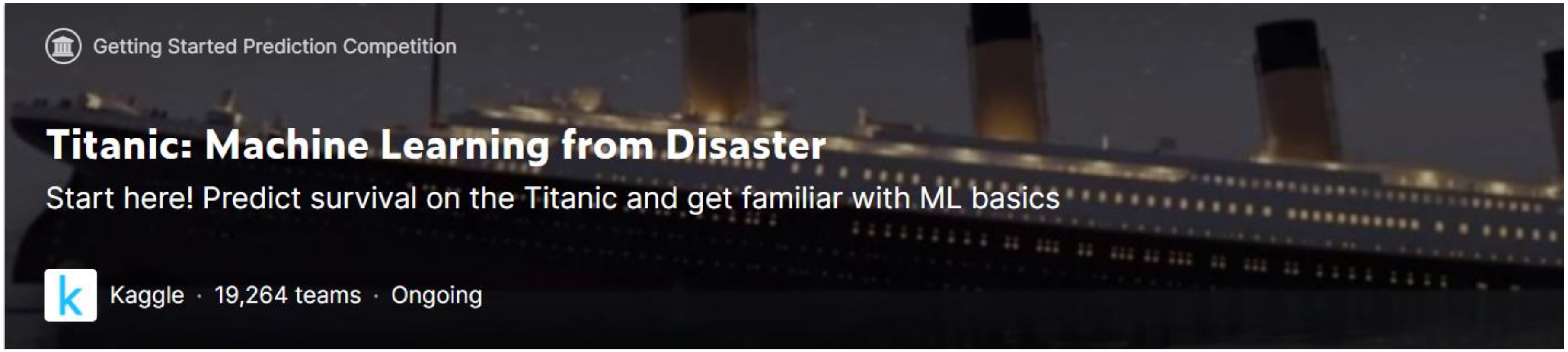
Titanic: Machine Learning from Disaster

徐永棚

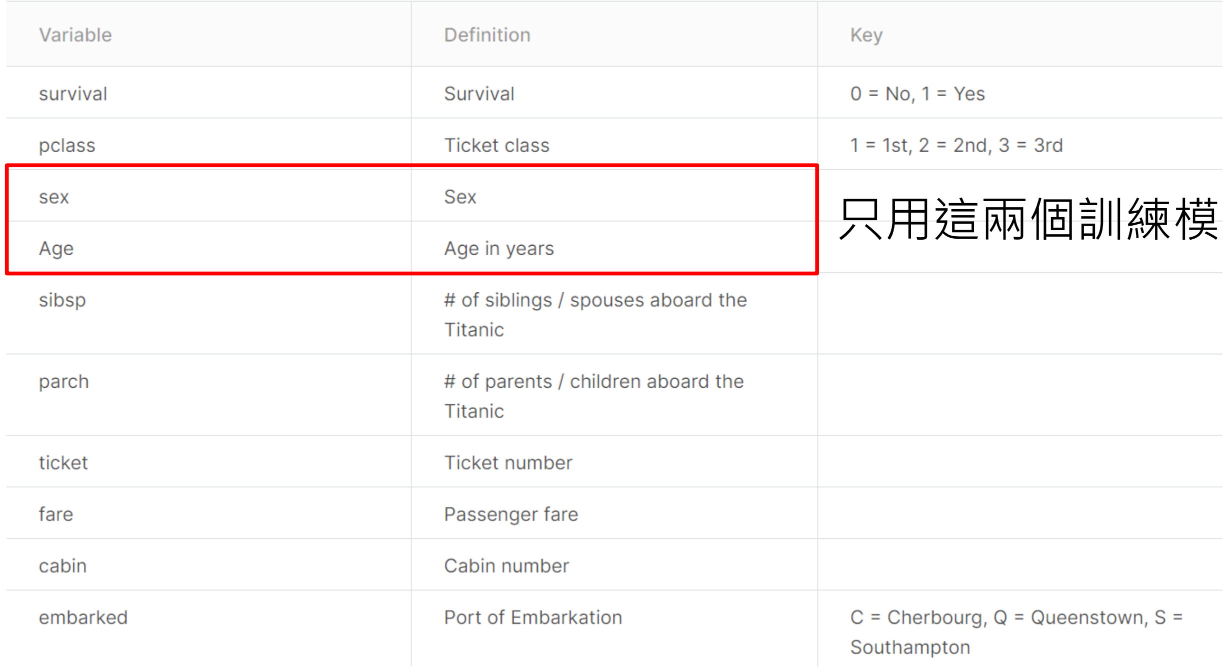


Getting Started Prediction Competition

Titanic: Machine Learning from Disaster

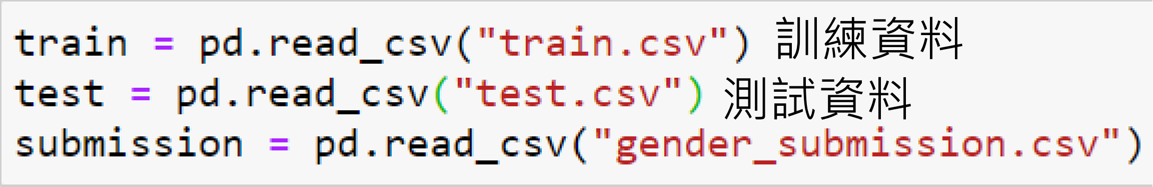
Start here! Predict survival on the Titanic and get familiar with ML basics

Kaggle · 19,264 teams · Ongoing



