

銘傳大學

資訊傳播工程學系

多媒體通訊期末專題報告

題目

班級：資傳二甲

組員：

曾宏鈞

彭俐嘉

黃旭雲

中華民國一〇八年五月十五日

1. 研究方法

1.1 研究資料

問題簡述(UCI 資料集). 資料是分類?迴歸?目的為何?

資料: Flags Data Set

目標: 分類

由各個國家及國旗數據中, 預測 國家的宗教及信仰

資料屬性為何(有哪些屬性,代表甚麼意思)

屬性

1.~~名稱: 有關國家的名稱~~

2.大陸: 1 = N.America, 2 = S.America, 3 =歐洲, 4 =非洲, 5 =亞洲, 6 =大洋洲

3.區域: 地理象限, 基於格林威治和赤道; 1 = NE, 2 = SE, 3 = SW, 4 = NW

4.面積: 數千平方公里

5.人口: 數百萬

6.語言: 1 =英語, 2 =西班牙語, 3 =法語, 4 =德語, 5 =斯拉夫語, 6 =其他印歐語, 7 =中文, 8 =阿

拉伯語，9 = 日語/土耳其語/芬蘭語/馬扎爾語，10 = 其他

~~7.宗教：0 = 天主教徒，1 = 其他基督徒，2 = 穆斯林，3 = 佛教徒，4 = 印度教徒，5 = 民族，6 = 馬克思主義者，7 = 其他~~

8.豎條紋：旗幟中的豎條數量

9.橫條紋：旗幟中的橫條紋

10.顏色：國旗的顏色數量

11.紅：0，如果不存在，1，如果存在

12.綠色：0，如果不存在，1，如果存在

13.藍色：0，如果不存在，1，如果存在

14.金：同為金（也黃色）

15.白：0，如果不存在，1，如果存在

16.黑色：0，如果不存在，1，如果存在

17.橙色：橙色相同（也是棕色）

18.主色調：旗幟中的主色調（通過採用最頂部的色調決定打破色調，如果失敗則取決於最中心的色調，如果失敗則最左邊的色調）。

19.圓圈：標誌中的圓圈數量

- 20.十字架:(直立) 十字架的 數量
- 21.撒鹽人數: 對角線十字架的 數量
- 22.四分之一: 四分區數量
- 23.太陽星數: 太陽或星形符號的數量
- 24.新月: 如果有月牙符號, 則為 1, 否則為 0
- 25.三角形: 1 如果存在任何三角形, 則為 0 否則為 0
- 26.圖標: 1 如果存在無生命圖像 (例如, 船), 否則為 0
- 27.動畫: 1 如果一個有生命的圖像 (例如, 一隻老鷹, 一棵樹, 一隻人的手) 出現, 0 否則
- 28.文字: 1 如果有任何字母或書寫在旗幟上 (例如, 座右銘或口號), 0 否則
- 29. **opleft**: color 在左上角 (向右移動以決定打破平局)
- 30. **botright**: 左下角的顏色 (向左移動以決定打破中斷)

資料標籤為何(你的標籤是什麼,代表甚麼意思)

1.2 使用的方法

1. 期末報告題目及使用的演算法

目標：

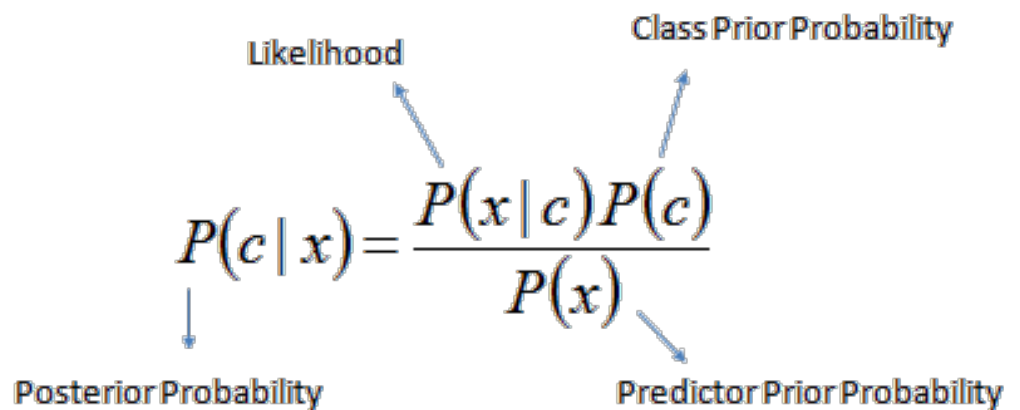
使用旗子的數據，來分類其國家可能的宗教。

演算法：高斯貝氏分類、SVC

2. 使用哪些演算法？可否簡介演算法的特性！分

類？分群？迴歸？預測？決策？

貝氏分類式子



The diagram shows the formula for Bayes' Theorem: $P(c | x) = \frac{P(x | c)P(c)}{P(x)}$. Arrows point from the terms to their labels: $P(c | x)$ points to 'Posterior Probability', $P(x | c)$ points to 'Likelihood', $P(c)$ points to 'Class Prior Probability', and $P(x)$ points to 'Predictor Prior Probability'.

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \cdots \times P(x_n | c) \times P(c)$$

注意事項:

- 訓練資料要有完整的類別，不然會有零概率問題
- 每個特徵彼此獨立(天真假設)

算法:

