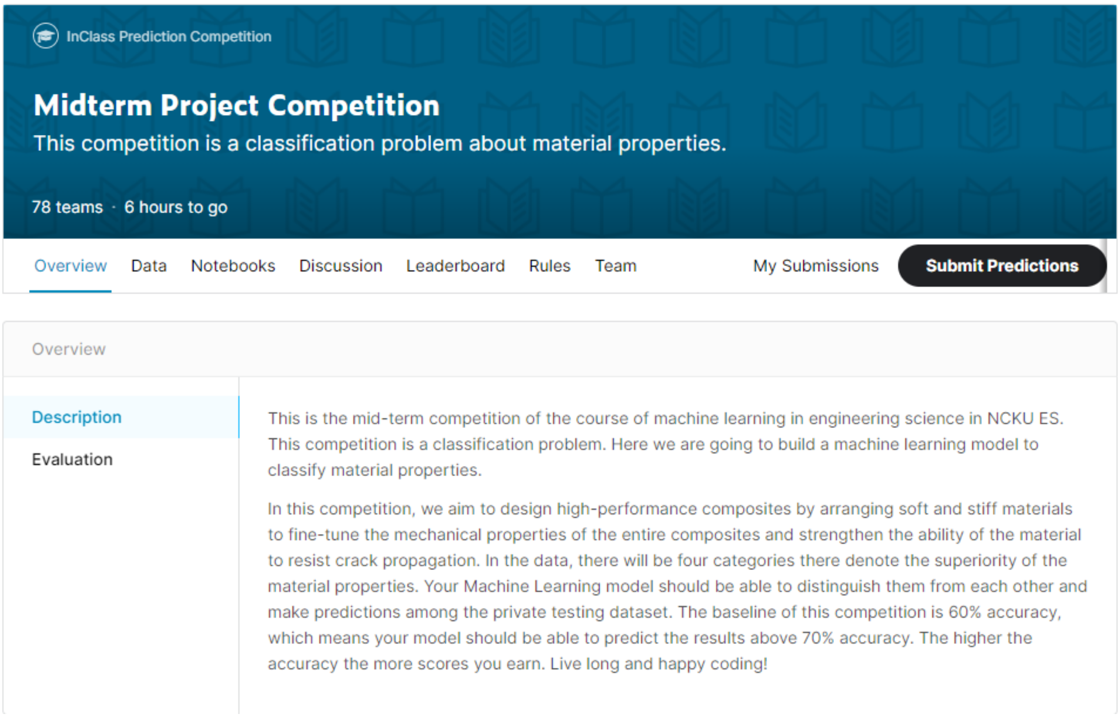
Midterm

Midterm Project Competition

N96094196 張維峻

工程科學系

0.摘要



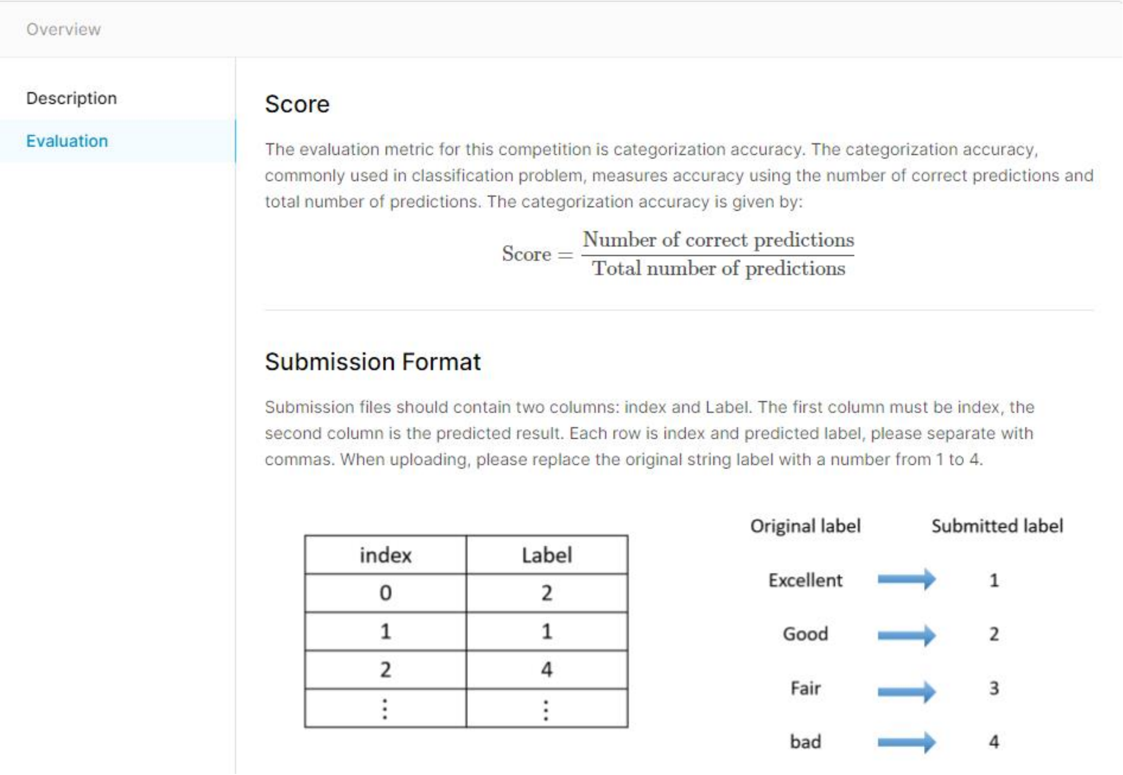
Midterm Project Competition

This competition is a classification problem about material properties.