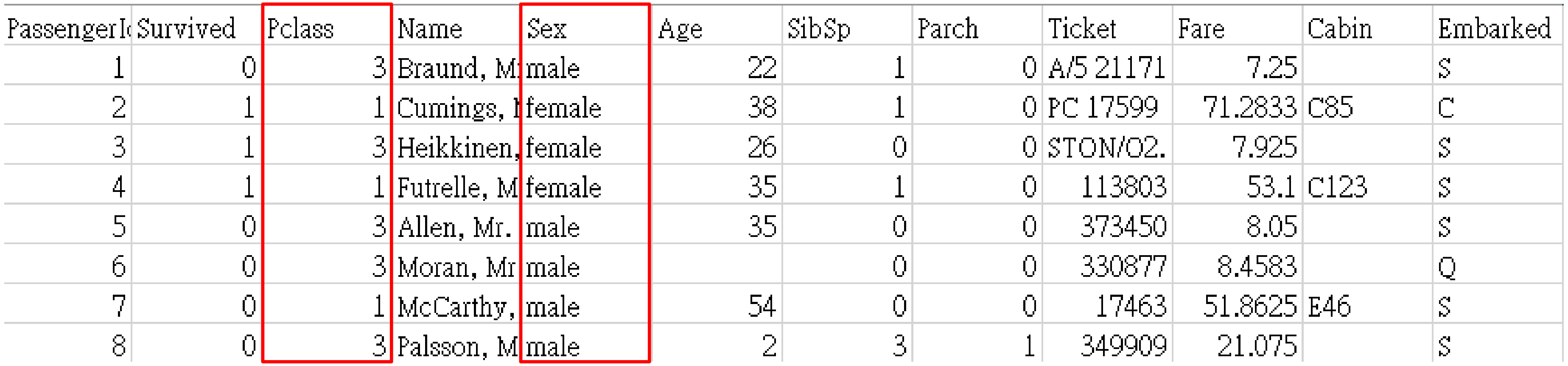
Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked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

只用這兩個訓練模型

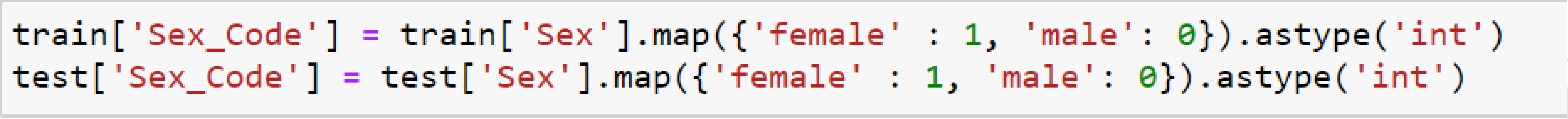


```
train = pd.read_csv("train.csv") 訓練資料
test = pd.read_csv("test.csv") 測試資料
submission = pd.read_csv("gender_submission.csv") 要的上傳檔案格式
```

