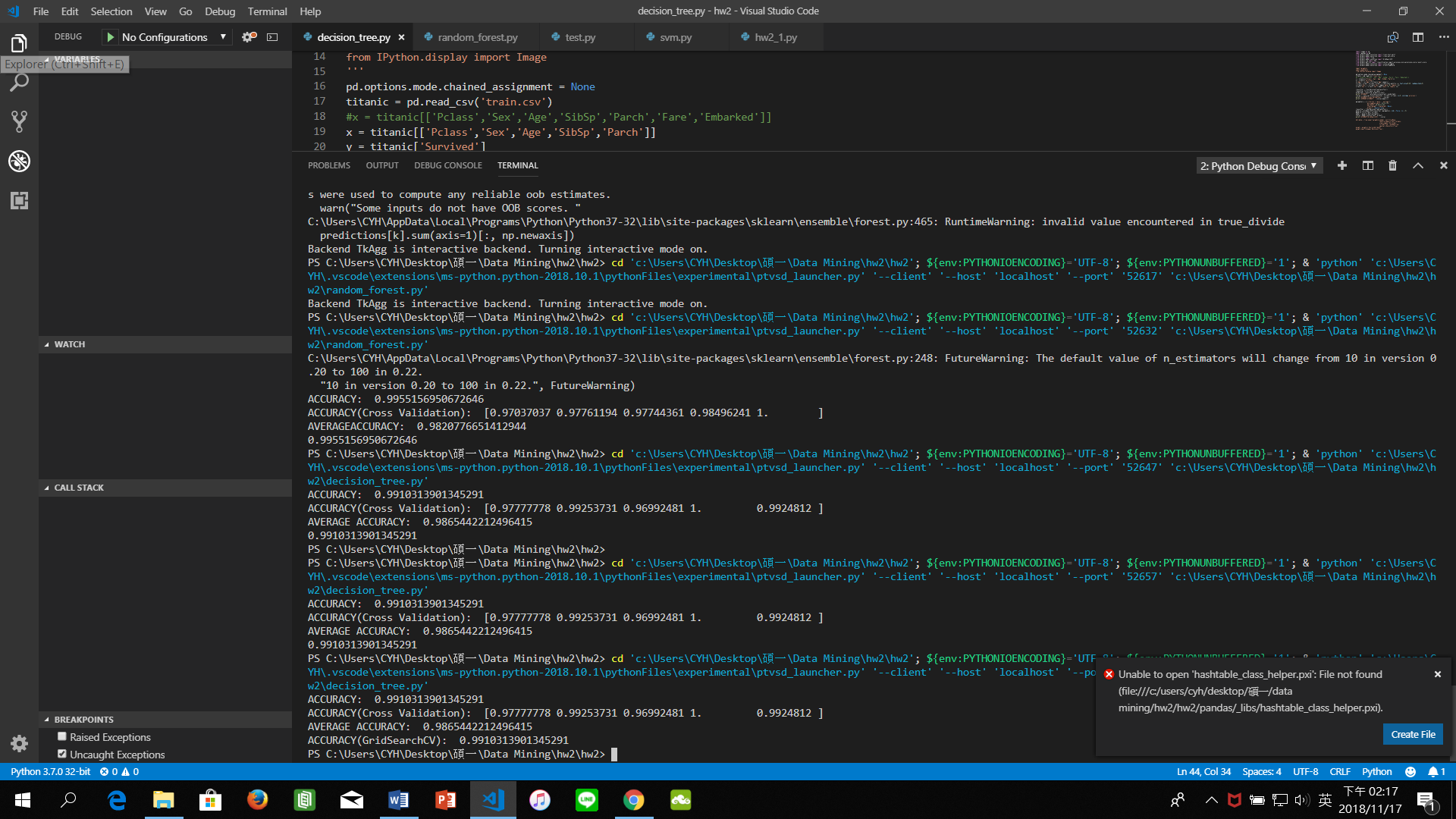
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Survived | Pclass | Sex | Age | SibSp | Parch |
| 0 | 3 |  | <=20 |  |  |
| 0 | 3 |  | >40 |  |  |
| 0 | 2 | Male | <=20 |  |  |
| 0 | 2 | Male | >40 |  |  |
| 0 | 1 | Male | <=10 |  |  |
| 0 | 1 | male | >50 |  |  |
| 0 |  |  | 40>=age>20 |  | >1 |
| 0 |  |  | >40 |  | 0 |
| 0 |  |  | <=20 | >1 |  |
| 1 | 不符合上面規則的, 都是可以存活的 | | | | |