78 teams · 6 hours to go

This is the mid-term competition of the course of machine learning in engineering science in NCKU ES. This competition is a classification problem. Here we are going to build a machine learning model to classify material properties.

In this competition, we aim to design high-performance composites by arranging soft and stiff materials to fine-tune the mechanical properties of the entire composites and strengthen the ability of the material to resist crack propagation. In the data, there will be four categories there denote the superiority of the material properties. Your Machine Learning model should be able to distinguish them from each other and make predictions among the private testing dataset. The baseline of this competition is 60% accuracy, which means your model should be able to predict the results above 70% accuracy. The higher the accuracy the more scores you earn. Live long and happy coding!



Score

The evaluation metric for this competition is categorization accuracy. The categorization accuracy, commonly used in classification problem, measures accuracy using the number of correct predictions and total number of predictions. The categorization accuracy is given by:

$$\text{Score} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Submission Format

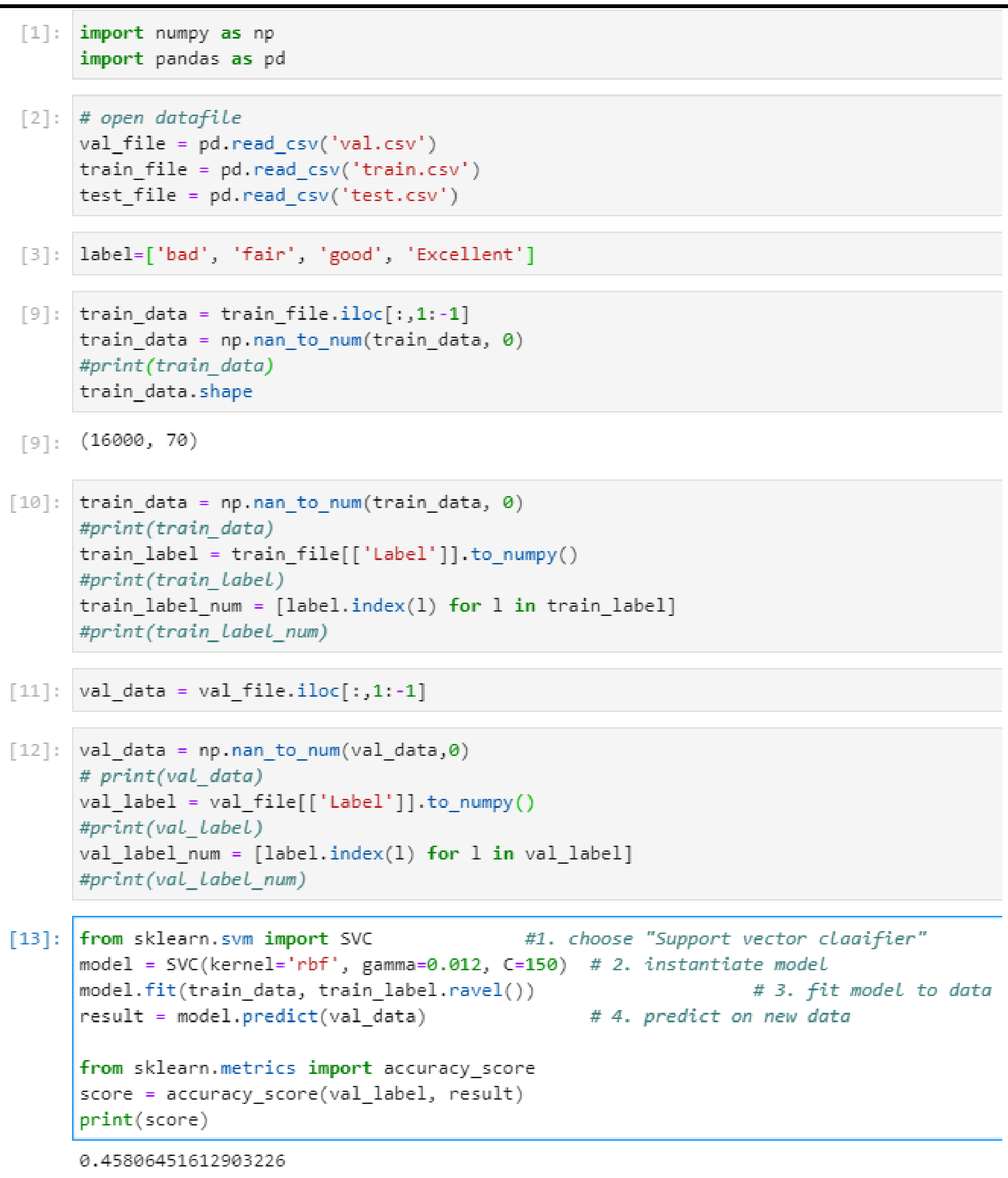
Submission files should contain two columns: index and Label. The first column must be index, the second column is the predicted result. Each row is index and predicted label, please separate with commas. When uploading, please replace the original string label with a number from 1 to 4.