前處理



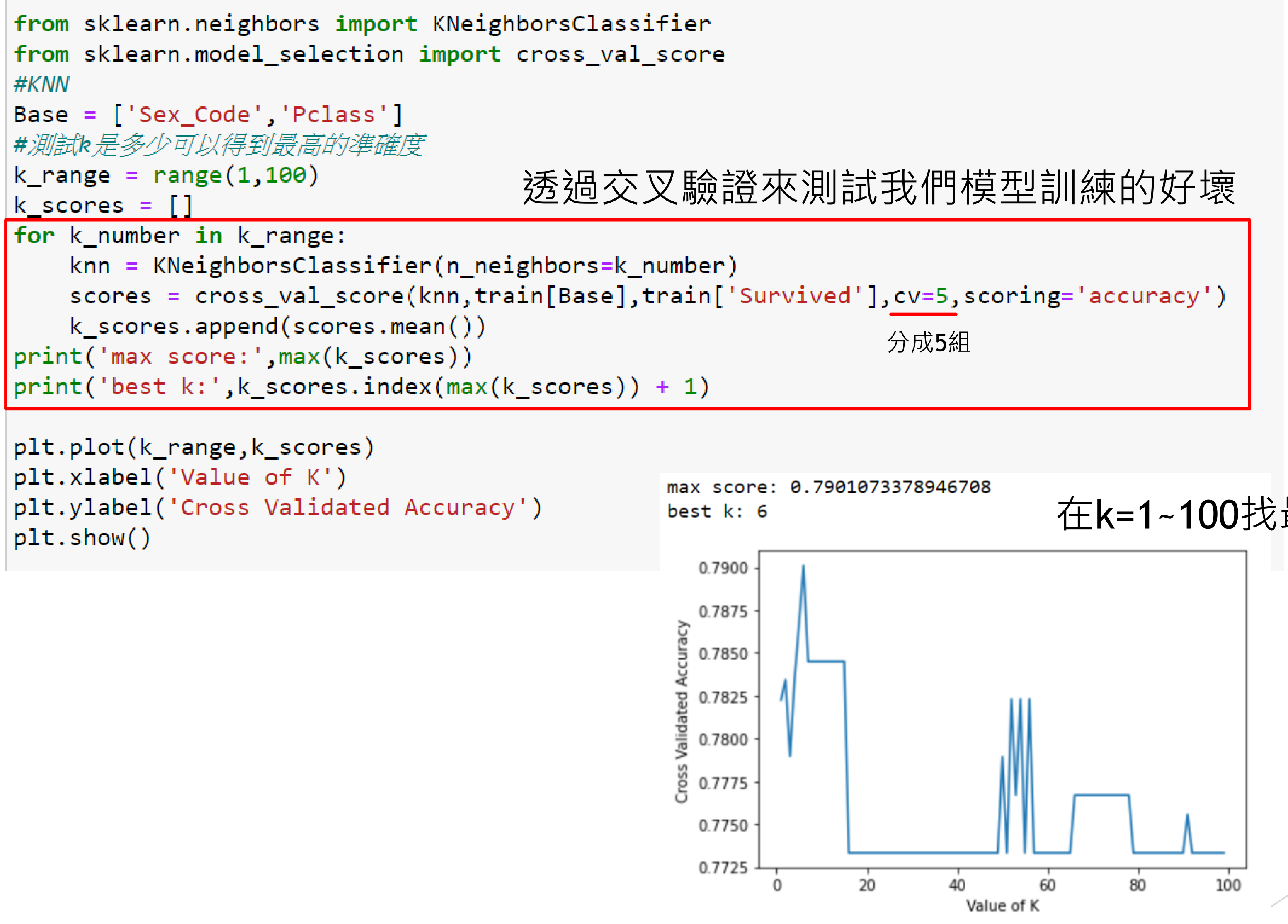
PassengerI	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, M	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, M	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen,	female	26	0	0	STON/O2.	7.925		S
4	1	1	Futrelle, M	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr.	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr	male		0	0	330877	8.4583		Q
7	0	1	McCarthy,	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, M	male	2	3	1	349909	21.075		S



```
train['Sex_Code'] = train['Sex'].map({'female' : 1, 'male': 0}).astype('int')
test['Sex_Code'] = test['Sex'].map({'female' : 1, 'male': 0}).astype('int')
```