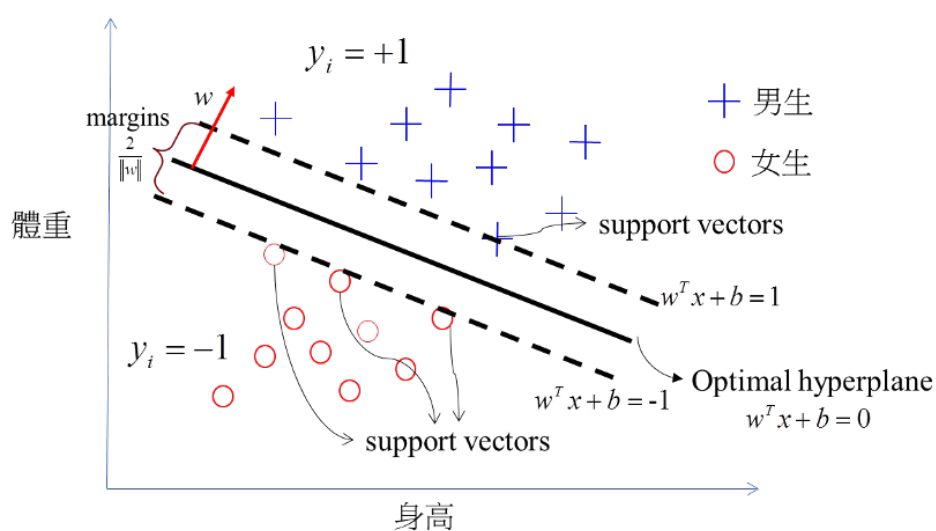
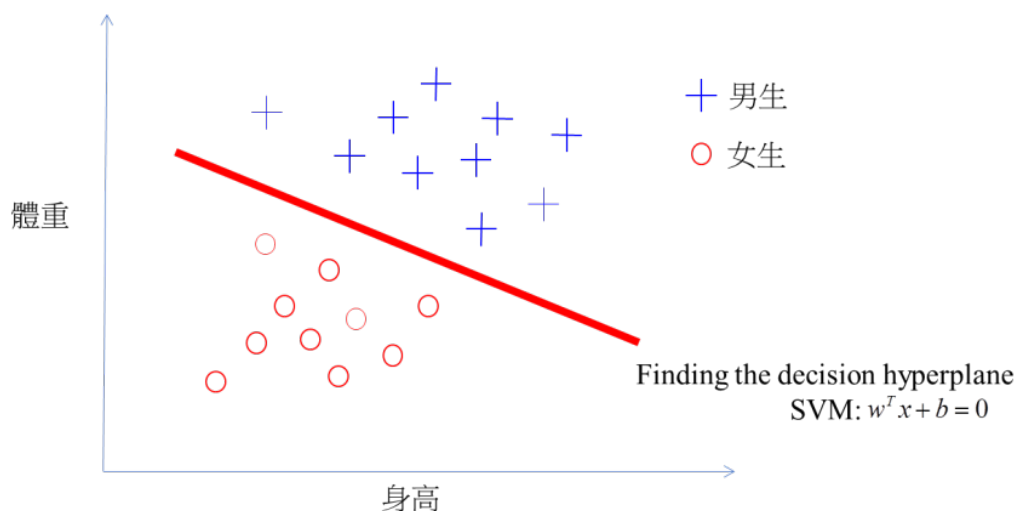
1. 先算 類別 的先驗機率
2. 找出每個 特徵的 似然機率
3. 在貝氏公式計算後驗
4. 找哪個 類別 機率最大

然而，實際運算的時候是取 \log 來避免計算耗時與精確度不足的問題。

其中有許多的變形，我們使用的是高斯貝氏，其假設每個特徵是常態分佈的。

SVC

找到一個決策邊界



上圖是從參考資料中解釋的圖，透過類別之間的支持向量，找到決策邊界，使兩類間的邊界大到最大化。

2. 實驗

2.1 程式碼講解(包含清晰的程式碼貼圖與說明)

#讀資料

```
rawdataAll = pd.read_csv('data/flag.csv')  
rawdataAll.head()
```

		1	2	3	4	5	6	7	8	9	10	...	21	22	23	24	25	26	27	28	29	30
0	Afghanistan	5	1	648	16	10	2	0	3	5	...	0	0	1	0	0	1	0	0	black	green	
1	Albania	3	1	29	3	6	6	0	0	3	...	0	0	1	0	0	0	1	0	red	red	
2	Algeria	4	1	2388	20	8	2	2	0	3	...	0	0	1	1	0	0	0	0	green	white	
3	American-Samoa	6	3	0	0	1	1	0	0	5	...	0	0	0	0	1	1	1	0	blue	red	
4	Andorra	3	1	0	0	6	0	3	0	3	...	0	0	0	0	0	0	0	0	blue	red	

#string to float

```
df['29'] = pd.factorize(df['29'])[0] + 1  
df['30'] = pd.factorize(df['30'])[0] + 1  
df['18'] = pd.factorize(df['18'])[0] + 1  
df['1'] = pd.factorize(df['1'])[0] + 1
```

	1	2	3	4	5	6	7	8	9	10	...	21	22	23	24	25	26	27	28	29	30
0	1	5	1	648	16	10	2	0	3	5	...	0	0	1	0	0	1	0	0	1	1
1	2	3	1	29	3	6	6	0	0	3	...	0	0	1	0	0	0	1	0	2	2
2	3	4	1	2388	20	8	2	2	0	3	...	0	0	1	1	0	0	0	0	3	3
3	4	6	3	0	0	1	1	0	0	5	...	0	0	0	0	1	1	1	0	4	2
4	5	3	1	0	0	6	0	3	0	3	...	0	0	0	0	0	0	0	0	4	2

5 rows × 30 columns


```
#統計每個類別數量(全部資料)
```

```
allTarget = target.value_counts()
```