index	Label
0	2
1	1
2	4
⋮	⋮

Original label		Submitted label
Excellent	→	1
Good	→	2
Fair	→	3
bad	→	4

1.選擇方法-SVM

選擇使用 SVM 來訓練分類器,讀取測試資料,把所以特徵都丟進去訓練

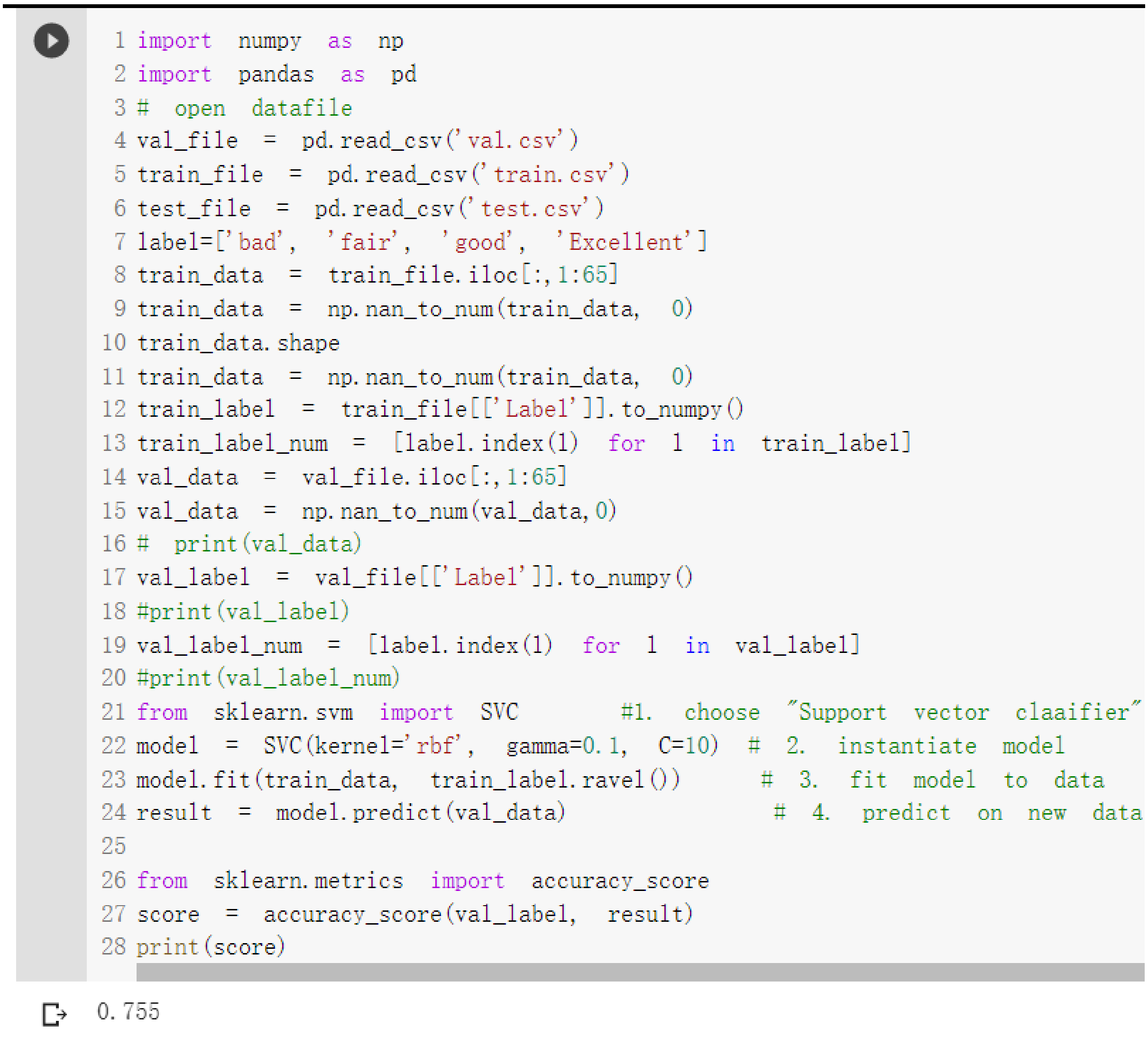


```
[1]:  import numpy as np
      import pandas as pd

[2]:  # open datafile
      val_file = pd.read_csv('val.csv')
      train_file = pd.read_csv('train.csv')
      test_file = pd.read_csv('test.csv')

[3]:  label=['bad', 'fair', 'good', 'Excellent']

[9]:  train_data = train_file.iloc[:,1:-1]
      train_data = np.nan_to_num(train_data, 0)
      #print(train_data)
      train_data.shape

[9]:  (16000, 70)

[10]: train_data = np.nan_to_num(train_data, 0)
      #print(train_data)
      train_label = train_file[['Label']].to_numpy()
      #print(train_label)
      train_label_num = [label.index(l) for l in train_label]
      #print(train_label_num)

[11]: val_data = val_file.iloc[:,1:-1]

[12]: val_data = np.nan_to_num(val_data,0)
      # print(val_data)
      val_label = val_file[['Label']].to_numpy()
      #print(val_label)
      val_label_num = [label.index(l) for l in val_label]
      #print(val_label_num)

[13]: from sklearn.svm import SVC              #1. choose "Support vector claaifier"
      model = SVC(kernel='rbf', gamma=0.012, C=150)  # 2. instantiate model
      model.fit(train_data, train_label.ravel())                    # 3. fit model to data
      result = model.predict(val_data)                # 4. predict on new data

      from sklearn.metrics import accuracy_score
      score = accuracy_score(val_label, result)
      print(score)

      0.458064516129032 26
```

