資料探勘　第六組書面報告

組長：B103040047 周安

組員：B103040013 陳佳琪 B103040044 林廷宇 B103040045 楊貽婷 B103040046 余承恩

1. 摘要

利用資料分類演算法，預測病人的各項身體數據與糖尿病的關聯性，透過train data來建立預測模型，再把測資丟入進行預測，並利用準確率跟召回率來檢驗計算結果。

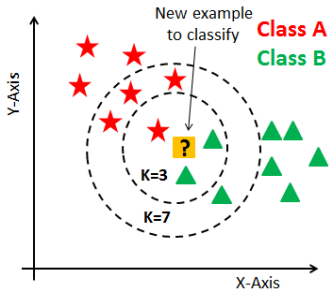
2. 前言簡介

我們先使用KNN演算法代入不同的k值進行預測，判斷k值的underfitting和overfitting的情況；其次我們使用scikit-learn套件來實作Random Forest和高斯貝氏定理兩方法來測試不同演算法下的結果。

3. 相關研究

3.1 KNN

KNN(K Nearest Neighbor)，利用train data建立預測模型，再輸入test data得到預測結果。藉由找放入的測資距離最近的k個鄰居，再根據鄰居的分類多數結果去預測這個資料的分類結果。



以上圖為例，k=3時，此測資被分類為Class B，k=7時，此測資被分類為Class A。

3.1.1 標準化

在資料集中每個資料的數值大小都不同，於是將資料標準化，以避免因為資料數值大小的不同，而導致預測結果有偏差。

標準化的方法：

Min-Max to [0,1]

按照資料的最大值與最小值按照比例縮放，並落在[0,1]之間

公式：$v' = \frac{v-min}{max-min}$

計算距離的方法：

歐幾里得距離

公式：$dis(x,y) = \sqrt{\sum_{i=1}^{n}(Xi - Yi)^2}$

3.2 Random Forest

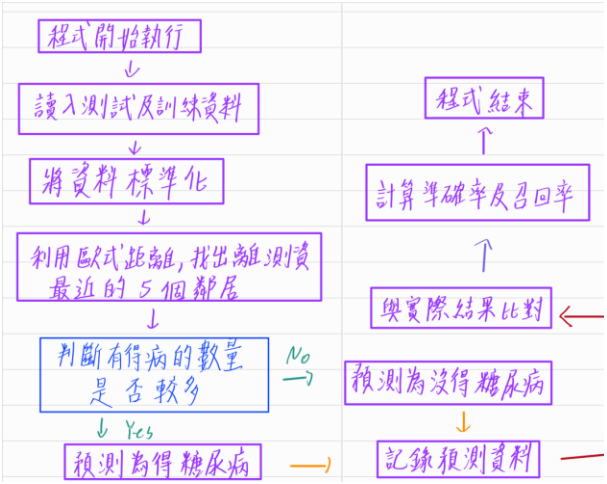
基本原理為決策樹，但比較不同的是Random Forest有多棵決策樹，並從這些樹投票出一個結果，此結果為較優的預測結果。

3.3 高斯貝氏分類器

用於連續型變數(像是年齡以及血糖濃度等.....)，透過計算每個變數的平均數及標準差後做計算，最後將yes機率與no的機率相乘透過大小值比對得出預測結果，基本原理是由機率學之貝氏定理延伸出來。

4. 程式設計方式

4.1 程式流程圖(KNN)



4.2 KNN

第一步將train_data.csv及test_data.csv用read_csv讀入，前8個特徵分為兩個列表儲存，讀取max及min（除了懷孕次數外其他值為0的不讀入），接下來則是計算平均值後將資料為0(無資料部分)改為平均數做預測，而後每一筆test資料丟進train所建立好的模型計算歐幾里得並同時是否與預測結果相符，最後計算準確率與召回率，以下是計算方式：

預測結果為有病且實際有病記為有-有
預測結果為有病但實際無病記為有-無
預測結果為無病且其實沒病記為無-無
預測結果為無病但其實有病記為無-有

準確率 ： （有-有)+(無-無）/ 全

召回率 ：（有-有）/（有-有)+(無-有)

4.3 Random Forest

一開始把train_data.csv和test_data.csv用read_csv讀入，套用sklearn套件中的完成對資料的標準化，然後透過train_test_split分割成訓練組和測試組，我們設定test占比30%，之後使用RandomForestClassifier訓練模型，最後使用accuracy_score和recall_score得到準確率和召回率。

