

# ML lab3

## Probability base learning

0416037 李家安

這次得作業事做 Probability base 的機器學習，基本上是以 Naive Bayes model 為主，分為兩個部分，第一個部分是用小測資，讓我們練習建 PDF，並預測一比 query 的答案，而第二部分主要用比較大量的資料，過程中偏重於資料的預處理，較為麻煩。基本上兩個都是以連續行的 feature 對上離散的 target 作出預測。

### Environment:

Ubuntu 16.04

### Language and Library:

Python 3.5

Pandas

NumPy

SciKit-learn

### How to run it:

第一部分：

```
$ python3 ML_lab3_A.py
```

第二部分：

```
$ python3 ML_lab3_B.py
```

### Code:

第一部分：

將 training data 讀入後，先用 binning 的方式將連續 feature 做分類，接著建出 Frequency Table，其中在建 Frequency Table 時，是使用到 Laplace smoothing 的方式做處理，只後把 testing data 讀入後，以同樣的 binning 方式分類，接著就可以比對 Frequency Table，很快的找到 testing data 的 target 了。

在第一部分，主要有兩個函式，一個是 Classify，負責將連續的 feature 以 binning 的方式分類，另一個則是 Fequency\_Table，負責將 binning 完的 training data 建成 Frequency Table 以利之後的 query 使用。

Classify 的分類方法是以 equal width 的方式分類，主要是我先看過資料後，以我認為比較合理的級距做分類，以簡單的除以級距做分類，並以商數作為類別名稱。

Frequency Table 的建法是將 training data 讀入後，先算出每個 target 總共有幾個，然後用兩層 for 迴圈切開 data，第一層先將不同的 feature 切開，第二層將不同的 target 切開，計算出這個 target 下的 feature 會被分成幾類，每類有幾種，將之記錄在 Frequency Table 上。由於我有使用 Laplace smooth，我的作法是另外紀錄這個 target 下的 feature 最低與最高的 value 是多少，然後把中間的區間以 Laplace 公式全部塞滿。

主函式的部分，我將資料以 pandas.read\_csv 的方式讀入，然後先用 Classify 做 binning，在用 Frequency\_Table 建表，接著把 test data 直接寫在 code 裡，用 pandas.DataFrame

的資料型態存，同樣對 test data 做 Classify，最後用兩層 for 迴圈將需要的機率值從 frequency table 裡抓出來相乘，用 try except 的方式檢查是否還會有 0 value，最後將三個 target 的 value 印出來比較，發現 settler 應該最符合 test data。

```
settler: 3.41833653499e-07      ok: 1.46867342024e-07      solids: 9.53684039117e-08
Best fit: settler
```

第二部分：

第二部分的困難點主要在資料處理，分為兩個部分，第一個要處理的是 attr 的資料，另一個則是 data 的資料，我 data 主要是利用網路工具，先將他轉成 csv 檔，並切成 feature 與 target 兩個部分，再用 pandas 讀入並處理，而 attr 是以字串處理的方式，一行一行讀進來處理。

過程中開了一個 Classify 函式負責處理 attr，將資料用 open 讀入後，for 迴圈的方式一行一行處理，以 regex 的 match 將資料前面的書名部分都清掉，直到遇到 class 後才開始工作，之後把每行讀入，用“,”做分隔再繼續處理，若字串中有 to，將 to 前面得自取最小的當 begin，to 後面取最大的當 end，一個一個將中間的日期都塞入 dict 中當 key，dict 對應到的 value 就會是日期所屬的 class。若字串中沒有 to，則直接塞入 dict 中，同樣 value 會試所屬的 class。

在主函式中，以 pandas 的方式讀入資料，將 feature 中有“?”的 row 全部 drop 掉，接著用 replace 的方式將日期改成剛剛 dict 裡面相對應的 class，同時 test data 做一樣的資料處理。

最後取出 sklearn 中的 GaussianNB 建 modal，用此 modal 對 test data 做預測以得到最終答案。

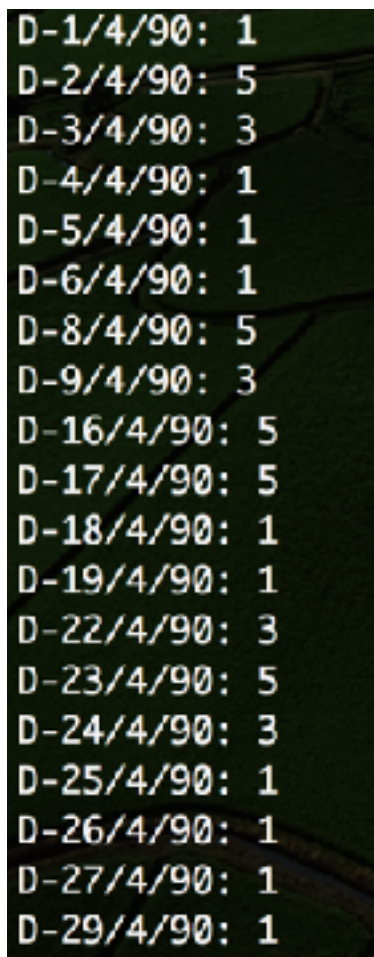
## Result:

第一部分：

如附檔 ML\_lab3\_A\_result.txt 所示

第二部分：

如右圖所示



```
D-1/4/90: 1
D-2/4/90: 5
D-3/4/90: 3
D-4/4/90: 1
D-5/4/90: 1
D-6/4/90: 1
D-8/4/90: 5
D-9/4/90: 3
D-16/4/90: 5
D-17/4/90: 5
D-18/4/90: 1
D-19/4/90: 1
D-22/4/90: 3
D-23/4/90: 5
D-24/4/90: 3
D-25/4/90: 1
D-26/4/90: 1
D-27/4/90: 1
D-29/4/90: 1
```