模型準確度:0.45806

2.選擇特徵參數

先去除訓練集中,後面一些材料特性參數,以前面 8*8=64 個點構成的圖去訓練模型



```
import numpy as np
import pandas as pd
# open datafile
val_file = pd.read_csv(’val.csv’)
train_file = pd.read_csv(’train.csv’)
test_file = pd.read_csv(’test.csv’)
label=[’bad’, ’fair’, ’good’, ’Excellent’]
train_data = train_file.iloc[:,1:65]
train_data = np.nan_to_num(train_data, 0)
train_data.shape
train_data = np.nan_to_num(train_data, 0)
train_label = train_file[[’Label’]].to_numpy()
train_label_num = [label.index(l) for l in train_label]
val_data = val_file.iloc[:,1:65]
val_data = np.nan_to_num(val_data,0)
# print(val_data)
val_label = val_file[[’Label’]].to_numpy()
#print(val_label)
val_label_num = [label.index(l) for l in val_label]
#print(val_label_num)
from sklearn.svm import SVC        #1. choose ″Support vector claaifier″
model = SVC(kernel=’rbf’, gamma=0.1, C=10) # 2. instantiate model
model.fit(train_data, train_label.ravel())     # 3. fit model to data
result = model.predict(val_data)               # 4. predict on new data

from sklearn.metrics import accuracy_score
score = accuracy_score(val_label, result)
print(score)
```