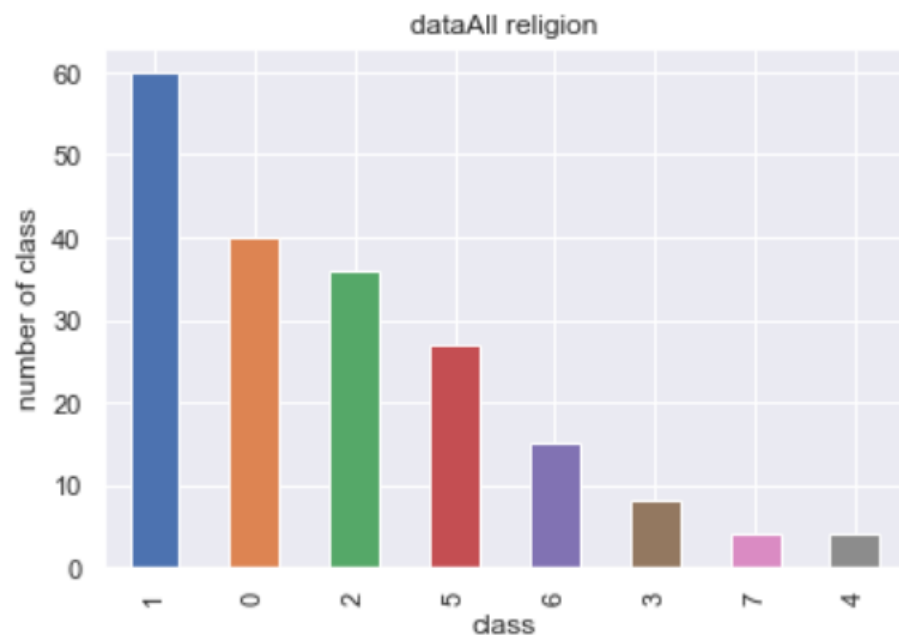
```
print(allTarget)
```

```
ax = allTarget.plot(kind='bar',title='dataAll religion')
```

```
ax.set_ylabel('number of class')
```

```
ax.set_xlabel('class')
```

```
Out[411]: Text(0.5, 0, 'class')
```



```
#建立貝氏分類器
```

```
nbm = GaussianNB()
```

```
nbm.fit(X_train,Y_train)
```

準確率

```
from sklearn.metrics import accuracy_score
```

```
accuracy_score(Y_test, Y_predict)
```

```
Out[442]: 0.40816326530612246
```

```

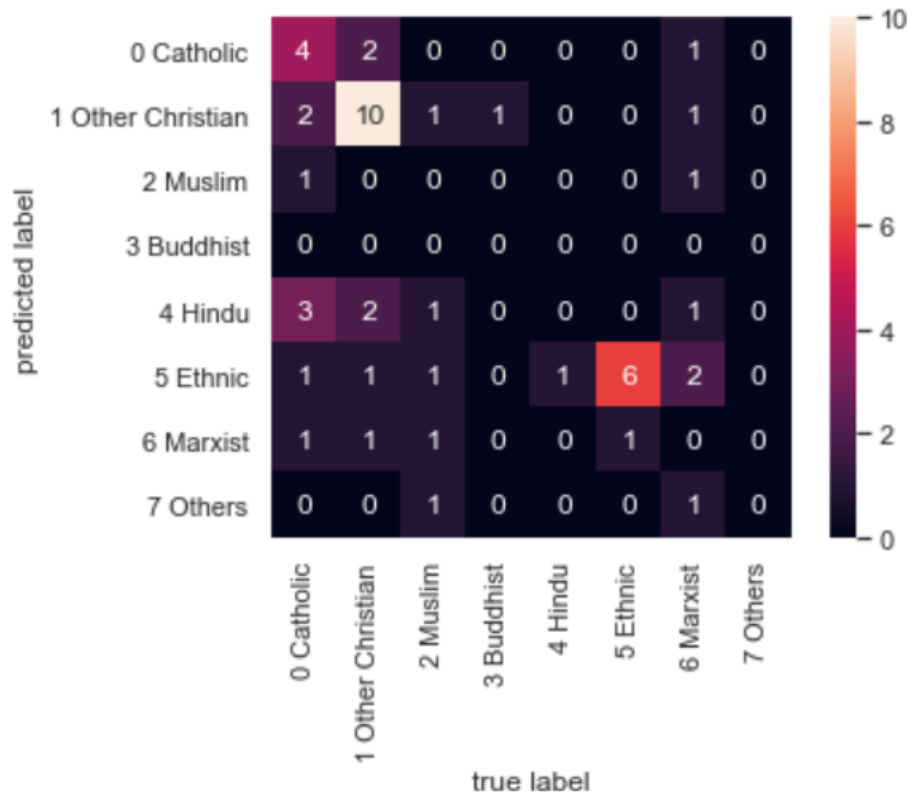
from sklearn.metrics import confusion_matrix
#混淆矩陣
mat = confusion_matrix(Y_test, Y_predict)
print(mat)
#畫圖
sns.heatmap(mat.T, square=True, annot=True, cbar=True,
             xticklabels=target_names, yticklabels=target_names
            )
plt.xlabel('true label')
plt.ylabel('predicted label');

```

```

[[ 4  2  1  0  3  1  1  0]
 [ 2 10  0  0  2  1  1  0]
 [ 0  1  0  0  1  1  1  1]
 [ 0  1  0  0  0  0  0  0]
 [ 0  0  0  0  0  1  0  0]
 [ 0  0  0  0  0  6  1  0]
 [ 1  1  1  0  1  2  0  1]
 [ 0  0  0  0  0  0  0  0]]

```



Precision 該項分對的 / 所有

Recall 該項分對的 / 所有分對的

F1-score

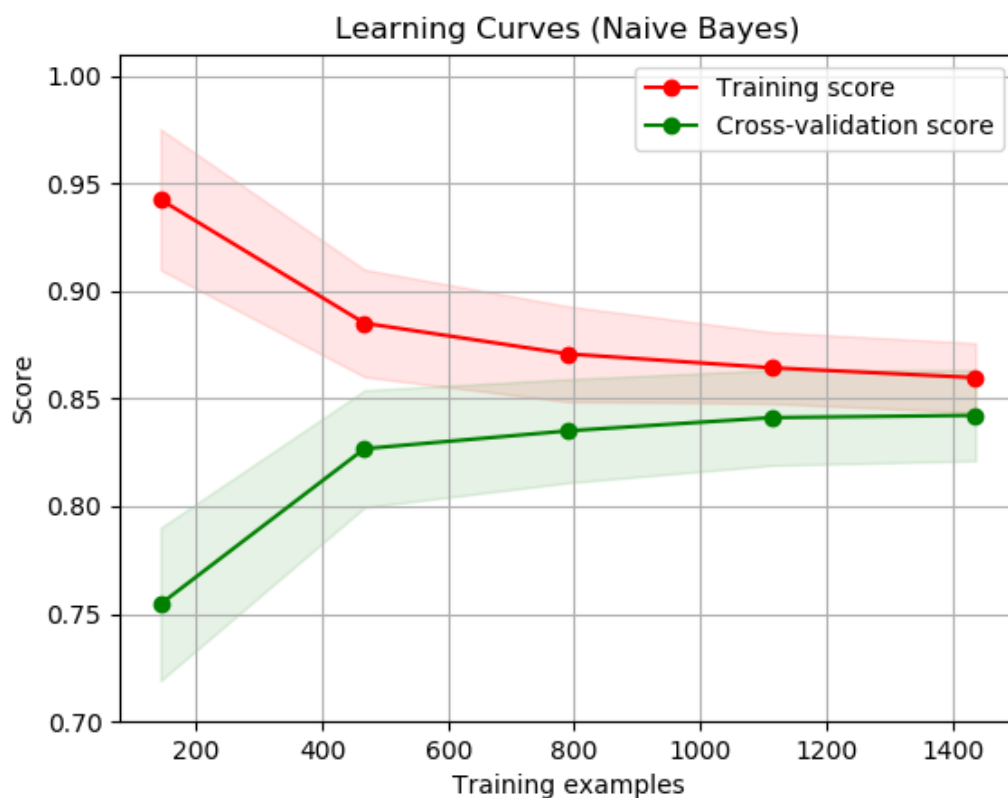
$$2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Support 有幾個