Data Dictionary:

|  |  |  |
| --- | --- | --- |
| Variable | Definition | Key |
| survived | Survival | 0 = No, 1 = Yes |
| pclass | Ticket class | 1 = 1st, 2 = 2nd, 3 = 3rd |
| sex | Sex | Male, Female |
| Age | Age in years |  |
| sibsp | # of siblings / spouses aboard the Titanic |  |
| parch | # of parents / children aboard the Titanic |  |
| ticket | Ticket number |  |
| fare | Passenger fare |  |
| cabin | Cabin number |  |
| embarked | Port of Embarkation | C = Cherbourg,  Q = Queenstown,  S = Southampton |

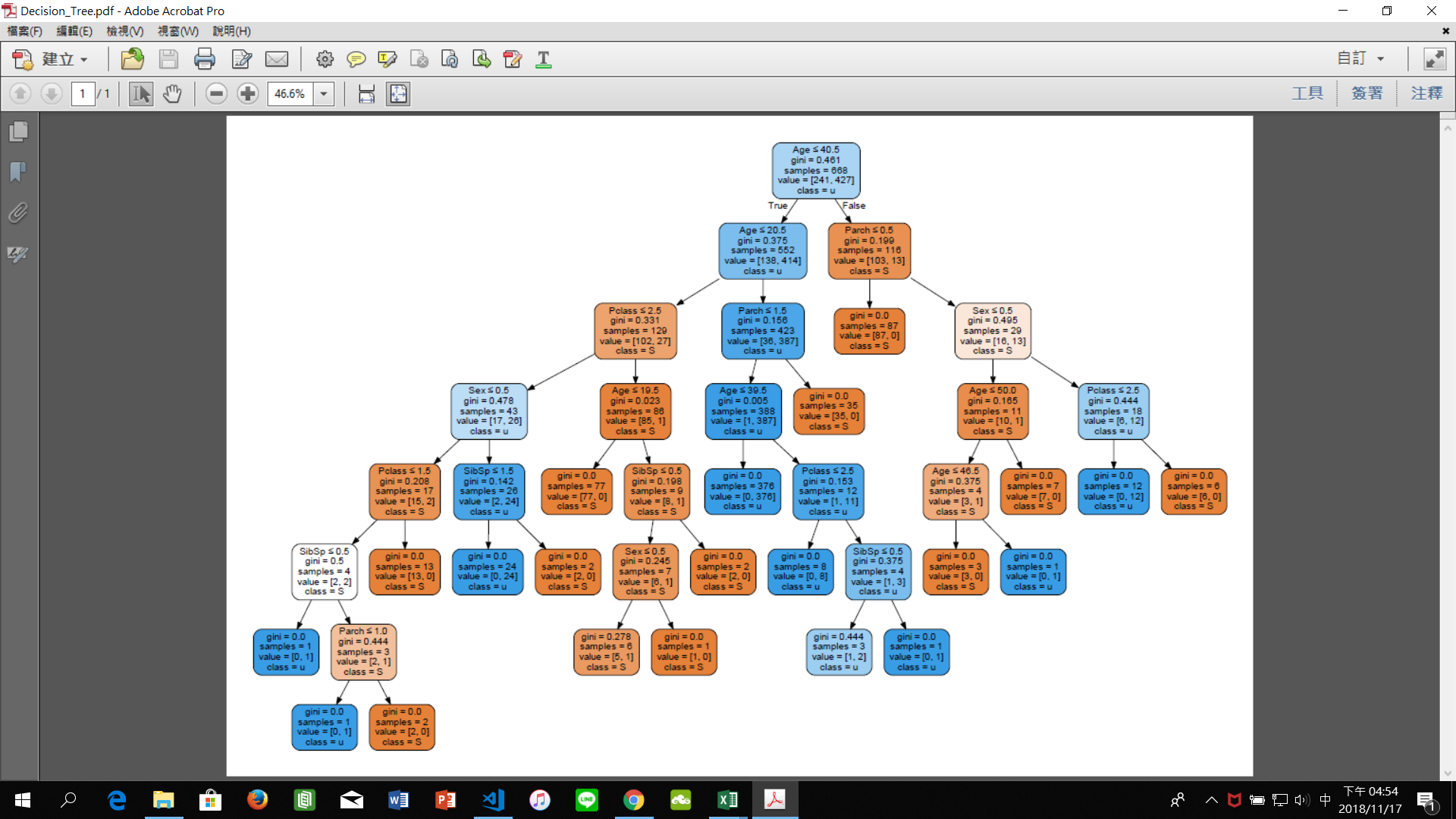
***Decision Tree***

結果:



* ACCURACY: 是使用預設參數的 accuracy 結果
* ACCURACY(CROSS Validation): 是使用交叉驗證五次後的 accuracy 結果
* cross\_val\_score(classifier, x\_train, y\_train, cv=5, scoring='accuracy')
* AVERAGE ACCURACY: 是交叉驗證後五次的平均 accuracy
* ACCURACY(GridSearchCV): 是使用GridSearchCV 函式, 對所設置的所有參數做所有的可能排列組合下去測, 然後取得一個最佳參數, 最後獲得的 accuracy結果
* classifier = tree.DecisionTreeClassifier()
* gsearch = GridSearchCV(classifier, parameters, iid = False, cv = 5)
* gsearch.fit(x\_train, y\_train)
* model = gsearch.best\_estimator\_
* score = model.score(x\_test, y\_test)

輸出圖:

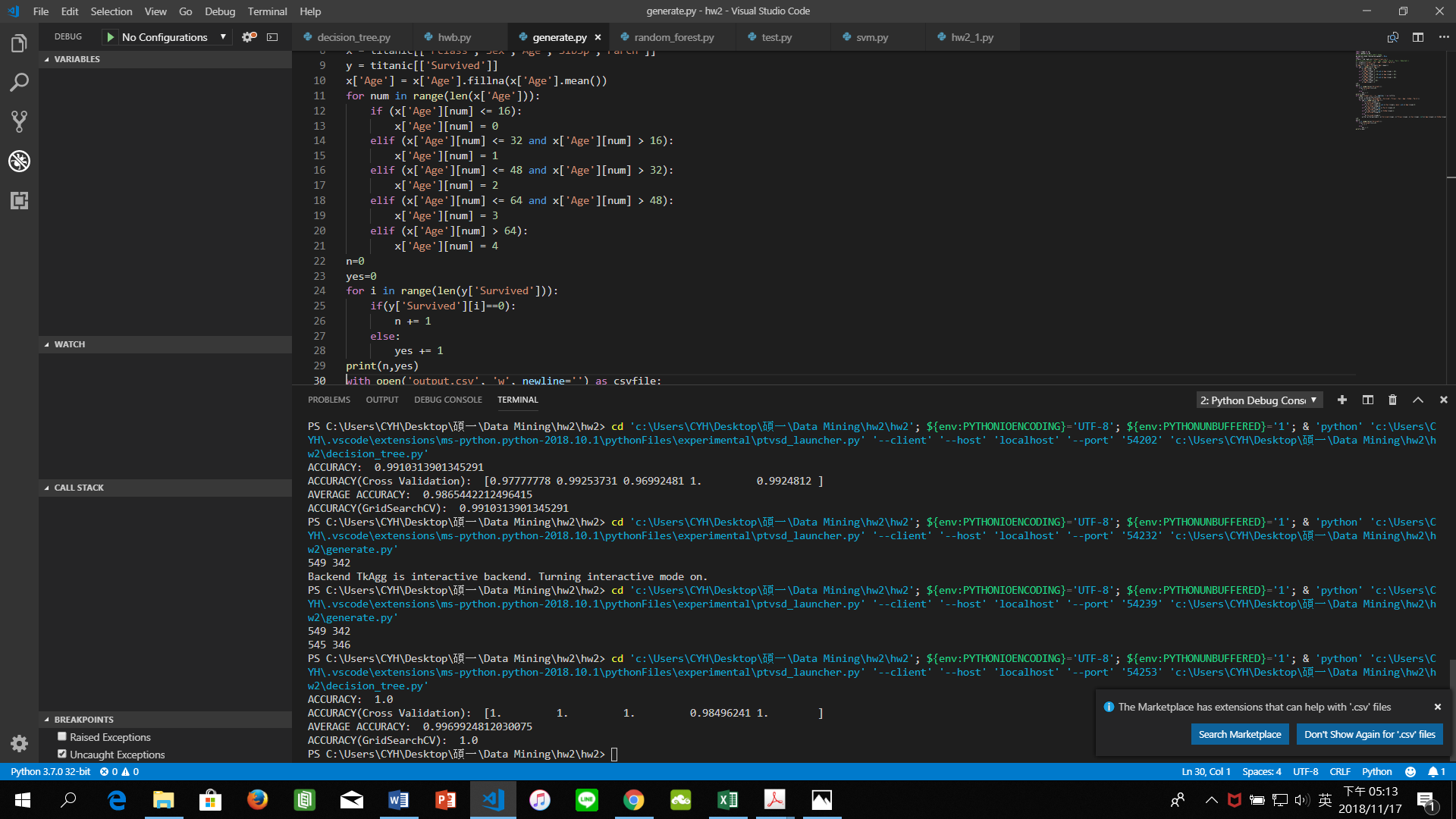


因為原先訂定的規則太多(總共有十個規則), 因此輸出的圖太過複雜, 不容易比較和觀察, 所以另外重新定義規則(共五個規則), 以此做討論與分析

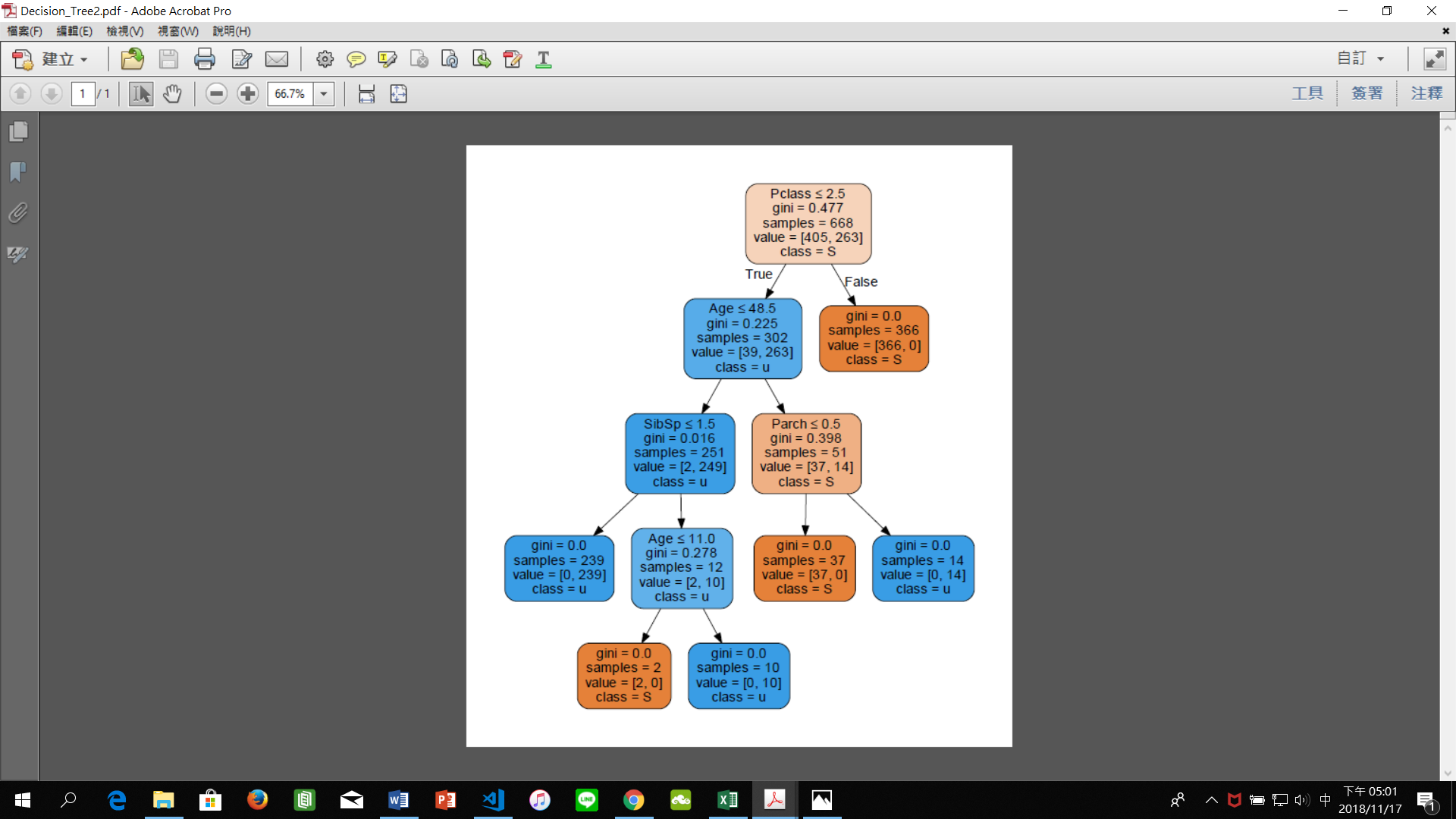
新的存亡規則:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Survived | Pclass | Sex | Age | SibSp | Parch |
| 0 | 3 |  |  |  |  |
| 0 | 2 | Male | >64 |  |  |
| 0 |  |  | >48 |  | 0 |
| 0 |  |  | <=16 | >1 |  |
| 1 | 不符合上面規則的, 都是可以存活的 | | | | |

結果:



輸出圖:



DecisionTree 所產生的規則如下:

1. pclass >2.5 (也就是 pclass = 3) 時, 會死亡

* 第一條規則

1. age > 48.5(也就是 age > 48) 且 parch <= 0.5(也就是 parch = 0) 時, 會死亡

* 第三條規則

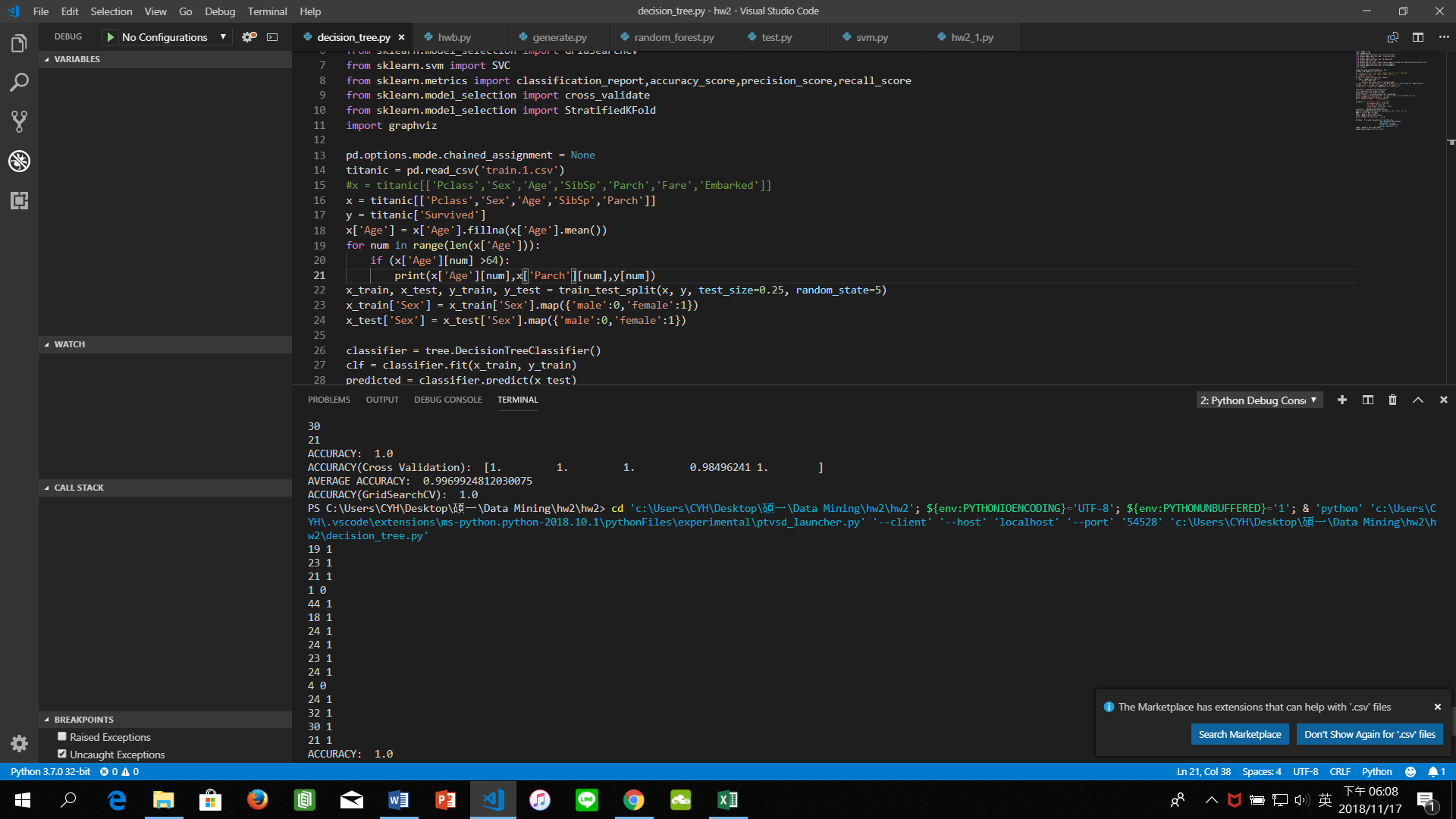
1. sibsp > 1.5(也就是 sibsp >1) 且 age <= 11時, 會死亡