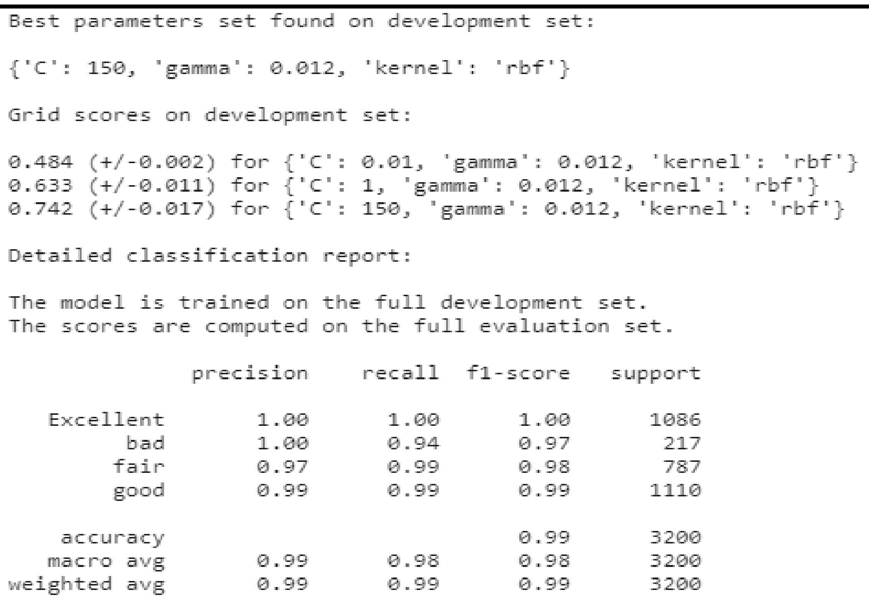
```
0.755
```

模型準確度:0.755

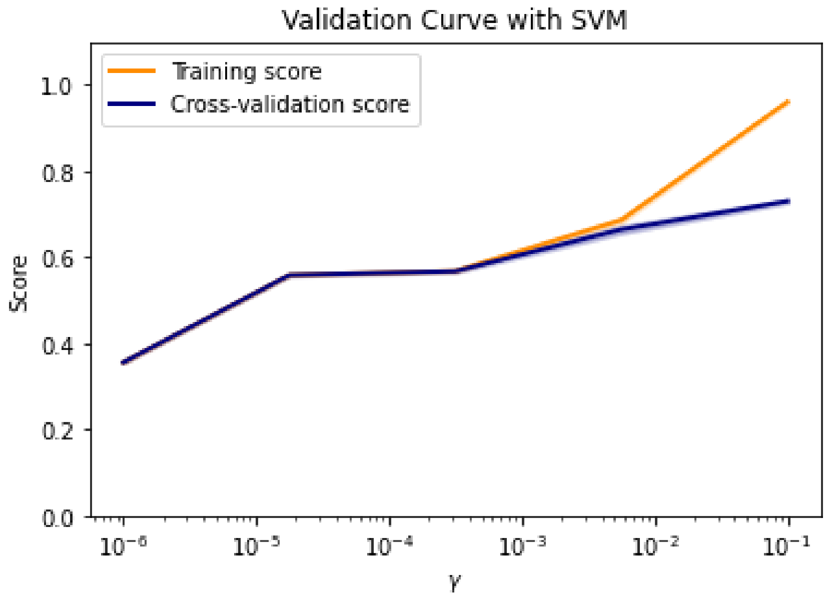
3.SVC 一些調參

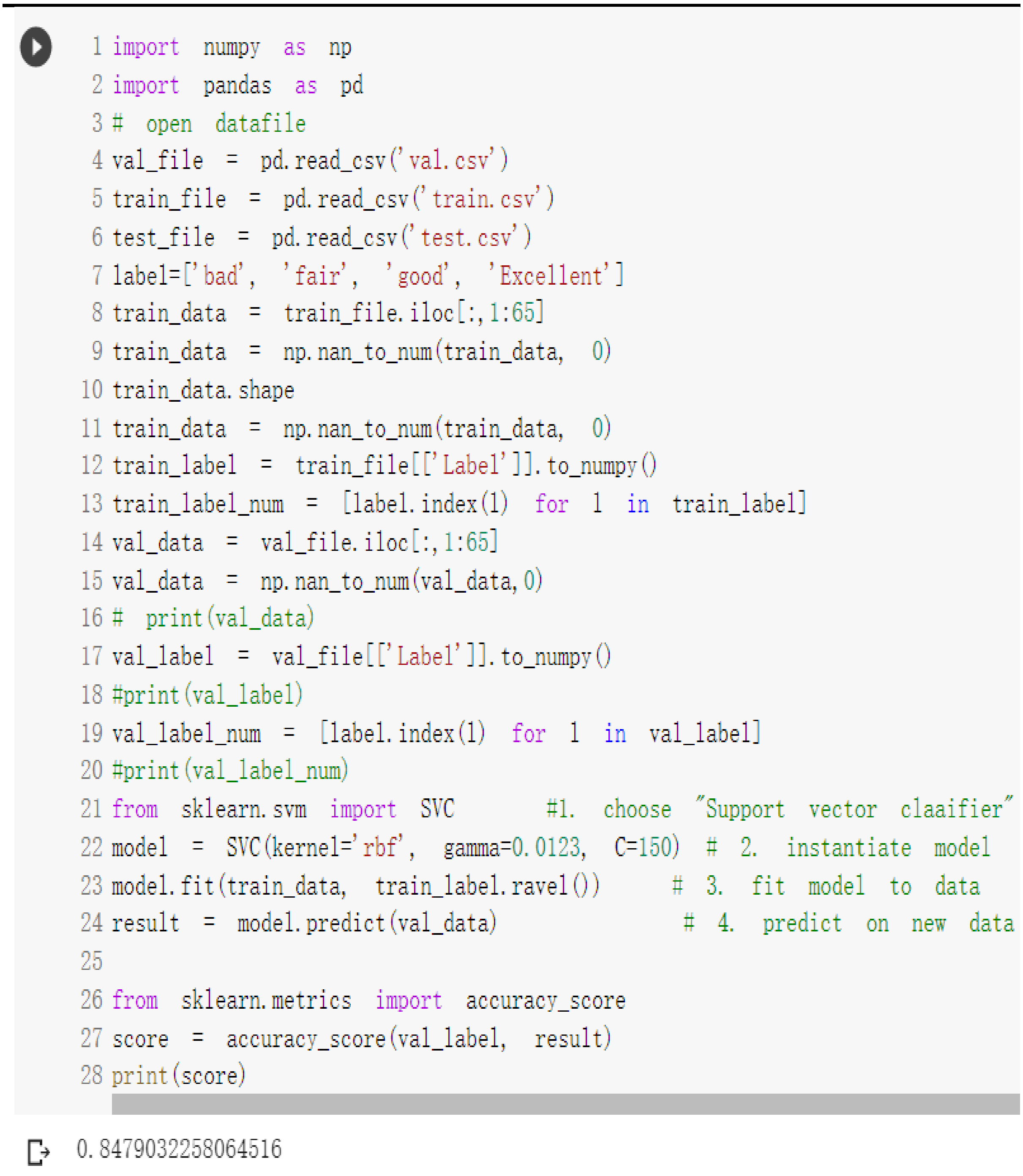
使用 GridSearchCV 和 Validation Curve 找到模型最佳的

gamma 和 C



```
Best parameters set found on development set:

{'C': 150, 'gamma': 0.012, 'kernel': 'rbf'}

Grid scores on development set:

0.484 (+/-0.002) for {'C': 0.01, 'gamma': 0.012, 'kernel': 'rbf'}
0.633 (+/-0.011) for {'C': 1, 'gamma': 0.012, 'kernel': 'rbf'}
0.742 (+/-0.017) for {'C': 150, 'gamma': 0.012, 'kernel': 'rbf'}

Detailed classification report:

The model is trained on the full development set.
The scores are computed on the full evaluation set.

              precision    recall  f1-score   support

   Excellent       1.00      1.00      1.00      1086
         bad       1.00      0.94      0.97       217
        fair       0.97      0.99      0.98       787
        good       0.99      0.99      0.99      1110

    accuracy                           0.99      3200
   macro avg       0.99      0.98      0.98      3200
weighted avg       0.99      0.99      0.99      3200
```





```
import numpy as np
import pandas as pd
# open datafile
val_file = pd.read_csv('val.csv')
train_file = pd.read_csv('train.csv')
test_file = pd.read_csv('test.csv')
label=['bad', 'fair', 'good', 'Excellent']
train_data = train_file.iloc[:,1:65]
train_data = np.nan_to_num(train_data, 0)
train_data.shape
train_data = np.nan_to_num(train_data, 0)
train_label = train_file[['Label']].to_numpy()
train_label_num = [label.index(l) for l in train_label]
val_data = val_file.iloc[:,1:65]
val_data = np.nan_to_num(val_data,0)
# print(val_data)
val_label = val_file[['Label']].to_numpy()
#print(val_label)
val_label_num = [label.index(l) for l in val_label]
#print(val_label_num)
from sklearn.svm import SVC       #1. choose "Support vector claaifier"
model = SVC(kernel='rbf', gamma=0.0123, C=150) # 2. instantiate model
model.fit(train_data, train_label.ravel())     # 3. fit model to data
result = model.predict(val_data)               # 4. predict on new data

from sklearn.metrics import accuracy_score
score = accuracy_score(val_label, result)
print(score)
```