學習曲線程式

在機器學習評估裡，還有個評估方法是學習曲線，通過此曲線，我們更可以知道模型的訓練過程，來觀察是否有過度擬和的情形。

在 `sk-learn` 中，`Learning Curve` 是先將原始數據拆分成不同大小的子集合，每一個子集合再去做 `cross validation` 使得分較公平。



```

#定義畫學習曲線的函式
def plot_learning_curve(estimator, title, X, y, ylim=None, cv=None,
                        n_jobs=None, train_sizes=np.linspace(.1, 1.0, 5)):
    """
    ylim y 的限制

    """
    #畫圖
    plt.figure()
    plt.title(title)
    if ylim is not None:
        plt.ylim(*ylim)
    plt.xlabel("Training examples")
    plt.ylabel("Score")
    train_sizes, train_scores, test_scores = learning_curve(
        estimator, X, y, cv=cv, n_jobs=n_jobs, train_sizes=train_sizes)
    train_scores_mean = np.mean(train_scores, axis=1)
    train_scores_std = np.std(train_scores, axis=1)
    test_scores_mean = np.mean(test_scores, axis=1)
    test_scores_std = np.std(test_scores, axis=1)

    plt.grid()

    plt.fill_between(train_sizes, train_scores_mean - train_scores_std,
                     train_scores_mean + train_scores_std, alpha=0.1,
                     color="r")
    plt.fill_between(train_sizes, test_scores_mean - test_scores_std,
                     test_scores_mean + test_scores_std, alpha=0.1,
color="g")
    plt.plot(train_sizes, train_scores_mean, 'o-', color="r",
             label="Training score")
    plt.plot(train_sizes, test_scores_mean, 'o-', color="g",
             label="Cross-validation score")

    plt.legend(loc="best")
    return plt

```

特徵及答案

```
X,y =features, target
```

切分 cross validation

```
#學習曲線
title = "Learning Curves (Naive Bayes)"
# Cross validation with 100 iterations to get smoother mean test and train
#隨機抽樣切分 0.2 作為 cross validation
cv = ShuffleSplit(n_splits=10, test_size=0.2, random_state=0)
```

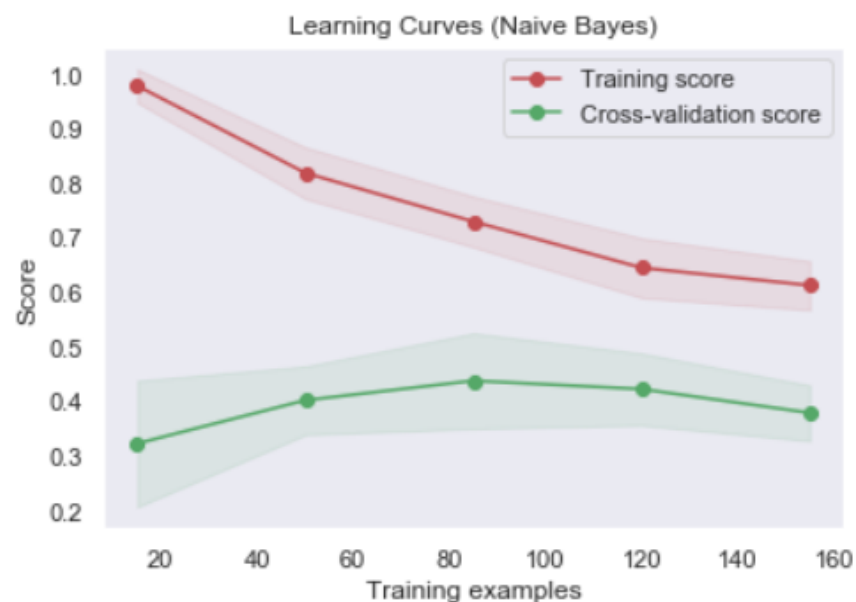
建立估計模型

```
#建立估計模型
estimator = GaussianNB()
```

呼叫函式畫圖

```
#畫圖
plot_learning_curve(estimator, title, X, y, cv=cv, n_jobs=4)
```

```
Out[1]: cut selected cells e 'matplotlib.pyplot' from 'C:\\Users\\alant\\Ana
ckages\\matplotlib\\pyplot.py'>
```



2.2 課堂報告時老師給了甚麼意見,改進了哪些部分?

可否多做幾個演算法比較?