* 第四條規則

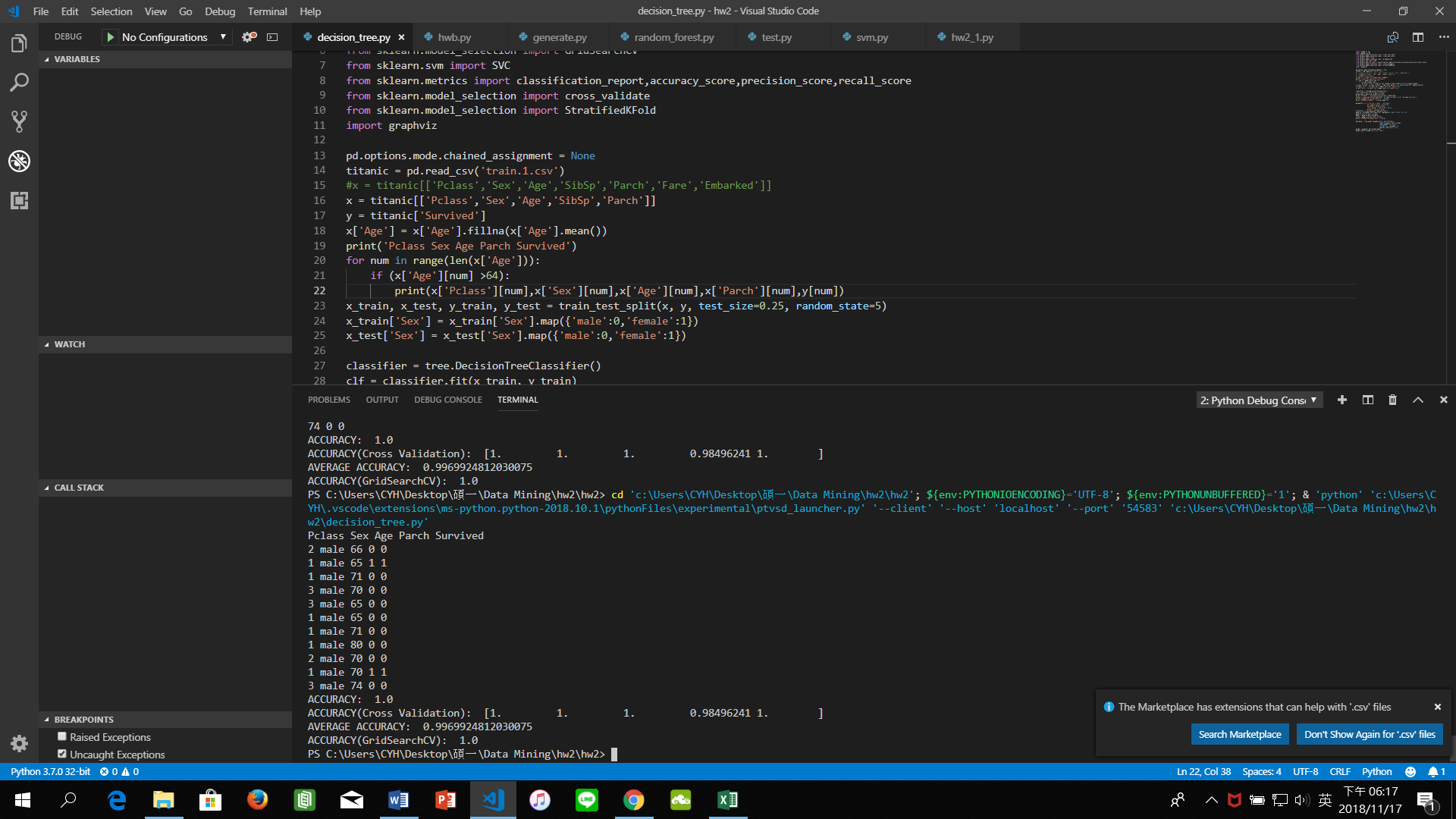
在第一點時,可以很明顯觀察到,完全符合absolutely right rules(自訂的規則)的第一條規則

在第二點時,也可以很明顯觀察到,完全符合absolutely right rules(自訂的規則)的第三條規則

在第三點時,可以看到(age < 11時, 會死亡), 這和原先訂定的規則不同(age < 16時, 會死亡), 這是因為在資料中符合pclass < 3 and age <= 48 and sibsp >1 的data中, 死亡的有年紀 1 和 4, 而存活的有年紀 18, 19, 21,…., 而(18+4)/2=11 , 所以取十一作為判定方法