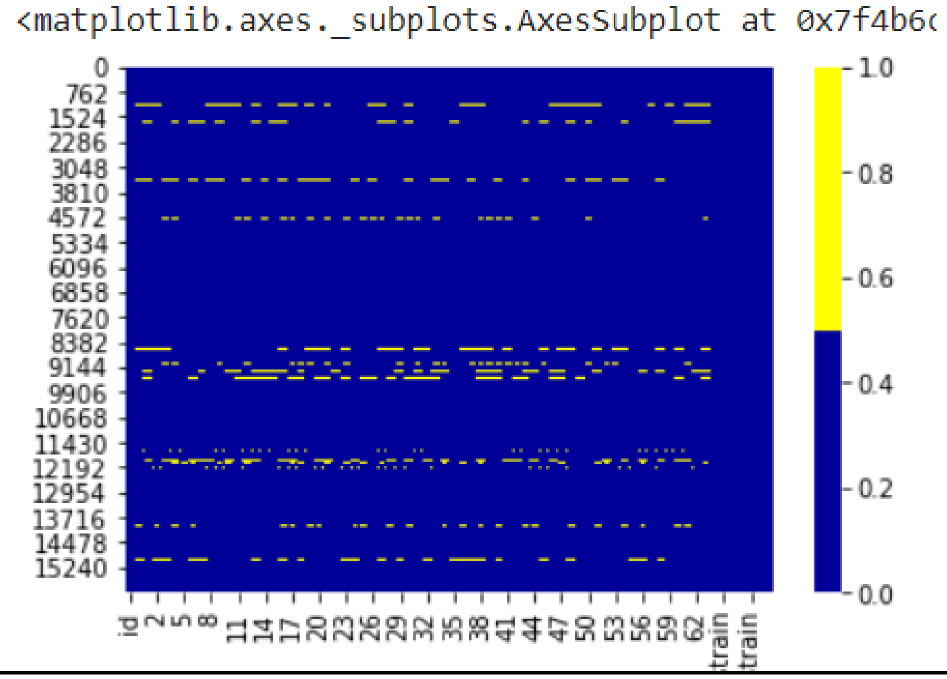
```
0.8479032258064516
```

模型準確度:0.84790

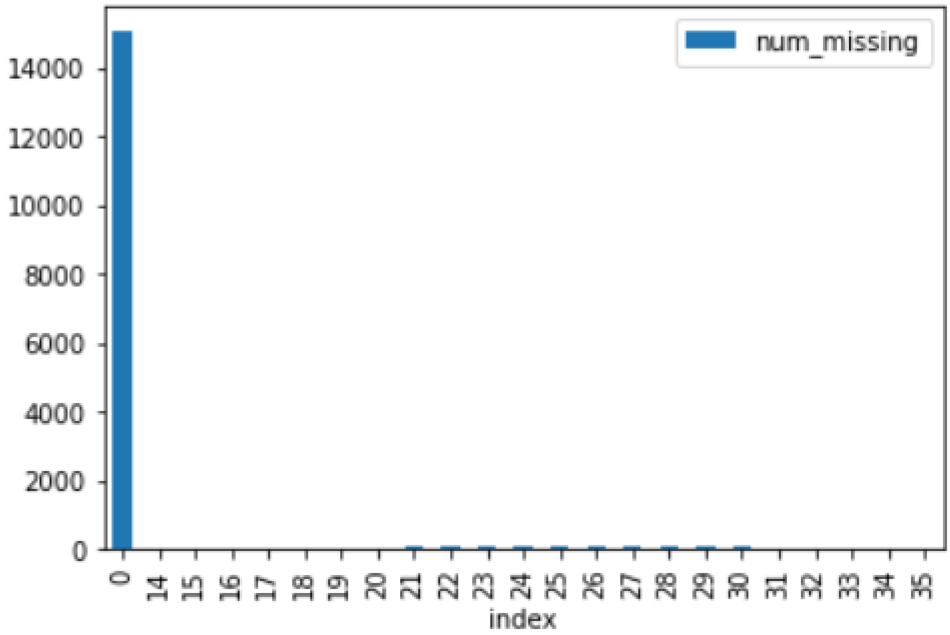
4.Data Clean- Missing data

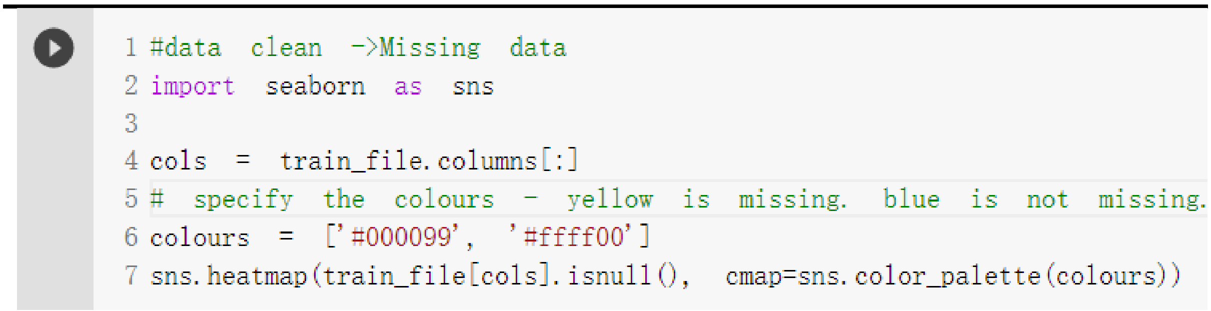
發現訓練集中某些欄位有缺值,因此刪除沒有值的特定項

熱圖,柱狀圖顯示出哪些欄位有缺值



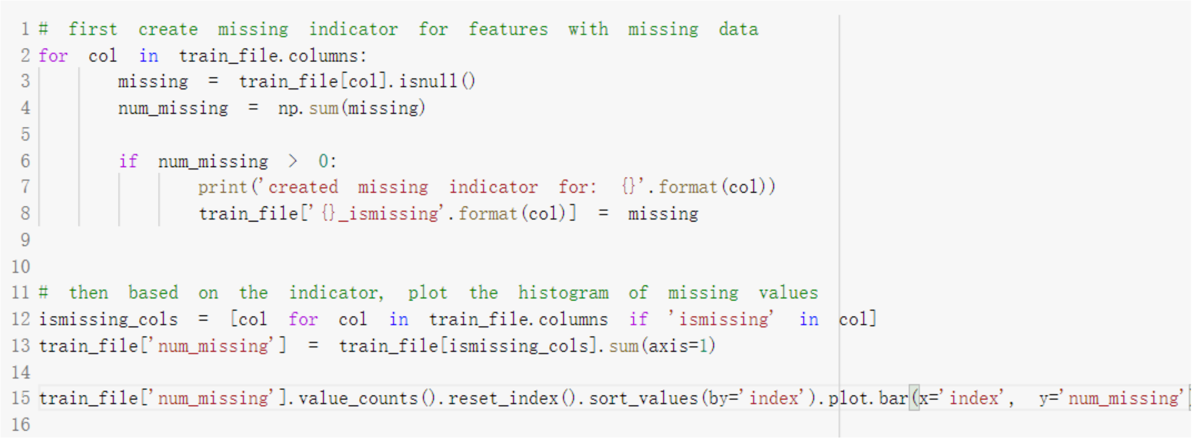
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4b6c
```





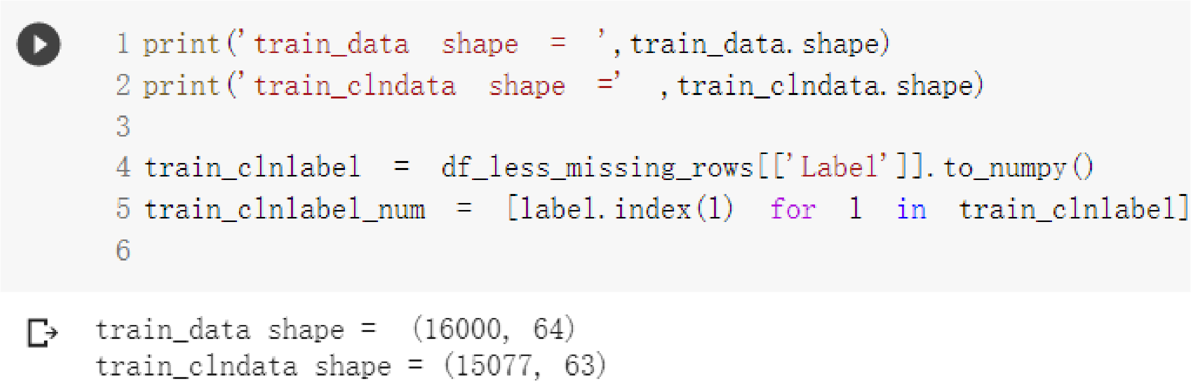
```
#data clean ->Missing data
import seaborn as sns

cols = train_file.columns[:]
# specify the colours - yellow is missing. blue is not missing.
colours = ['#000099', '#ffff00']
sns.heatmap(train_file[cols].isnull(), cmap=sns.color_palette(colours))
```

用熱點圖表示 有缺值的欄位



```
