# Artificial Intelligence Final Project – Spring 2017

## Outline

- 執行方法
- 程式架構與流程
- 結果與討論
- 遇到的困難

## 執行方法

make 過後以 “ ./main [Dataset 的名稱(不需加副檔名)] ” 即可 .

P.S. 1. Datasets 需放在 working 目錄下的”dataset”目錄中 .

2. 執行完後即可在”foldedDatatsets”目錄下找到 C.V.分完後的 dataset .

## 程式架構與流程

### Global Variables :

- Data Table : 紀錄每一筆 Data .
- Attribute’s value Table : 紀錄每個 Attribute 各別擁有的 Value 值 .
- Class’s information Table : 紀錄各個 Class 分別為哪些 instance 的 Class .
- Continuous Attribute’s Index : 紀錄哪些 Attribute 是 Continuous 的 .
- Folded Training Data : 紀錄各個 Fold 的 Training Data .
- Folded Testing Data : 紀錄各個 Fold 的 Testing Data .

### Step 01 : Parsing

設定好 Data Table、Attribute’s value Table、Class’s information Table、Continuous Attribute’s Index 這些 Global Variables.

### Step 02 : 10 – Folding for Cross Validation

首先會先計算各個 Class 在每個 Fold 中應該要有的個數，利用 Class’s information Table 來得知各個 Class 在 Data set 中所佔的比例，在轉換成在各個 Fold 應該要有的個數。

接著依據這些個數資訊來進行 Folding，一樣利用 Class’s information Table 來取 Data set 中的 instance，形成十個 folds，並將

這些 folds 儲存下來(Folded Training Data & Folded Testing Data) .

### Step 03 : Naïve Bayesian

依序以 Folded Training Data 進行訓練，Folded Testing Data 進行測試。

Training -

- 針對「各個」Attribute，建立以下 lookup Table :

| Attribute i | Value 1 | Value 2 | … | … | … | Value N |
|---|---|---|---|---|---|---|
| Class 1 | P(V1\|C1) | P(V3\|C3) | … | | | |
| Class 2 | P(V2\|C2) | P(V4\|C4) | … | | | |
| … | … | … | … | | | |
| … | … | … | … | | | |
| Class N | … | | | | | P(CN\|VN) |

每個元素為 P(Attribute i = Value x | Class y )，即在 Class y 下，該 Attribute 值為 Value x 的條件機率。

- 紀錄各個 Class 的 P(Ci)，即 Class Ci 發生的機率 。
- Continuous Attribute 的處理 ：
  - 利用 Continuous Attribute's Index 來得知哪些是數值形態的 Attribute，取出該 Attribute 的所有 Value 計算三個值：
    - first cut point = (second cut point + Min) / 2
    - second cut point = (Max + Min) / 2
    - third cut point = (Max + second cut point) / 2

    因此可以得到四個 Segment, 分別是：
    - [Min ,first cut point ]
    - [first cut point , second cut point]
    - [second cut point , third cut point]
    - [ third cut point,Max ]

    也就是每個 Continuous Attribute 最後會後都會濃縮成四個 Value 值，就是以上的四個區段。

Testing -

利用 Training 出來的 lookup table 進行預測，然後和實際結果比較，計算預測正確的個數，最後輸出正確率。

- Continuous Attribute 的處理 ：

  當遇到 continuous attribute 時，會先將該 Value map 到對應

的區段 Value，再去查找 look up table .

# 結果與討論

## Result

| dataset | | cv1 | cv2 | cv3 | cv4 | cv5 | cv6 | cv7 | cv8 | cv9 | cv10 | avg | p-value |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| adult | NB_Acc | 0.78 | 0.78 | 0.78 | 0.77 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 4.1167E-13 |
| | c4.5_Acc | 0.85 | 0.86 | 0.86 | 0.85 | 0.86 | 0.86 | 0.86 | 0.86 | 0.86 | 0.85 | 0.86 | |
| | | | | | | | | | | | | | |
| car | NB_Acc | 0.72 | 0.66 | 0.89 | 0.71 | 0.86 | 0.85 | 0.76 | 0.72 | 0.84 | 0.74 | 0.77 | 0.135937586 |
| | c4.5_Acc | 0.77 | 0.77 | 0.72 | 0.76 | 0.81 | 0.85 | 0.88 | 0.88 | 0.82 | 0.84 | 0.81 | |
| | | | | | | | | | | | | | |
| isolet | NB_Acc | 0.83 | 0.93 | 0.88 | 0.92 | 0.88 | 0.89 | 0.80 | 0.82 | 0.75 | 0.85 | 0.86 | 3.87555E-06 |
| | c4.5_Acc | 0.70 | 0.81 | 0.79 | 0.77 | 0.86 | 0.79 | 0.69 | 0.70 | 0.67 | 0.70 | 0.75 | |
| | | | | | | | | | | | | | |
| page-blocks | NB_Acc | 0.82 | 0.85 | 0.83 | 0.90 | 0.91 | 0.90 | 0.76 | 0.65 | 0.60 | 0.61 | 0.78 | 0.000405166 |
| | c4.5_Acc | 0.95 | 0.98 | 0.96 | 0.97 | 0.97 | 0.97 | 0.97 | 0.98 | 0.95 | 0.94 | 0.96 | |
| | | | | | | | | | | | | | |
| winequality | NB_Acc | 0.33 | 0.34 | 0.30 | 0.31 | 0.36 | 0.34 | 0.30 | 0.35 | 0.38 | 0.29 | 0.33 | 4.18287E-06 |
| | c4.5_Acc | 0.43 | 0.42 | 0.37 | 0.40 | 0.41 | 0.49 | 0.46 | 0.47 | 0.46 | 0.39 | 0.43 | |

## Discuss

由結果得知，Decision Tree 的準確度大多優於 N.B.，我的猜想可能的原因有 ：

1. 在 continuous attribute 處理上不夠嚴謹，目前是只將整個值域(min ~ max)分成四個區段，若多切幾個區段，也許可以提高預測的準確度 . 或是利用常態分佈方法，將所有 Value 視為是常態分佈，因此若給一個值我們即可反推他的機率為何 .
2. 各個 dataset 中的 attributes 中，多少互相會有一些關聯，不能以各自獨立的角度去看 .

# 遇到的困難

在撰寫的過程中沒有遇到什麼太大的困難，但有些小地方像是使用了蠻多 Nested map/vector 資料結構，導致最後遇到 bug 要 trace 時花了不少時間 .