➤ 已更新(SVC)

2.3 實驗資料筆數, 訓練資料 vs. 測試資料數量, 實驗

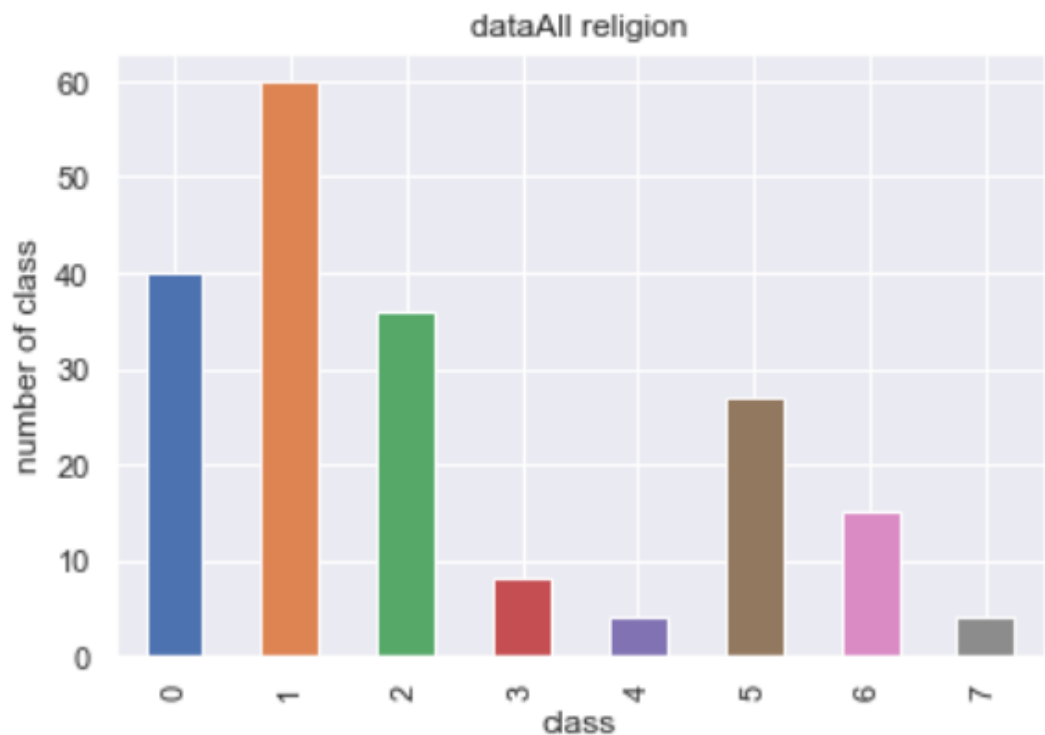
結果分析(針對自己輸出的結果進行分析, 例如

recall, precision, F1, confusion matrix, accuracy,

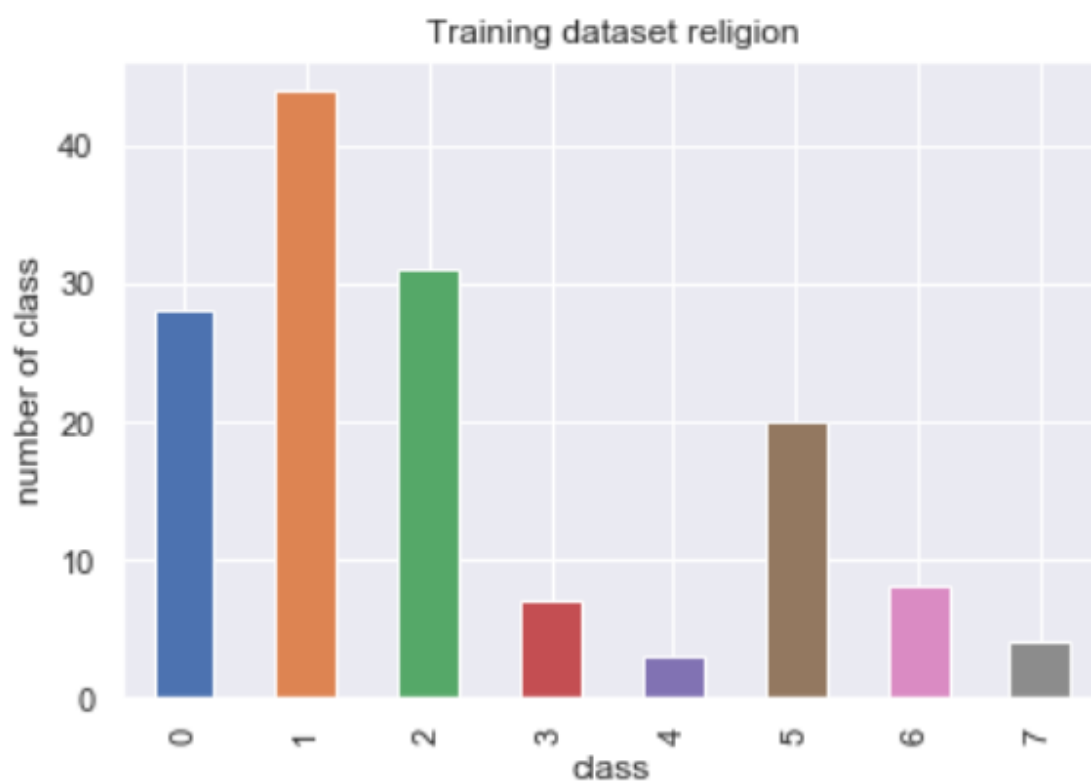
sensitivity, specificity, TP, TN, FP, FN, MSE...).

若有測試多種演算法,也請比較數據.