# first create missing indicator for features with missing data
for col in train_file.columns:
    missing = train_file[col].isnull()
    num_missing = np.sum(missing)

    if num_missing > 0:
        print('created missing indicator for: {}'.format(col))
        train_file['{}_ismissing'.format(col)] = missing


# then based on the indicator, plot the histogram of missing values
ismissing_cols = [col for col in train_file.columns if 'ismissing' in col]
train_file['num_missing'] = train_file[ismissing_cols].sum(axis=1)

train_file['num_missing'].value_counts().reset_index().sort_values(by='index').plot.bar(x='index', y='num_missing'

```

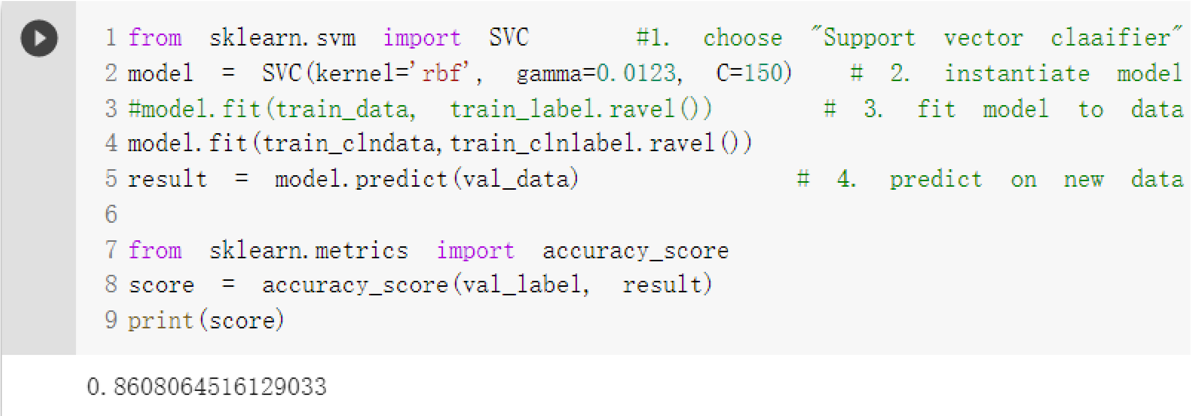
用柱狀圖表示 並且標出有缺值的欄位