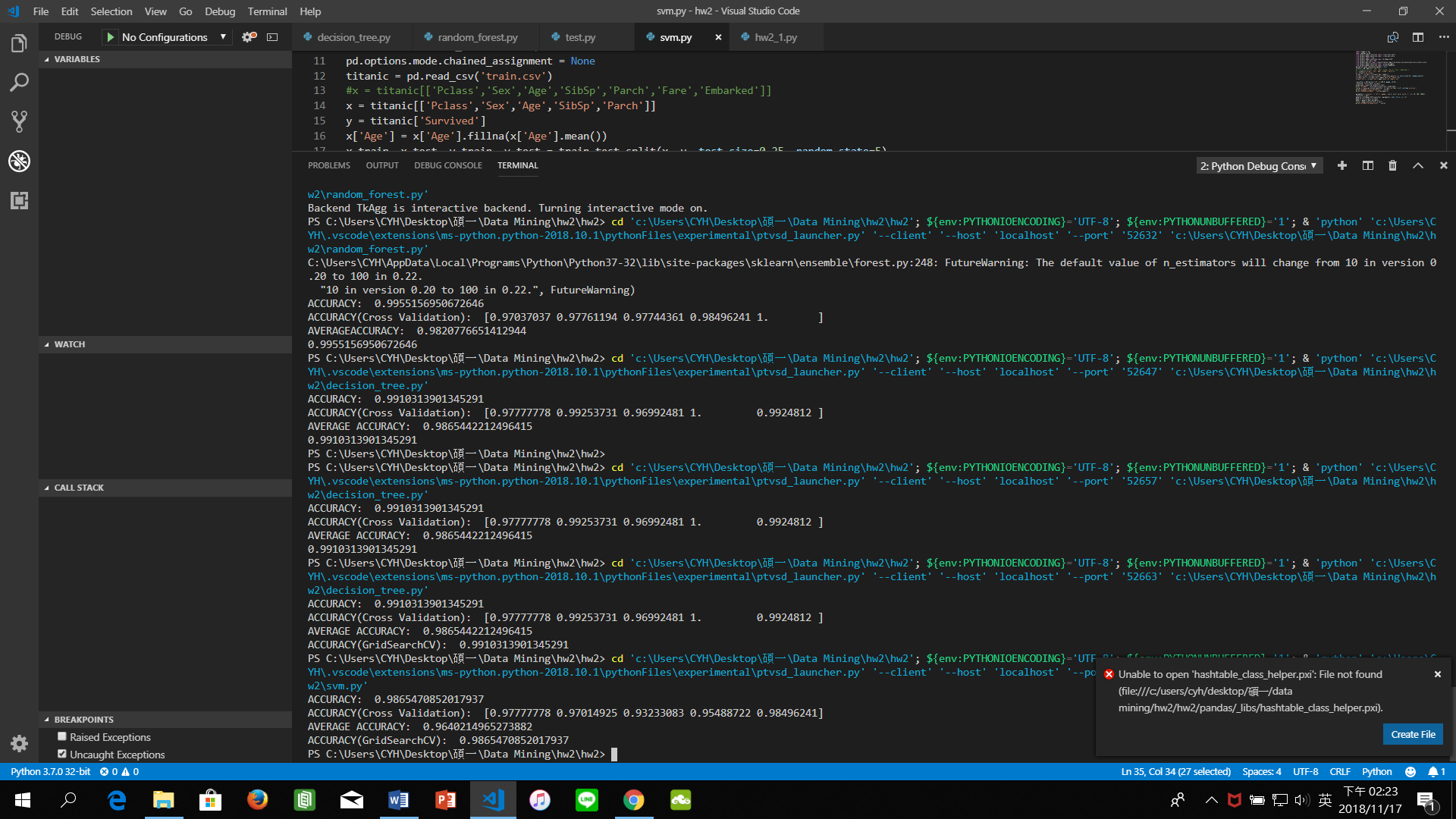


而發現沒有符合absolutely right rules(自訂的規則)的第二條規則, 這是因為在資料中所有符合第二條規則的都符合第三條規則,因此只要第三條規則就可以了,所以也可以發現我們可以藉此決策樹找到一些不需要的規則