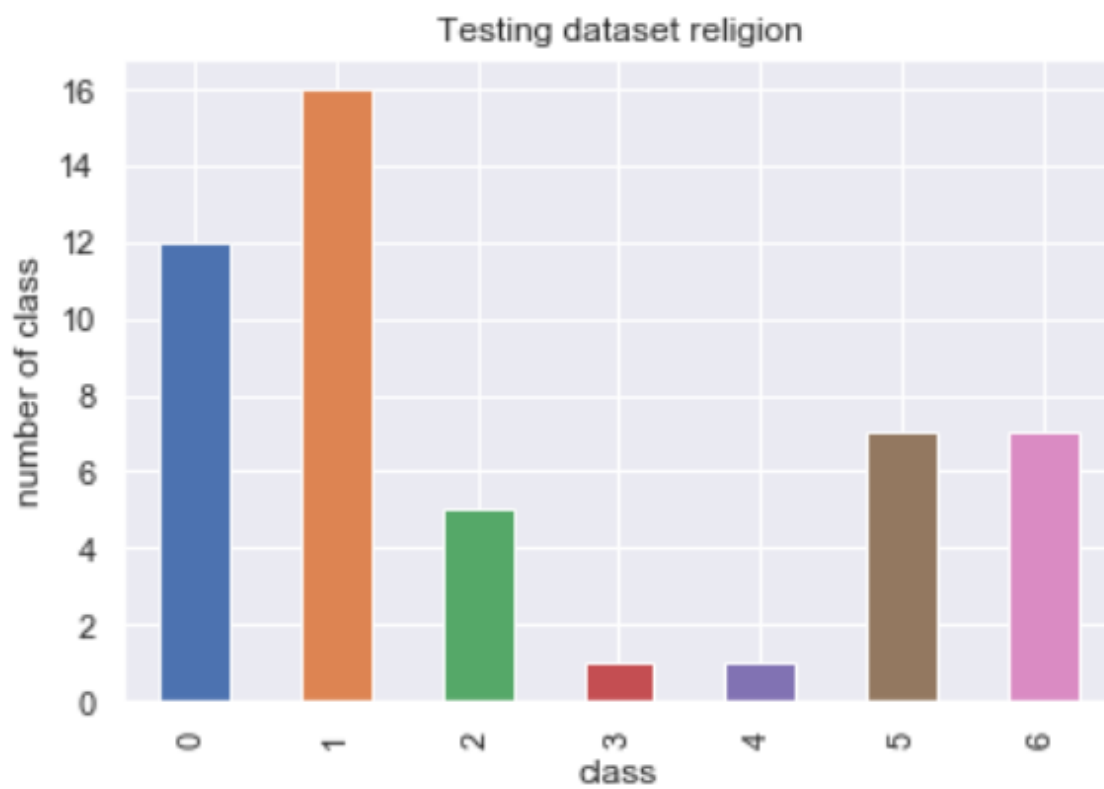
資料分布



原始數據



訓練資料



測試資料

all		train		test	
1	60	1	44	1	16
0	40	2	31	0	12
2	36	0	28	6	7
5	27	5	20	5	7
6	15	6	8	2	5
3	8	3	7	4	1
7	4	7	4	3	1
4	4	4	3		

我們建立了兩個模型：Model1、Model2，Model1 是所有特徵，Model2 是經過特徵選擇去做訓練的。

Model2 是根據各個宗教的參考資料，發現宗教與分布的地區、語言相關，因此我們選擇 2、6、24、29 來作為我們的特徵訓練來提高準確度。

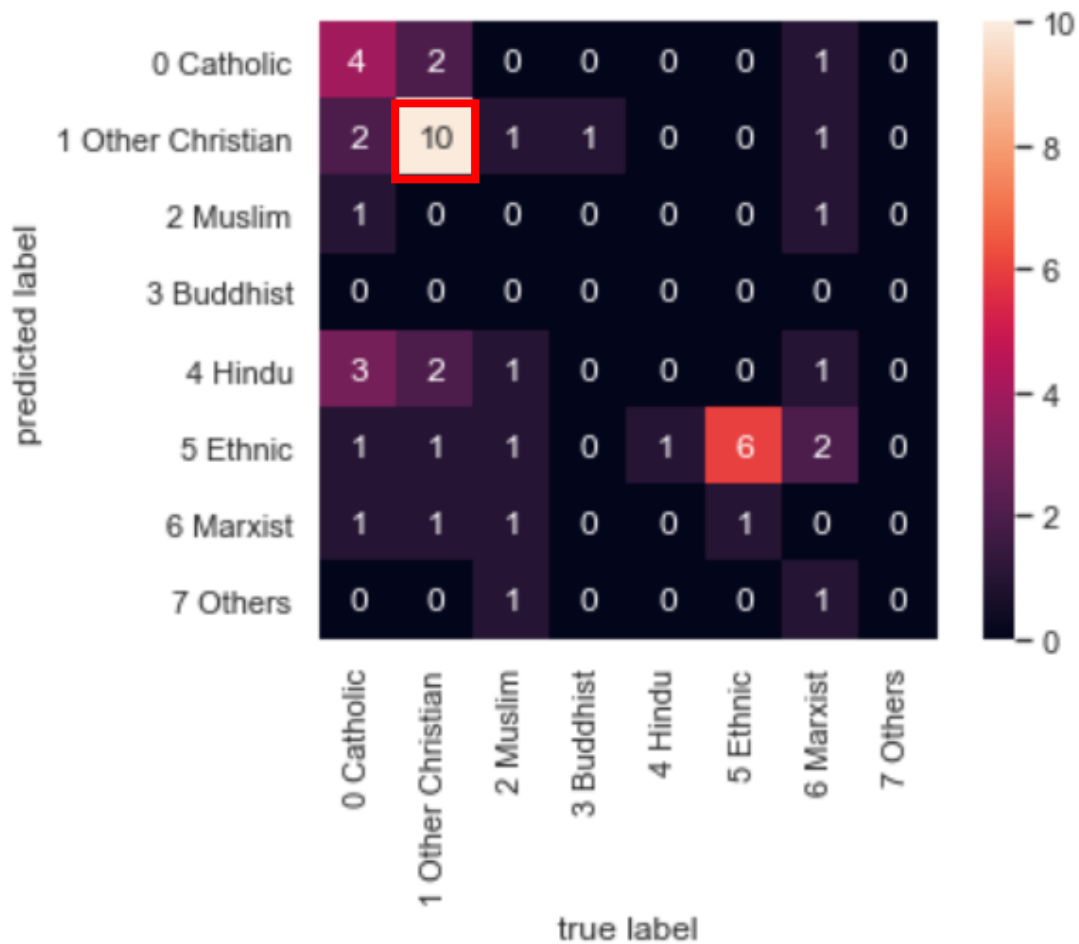
Model1(全部)