```
print('train_data shape = ',train_data.shape)
print('train_clndata shape =' ,train_clndata.shape)

train_clnlabel = df_less_missing_rows[['Label']].to_numpy()
train_clnlabel_num = [label.index(1) for l in train_clnlabel]

```

```
train_data shape =  (16000, 64)
train_clndata shape = (15077, 63)
```

移除有缺值得欄位,對比原先訓練集 ,新的訓練及欄位減少一些



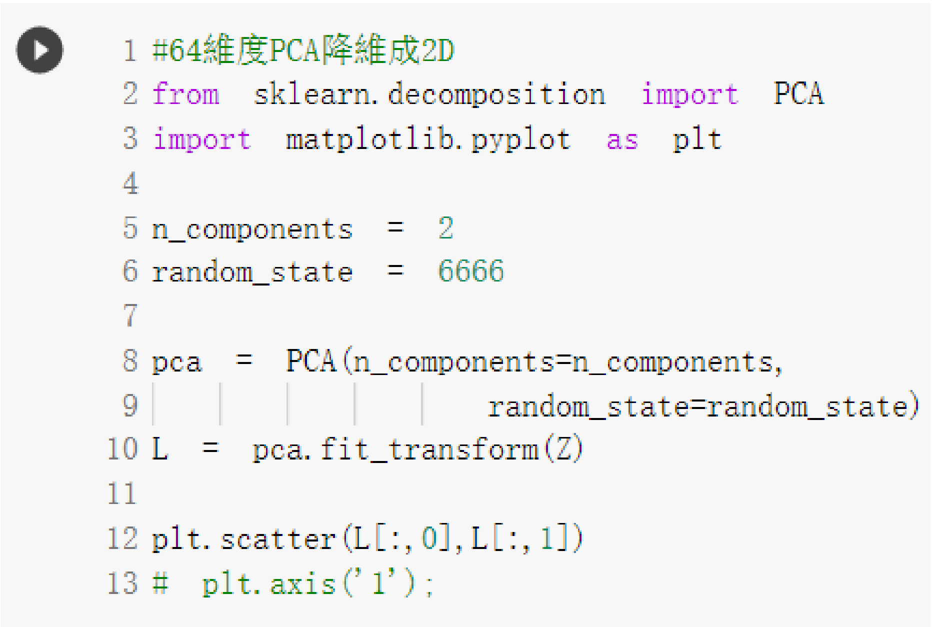
```
from sklearn.svm import SVC         #1. choose "Support vector claaifier"
model = SVC(kernel='rbf', gamma=0.0123, C=150)    # 2. instantiate model
#model.fit(train_data, train_label.ravel())      # 3. fit model to data
model.fit(train_clndata,train_clnlabel.ravel())
result = model.predict(val_data)              # 4. predict on new data

from sklearn.metrics import accuracy_score
score = accuracy_score(val_label, result)
print(score)
```

```
0.8608064516129033
```

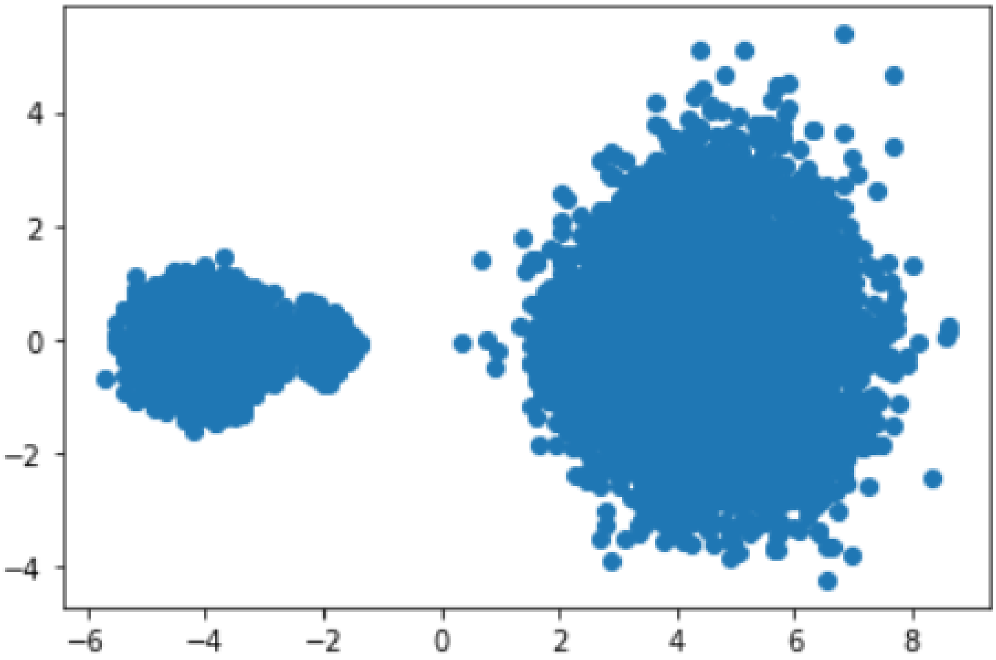
模型準確度:0.86080

其他嘗試

使用 PCA 把 8*8 的圖形降低維度,在 2D 空間中表現



```
#64維度PCA降維成2D
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

n_components = 2
random_state = 6666

pca = PCA(n_components=n_components,
          random_state=random_state)
L = pca.fit_transform(Z)

plt.scatter(L[:,0],L[:,1])
# plt.axis('1');
```



過濾掉一些離群值,或許預測準度會上升