4.4 高斯貝氏分類器

大致方法與Random Forest雷同，差別在於此分類器使用GaussianNB()的function建立模型。

5. 結論

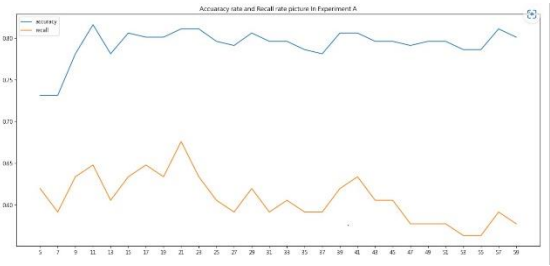
5.1 模擬結果

A. KNN

藍線為準確率，黃線為召回率

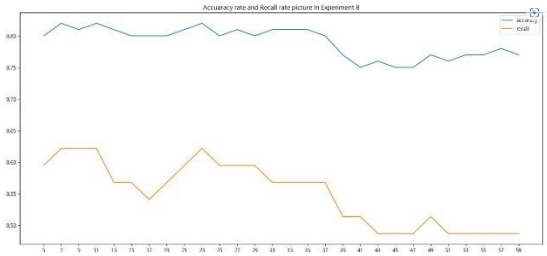
測試在不同的K值下，準確率與召回率會如何變化

實驗A：



```
Accuracy rate(for k = 5) is : 0.7313432835820896
Recall rate(for k = 5) is : 0.6197183098591549
```

實驗B：



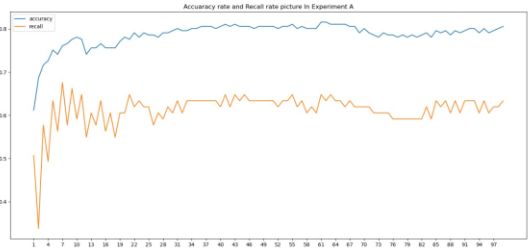
```
Accuracy rate(for k = 5) is : 0.8
Recall rate(for k = 5) is : 0.5945945945945946
```

B. Random Forest

藍線為準確率，黃線為召回率

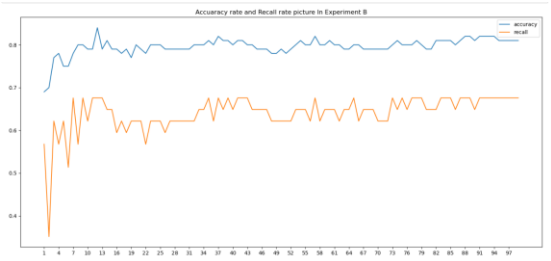
測試在不同的決策樹數量下，準確率與召回率會如何變化

實驗A：



```
Accuracy(tree = 50): 0.8059701492537313
Recall(tree = 50): 0.6338028169014085
```

實驗B：



```
Accuracy(tree = 50): 0.79
Recall(tree = 50): 0.6216216216216216
```

C. 高斯貝氏分類器

實驗A：

```
在實驗A中:
Accuracy: 0.7661691542288557
Recall: 0.5915492957746479
```

實驗B：

```
在實驗B中:
Accuracy: 0.77
Recall: 0.5405405405405406
```

5.2 結論分析

A. KNN

在KNN的部分，可以看到K值增加時準確率及召回率一開始有明顯上升的趨勢，但是到後面K值到了一定的數量後有下滑趨勢也就是overfitting的現象，因此在使用KNN演算法時需要注意是否尋找過多鄰居。

B. Random Forest

Random Forest的狀況我們則是分析建立的樹多寡與準確率和召回率的相關性，透過實驗得知建立多個決策

樹可以讓投票數量選擇更多，進而提升準確率與召回率。

C. 高斯貝氏分類器

由於數據為連續，因此選擇透過高斯貝氏分類器將資料離散化，丟入套件後，可以透過比較outcome=1與outcome=0的兩者機率大小預測測資之outcome。

6. 參考文獻

1. https://www.geeksforgeeks.org/random-forest-regression-in-python/
2. https://ithelp.ithome.com.tw/articles/10278807
3. https://ithelp.ithome.com.tw/articles/10297660?sc=rss.iron
4. Random Forest in Python and coding it with Scikit-learn (tutorial) (data36.com)
5. https://ithelp.ithome.com.tw/m/articles/10224036
6. https://ithelp.ithome.com.tw/articles/10269826