【Lab1】Kaggle Titanic - Machine Learning from Disaster

Deep 柴

1. 資料前處理
   A. 將 Unique 的資料行去掉: PassengerId、Name、Ticket
   B. 處理NA的資料

      右方圖片說明有NA資料的資料欄位，依照不同狀況做處理

      i. Age: 約20% 的資料有NA，數量不算多、且為數值型態，所以用中位數做填補
      ii. Cabin: 約77%的資料是NA，所以直接Drop掉這一個Feature
      iii. Embarked: 有NA的資料很少(兩筆)，所以直接刪去有NA的Row
      iv. 女性存活之機率高達74%，而男性存活率只有18%，相差甚鉅。

```
Survived     0.000000
Pclass       0.000000
Sex          0.000000
Age         19.865320
SibSp        0.000000
Parch        0.000000
Fare         0.000000
Cabin       77.104377
Embarked     0.224467
dtype: float64
```

   C. 將非數值Feature進行One-hot encoding
      i. Sex: 拆成Sex_female、Sex_male
      ii. Embarked: 拆成 Embarked_C、Embarked_Q、Embarked_S
   D. StandardScaler
      i. 進行標準化

2. 程式碼: 使用 Jupyter Notebook 運行，下圖是模型的訓練程式碼

```
kf = KFold(n_splits=5)
kf.get_n_splits(X)

ind_round = 1

classifier = keras.Sequential()

classifier.add(Dense(units = 3,activation = 'sigmoid', input_dim = 10))

classifier.add(Dense(units = 2, activation = 'sigmoid'))

classifier.add(Dense(units = 1, activation = 'sigmoid'))

classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

for train_index, test_index in kf.split(X):

    print("ROUND: "+str(ind_round))

    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    classifier.fit(X_train, y_train,  epochs = 100)

    classifier.evaluate(X_test, y_test)
    #print(classifier.predict(X_test))

    ind_round += 1
```

A. 使用三層Dense Layer搭建模型，測試結果最好的是三層都使用sigmoid作為activation function

3. 模型訓練: K-Fold、嘗試不同 Threshold

A. K-Fold: K = 5

B. Threshold: 用 Excel 進行嘗試

如下圖，將輸出的機率以0.3~0.9的Threshold嘗試，最後的最佳結果是0.7

	A	B	C	D	E	F	G	H
1	PassengerI	Survived	0.3	0.4	0.5	0.6	0.7	0.9
2	892	0.108534	0	0	0	0	0	0
3	893	0.453848	1	1	0	0	0	0
4	894	0.128343	0	0	0	0	0	0
5	895	0.109582	0	0	0	0	0	0
6	896	0.366424	1	0	0	0	0	0
7	897	0.134974	0	0	0	0	0	0
8	898	0.685717	1	1	1	1	0	0
9	899	0.13159	0	0	0	0	0	0
10	900	0.5747	1	1	1	0	0	0
11	901	0.108451	0	0	0	0	0	0
12	902	0.109581	0	0	0	0	0	0

4. 結果