準確率: 0.408

Report:

	precision	recall	f1-score	support
0	0.57	0.33	0.42	12
1	0.67	0.62	0.65	16
2	0.00	0.00	0.00	5
3	0.00	0.00	0.00	1
4	0.00	0.00	0.00	1
5	0.50	0.86	0.63	7
6	0.00	0.00	0.00	7
7	0.00	0.00	0.00	0
micro avg	0.41	0.41	0.41	49
macro avg	0.22	0.23	0.21	49
weighted avg	0.43	0.41	0.40	49

混淆矩陣:



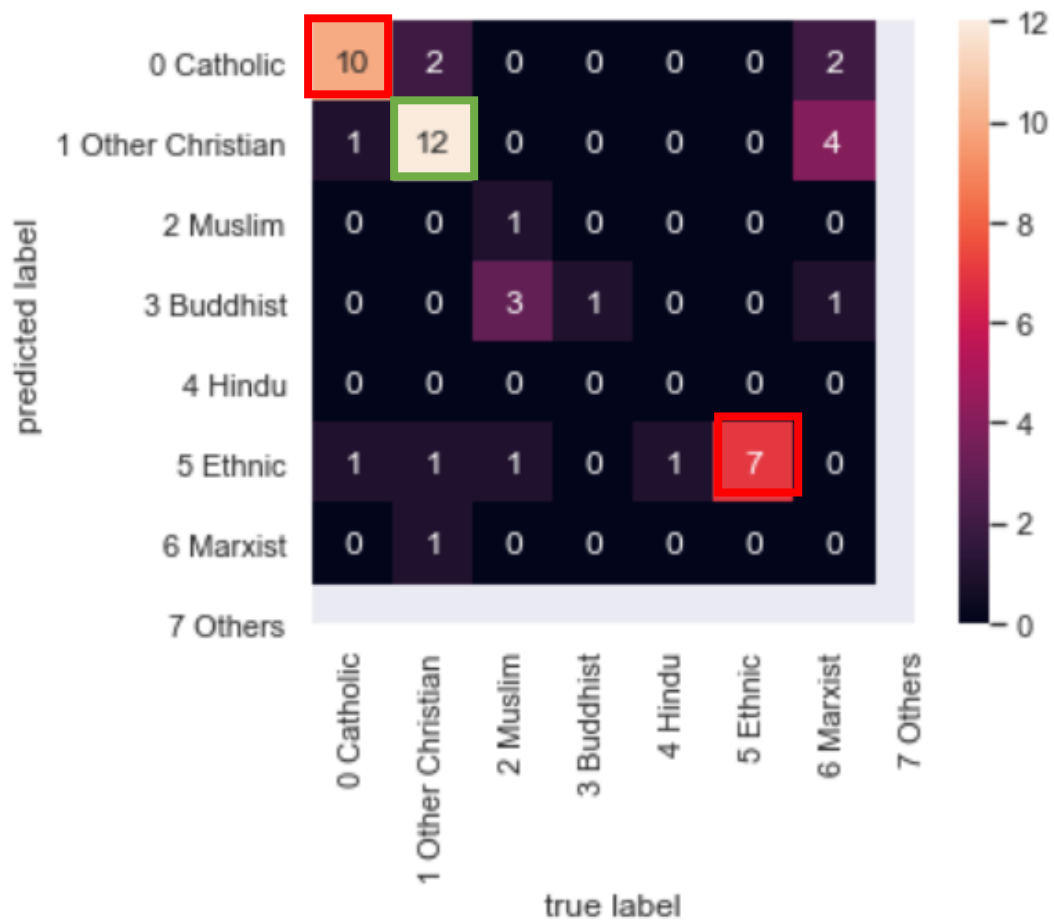
Model2(特徵選擇 2、6、24、29)

準確率: 0.632

Report:

	precision	recall	f1-score	support
0	0.71	0.83	0.77	12
1	0.71	0.75	0.73	16
2	1.00	0.20	0.33	5
3	0.20	1.00	0.33	1
4	0.00	0.00	0.00	1
5	0.64	1.00	0.78	7
6	0.00	0.00	0.00	7
micro avg	0.63	0.63	0.63	49
macro avg	0.47	0.54	0.42	49
weighted avg	0.60	0.63	0.58	49

混淆矩陣:



3. 結論與未來工作

	precision	recall	f1-score	support
0	0.57	0.33	0.42	12
1	0.67	0.62	0.65	16
2	0.00	0.00	0.00	5
3	0.00	0.00	0.00	1
4	0.00	0.00	0.00	1
5	0.50	0.86	0.63	7
6	0.00	0.00	0.00	7
7	0.00	0.00	0.00	0
micro avg	0.41	0.41	0.41	49
macro avg	0.22	0.23	0.21	49
weighted avg	0.43	0.41	0.40	49

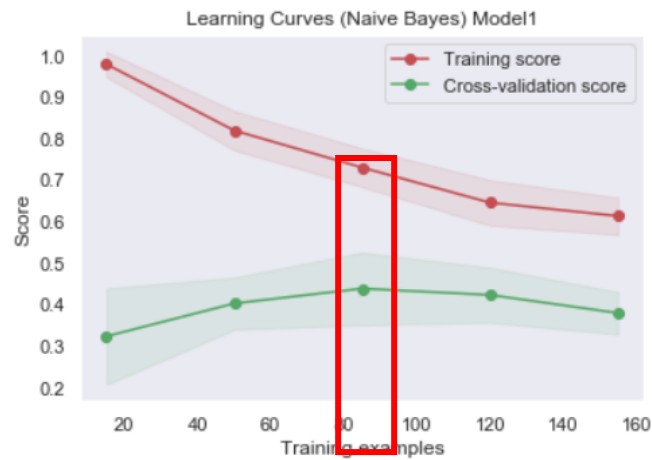
Model1(ALL)

	precision	recall	f1-score	support
0	0.71	0.83	0.77	12
1	0.71	0.75	0.73	16
2	1.00	0.20	0.33	5
3	0.20	1.00	0.33	1
4	0.00	0.00	0.00	1
5	0.64	1.00	0.78	7
6	0.00	0.00	0.00	7
micro avg	0.63	0.63	0.63	49
macro avg	0.47	0.54	0.42	49
weighted avg	0.60	0.63	0.58	49