新增一個欄位：Sex_Code
把female轉成1
把male轉成0

knn



```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
#KNN
Base = ['Sex_Code','Pclass']
#測試k是多少可以得到最高的準確度
k_range = range(1,100)
k_scores = []
for k_number in k_range:
    knn = KNeighborsClassifier(n_neighbors=k_number)
    scores = cross_val_score(knn,train[Base],train['Survived'],cv=5,scoring='accuracy')
    k_scores.append(scores.mean())
print('max score:',max(k_scores))
print('best k:',k_scores.index(max(k_scores)) + 1)

plt.plot(k_range,k_scores)
plt.xlabel('Value of K')
plt.ylabel('Cross Validated Accuracy')
plt.show()
```

透過交叉驗證來測試我們模型訓練的好壞

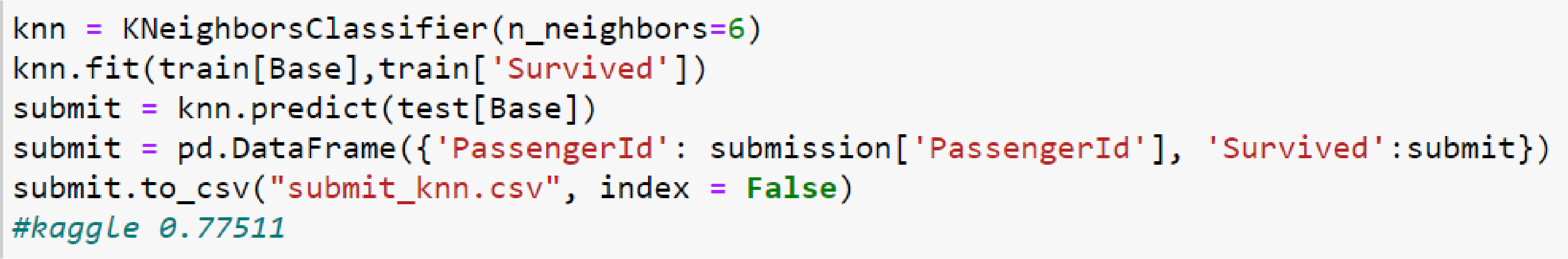
分成5組

```
max score: 0.7901073378946708
best k: 6
```