***SVM***

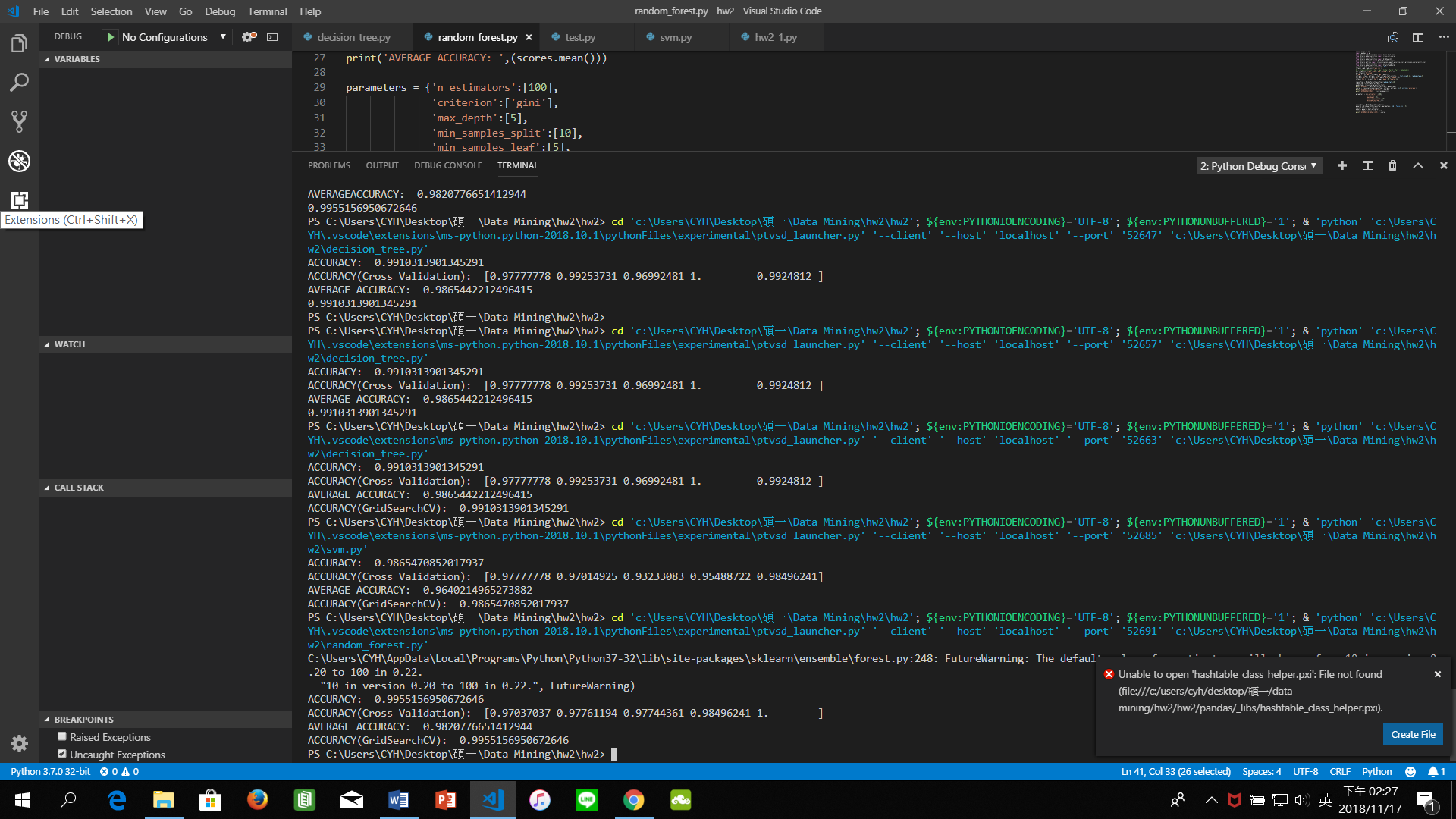
結果:



* ACCURACY: 是使用參數random\_state=10的 accuracy 結果
* ACCURACY(CROSS Validation): 是使用交叉驗證五次後的 accuracy 結果
* cross\_val\_score(classifier, x\_train, y\_train, cv=5, scoring='accuracy')
* AVERAGE ACCURACY: 是交叉驗證後五次的平均 accuracy
* ACCURACY(GridSearchCV): 是使用GridSearchCV 函式, 對所設置的所有參數做所有的可能排列組合下去測, 然後取得一個最佳參數, 最後獲得的 accuracy結果
* classifier = tree.DecisionTreeClassifier()
* gsearch = GridSearchCV(classifier, parameters, iid = False, cv = 5)
* gsearch.fit(x\_train, y\_train)
* model = gsearch.best\_estimator\_
* score = model.score(x\_test, y\_test)

***Random Forest***

結果:



* ACCURACY: 是使用參數kernel="rbf", C=100.0, gamma = 0.01的 accuracy 結果
* ACCURACY(CROSS Validation): 是使用交叉驗證五次後的 accuracy 結果
* cross\_val\_score(classifier, x\_train, y\_train, cv=5, scoring='accuracy')
* AVERAGE ACCURACY: 是交叉驗證後五次的平均 accuracy
* ACCURACY(GridSearchCV): 是使用GridSearchCV 函式, 對所設置的所有參數做所有的可能排列組合下去測, 然後取得一個最佳參數, 最後獲得的 accuracy結果
* classifier = tree.DecisionTreeClassifier()
* gsearch = GridSearchCV(classifier, parameters, iid = False, cv = 5)
* gsearch.fit(x\_train, y\_train)
* model = gsearch.best\_estimator\_
* score = model.score(x\_test, y\_test)

***Decision Tree VS SVM VS Random Forest***

ACCURACY 比較表:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Decision Tree | SVM | Random Forest |
| ACCURACY | 0.99103 | 0.98655 | 0.99552 |
| ACCURACY(Cross Validation) | 0.97778  0.99254  0.96992  1  0.99249 | 0.97778  0.97015  0.93233  0.95489  0.98496 | 0.97037  0.97761  0.97744  0.98496  1 |
| AVERAGE ACCURACY | 0.98654 | 0.96402 | 0.98208 |
| ACCURACY(GridSearchCV) | 0.99103 | 0.98655 | 0.99552 |

ACCURACY:

Random Forest > Decision Tree > SVM