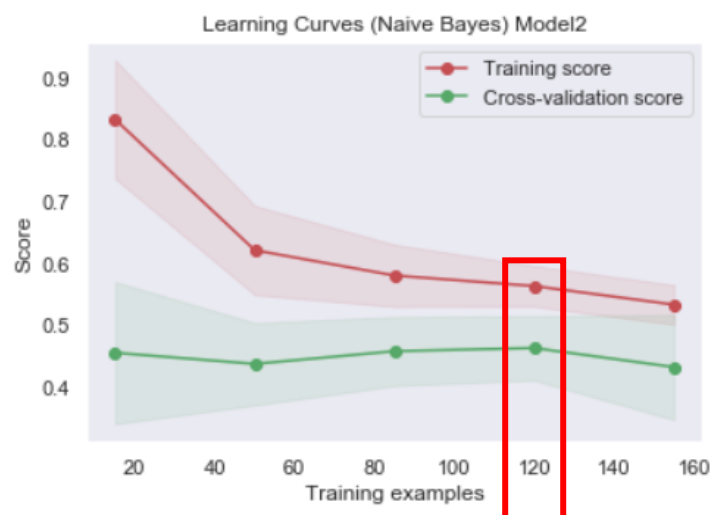
Model2(feature select)

觀察 model1、model2 我們可以發現，經過特徵選擇後，0~5 類的 f1-score 明顯的上升，因此在使用傳統機器學習演算法時，根據常識或者專家給定的知識來做特徵選擇及篩選是非常重要的。

學習曲線



Model1

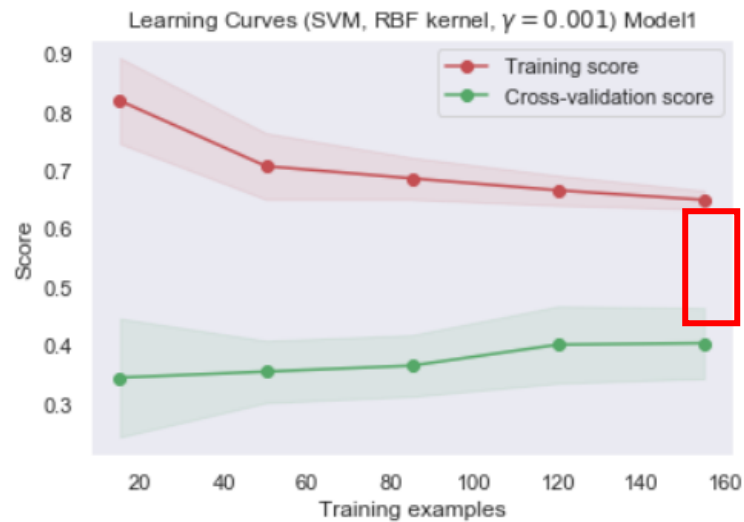


Model2

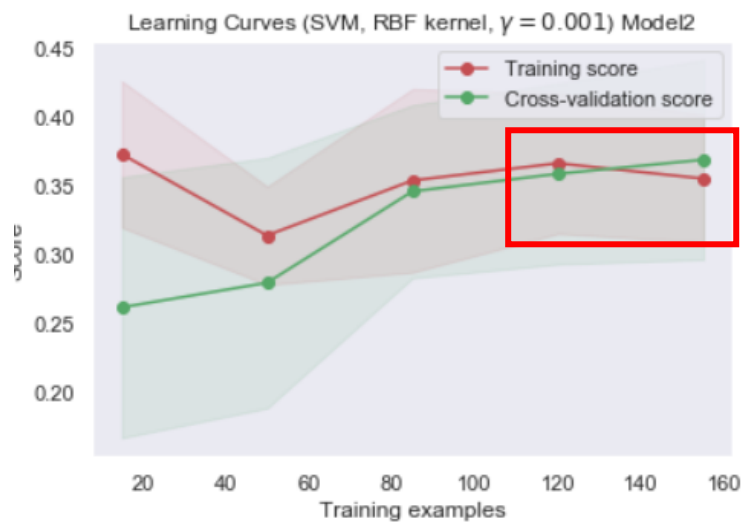
Model1 在子集數目 90 時，開始發生 overfitting 的現象，Cross validation 分數不斷下降，使其整體訓練分數不斷降低。發生 high-variance。

Model2 在子集數目 120 時，開始發生 overfitting，準確率不斷下降，

綜合上述，使用 GaussianNB 的 Model1 與 Model2 皆發生 High-variance 的現象，增加訓練資料可以使其訓練分數提高。



Model1



Model2

Model1 在子集數目 120 時，cross-validation 的分數趨近平緩，而整體分數不在上升，有間距，整體分數較低。發生 high-variance 的現象，也是增加更多訓練資料可以改善。

Model2 在子集數目 130 時，cross-validation 與 Training-score 開始發生交叉，整體的分數較低，發生 high-bias 的問題，若增加訓練資料無法使其分數提高，因此應該再進行 特徵選擇 使其準確率上升。

做此研究有何心得，有什麼意見想反映給老師，未來工作？

這次的作業讓我們瞭解收集資料、整理資料、參考資料的重要性，理論與實務的重要性，勇於發問和求證不懈的毅力。

老師認真負責對待學生的態度，使我們對這次的作業加倍加倍加倍用心，或許我們仍有許多的不足，但我們仍會秉持初心，認真對待每一次的作業，努力解決每一個遇到的困難。

附錄

程式: <https://github.com/alanhc/MMclass>

參考文獻

<https://mropengate.blogspot.com/2015/06/ai-ch14-3-naive-bayes-classifier.html>

https://scikit-learn.org/stable/modules/naive_bayes.html

https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes

<https://archive.ics.uci.edu/ml/datasets/Flags>

<https://zh.wikipedia.org/wiki/%E5%AE%97%E6%95%99>

<https://medium.com/@chih.sheng.huang821/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E6%94%AF%E6%92%90%E5%90%91%E9%87%8F%E6%A9%9F-support-vector-machine-svm-%E8%A9%B3%E7%B4%B0%E6%8E%A8%E5%B0%8E-c320098a3d2e>

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.learning_curve.html