在k=1~100找最好的k

knn

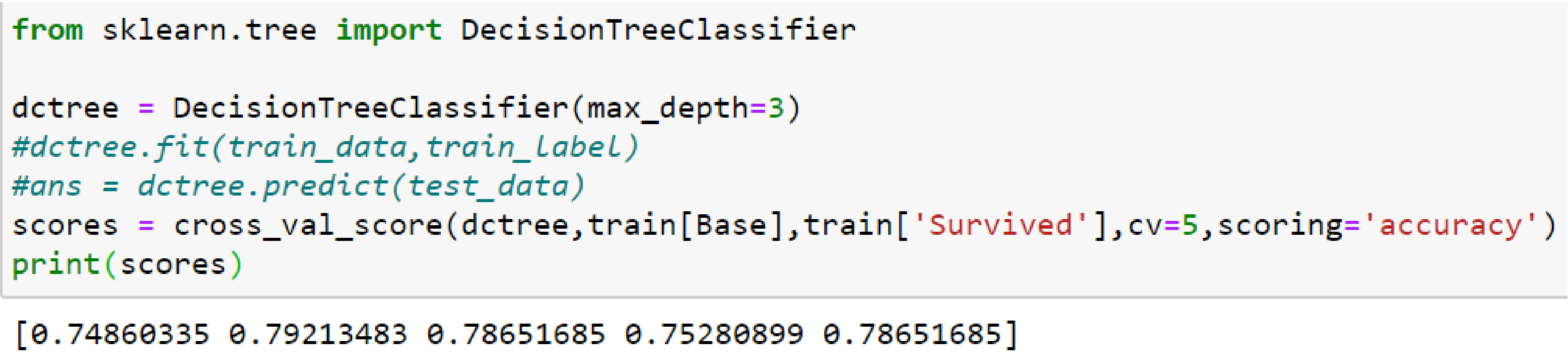
- 直接提交到kaggle上，準確率為0.77511



```
knn = KNeighborsClassifier(n_neighbors=6)
knn.fit(train[Base],train['Survived'])
submit = knn.predict(test[Base])
submit = pd.DataFrame({'PassengerId': submission['PassengerId'], 'Survived':submit})
submit.to_csv("submit_knn.csv", index = False)
#kaggle 0.77511
```

Decision tree

- 一樣用cross validation，大概看一下準確率

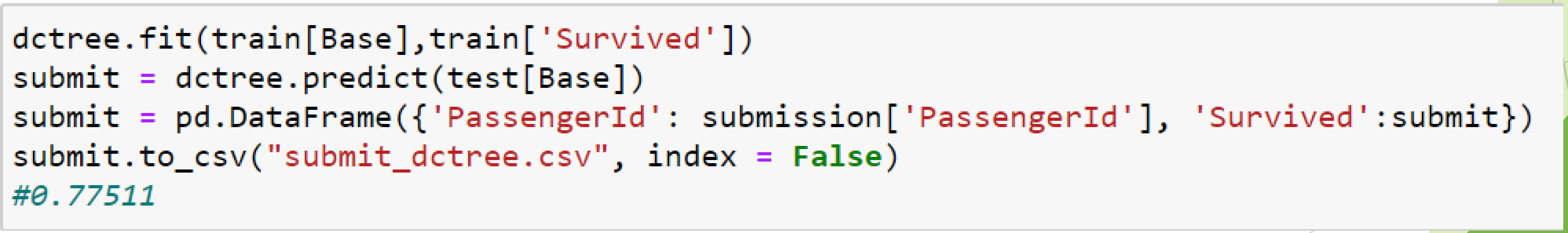


```
from sklearn.tree import DecisionTreeClassifier

dctree = DecisionTreeClassifier(max_depth=3)
#dctree.fit(train_data,train_label)
#ans = dctree.predict(test_data)
scores = cross_val_score(dctree,train[Base],train['Survived'],cv=5,scoring='accuracy')
print(scores)
```

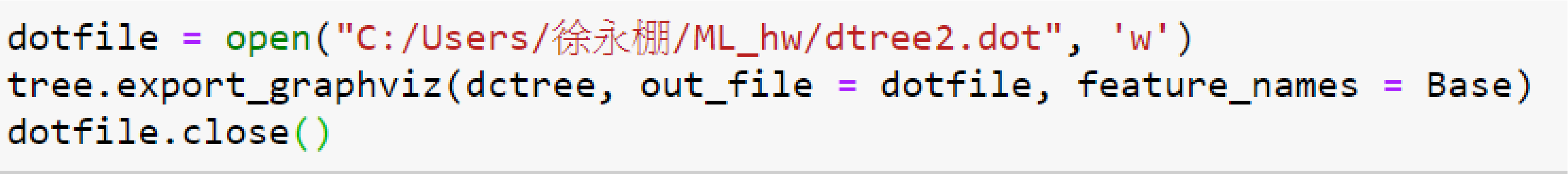
```
[0.74860335 0.79213483 0.78651685 0.75280899 0.78651685]
```

- 直接上傳到kaggle，準確率為0.77511

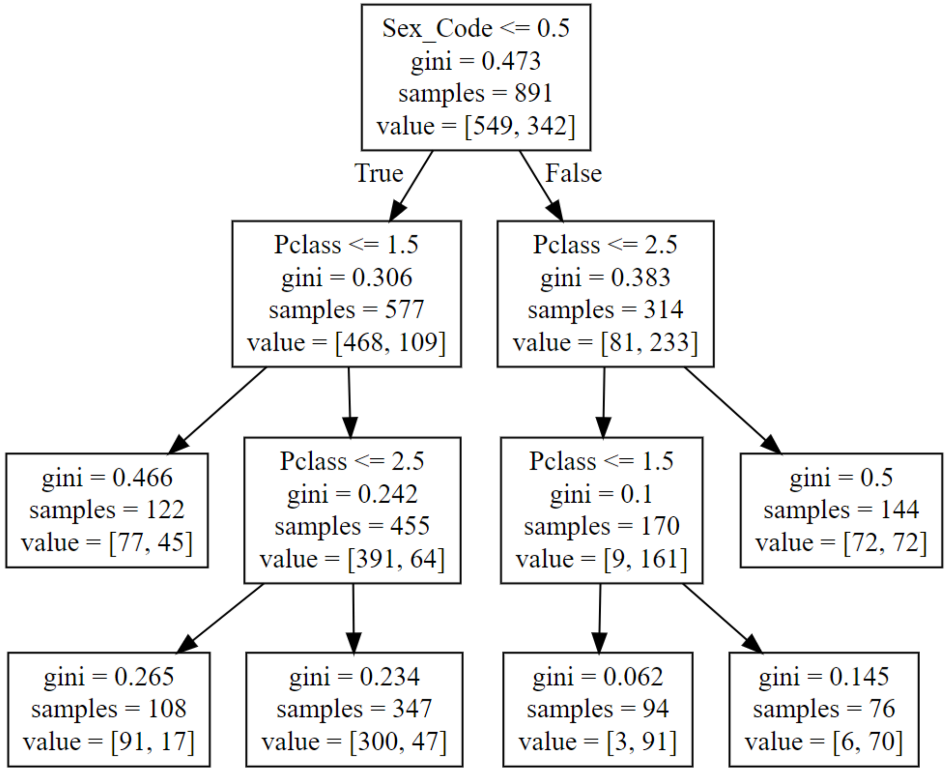


```
dctree.fit(train[Base],train['Survived'])
submit = dctree.predict(test[Base])
submit = pd.DataFrame({'PassengerId': submission['PassengerId'], 'Survived':submit})
submit.to_csv("submit_dctree.csv", index = False)
#0.77511
```

Decision tree



```
dotfile = open("C:/Users/徐永棚/ML_hw/dtree2.dot", 'w')
tree.export_graphviz(dctree, out_file = dotfile, feature_names = Base)
dotfile.close()
```



Sex_Code <= 0.5
gini = 0.473
samples = 891
value = [549, 342]

True

False

Pclass <= 1.5
gini = 0.306
samples = 577
value = [468, 109]

Pclass <= 2.5
gini = 0.383
samples = 314
value = [81, 233]

gini = 0.466
samples = 122
value = [77, 45]

Pclass <= 2.5
gini = 0.242
samples = 455
value = [391, 64]

Pclass <= 1.5
gini = 0.1
samples = 170
value = [9, 161]

gini = 0.5
samples = 144
value = [72, 72]

gini = 0.265
samples = 108
value = [91, 17]

gini = 0.234
samples = 347
value = [300, 47]

gini = 0.062
samples = 94
value = [3, 91]

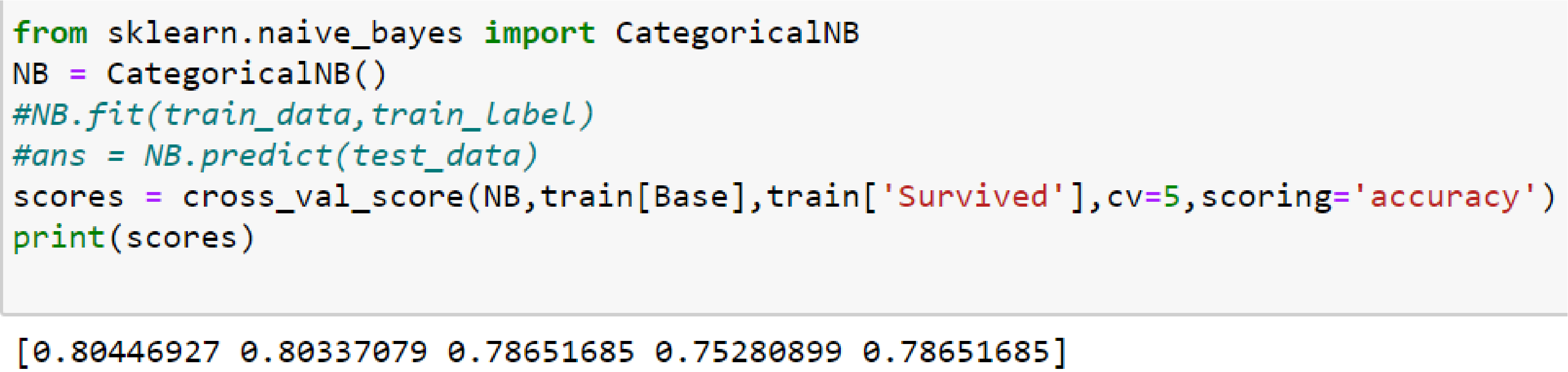
gini = 0.145
samples = 76
value = [6, 70]

將tree視覺化：
Sex_Code：0、1
Pclass：1、2、3

WebGraphviz is Graphviz in the Browser: http://webgraphviz.com/

Naïve Bayes

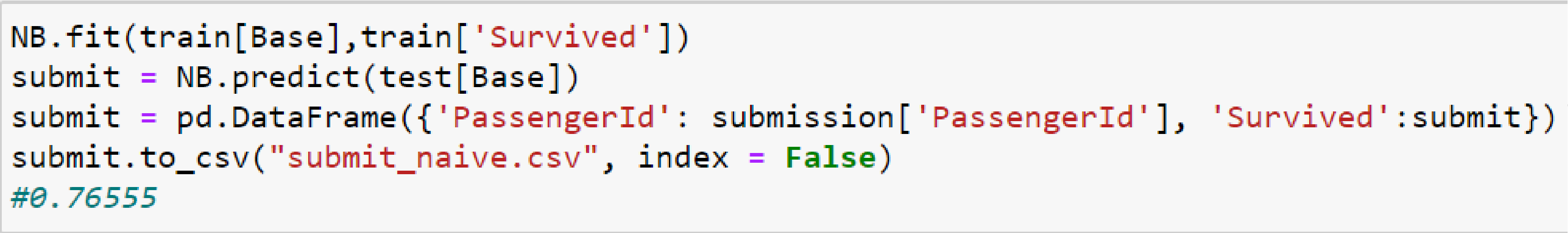
- 一樣用cross validation，大概看一下準確率



```
from sklearn.naive_bayes import CategoricalNB
NB = CategoricalNB()
#NB.fit(train_data,train_label)
#ans = NB.predict(test_data)
scores = cross_val_score(NB,train[Base],train['Survived'],cv=5,scoring='accuracy')
print(scores)
```

```
[0.80446927 0.80337079 0.78651685 0.75280899 0.78651685]
```

- 直接上傳到kaggle，準確率為0.76555



```
NB.fit(train[Base],train['Survived'])
submit = NB.predict(test[Base])
submit = pd.DataFrame({'PassengerId': submission['PassengerId'], 'Survived':submit})
submit.to_csv("submit_naive.csv", index = False)
#0.76555
```