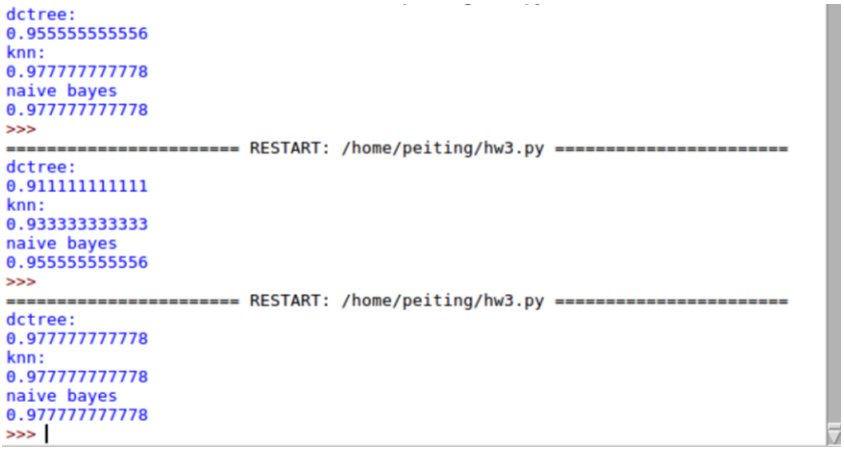
機器學習 hw3 report

Environment: Ubuntu 16.04.3 python3.5 idle

Language: python

Iris data:

Accuracy result:



Used library:

Csv ,os ,sys,numpy

from sklearn import tree

from sklearn.cross_validation import train_test_split

from sklearn import metrics

from sklearn import neighbors

from sklearn.naive_bayes import GaussianNB

code explanation:

kk represents target data

rr represents feature data

每執行一次 random 一次

train_X,test_X,train_y,test_y = train_test_split(rr,kk,test_size = 0.3)

→30% test 70% train

Use train_X(70% feature data),train_y(70% target data) to fit clf then predict

Decision tree:

clf = tree.DecisionTreeClassifier()

Knn : I applied k = 10

knn = neighbors.KNeighborsClassifier(10)

Naïve bayes: because all features in iris_data are continuous(real number)

So I adopt GaussianNB() (calculate pdf)

Analysis of 3 classifier:

我執行了三次，由於 iris data 數據不多，因此可以發現三種分類方式無巨大差別，但仍可以看出 decision tree 有略低於另外兩種的情況，可能是因為 decision tree 每次只會取跟他 feature 最相似的一點當作預測值，但 knn 我設 10 是取 feature 最相近 10 個點去算哪個 Class 的所占機率較大而預測，樣本數較多，而 naïve bayes 則是以 traindata 每個 feature 種類對應到各 class 的機率去算，因為採用 gaussian 所以對於連續性資料在同個 bin size 範圍內會規到同一種 feature，也算是以所有 train data 所生成的機率結果去預測，也比只採用最近點去預測的 decision tree 取樣範圍廣，因此在此測資中有時候比 dctree 的準確性略高。

Forest data:

Accuracy result:

```
==================== RESTART: /home/peiting/hw3_forest.py ====================
decision tree:
0.371794871795
knn:
0.442307692308
[0 1 2 3 4 5]
naive bayes:
0.461538461538
>>>
==================== RESTART: /home/peiting/hw3_forest.py ====================
decision tree:
0.25
knn:
0.429487179487
naive bayes:
0.275641025641
>>>
==================== RESTART: /home/peiting/hw3_forest.py ====================
decision tree:
0.384615384615
knn:
0.455128205128
naive bayes:
0.333333333333
>>>
==================== RESTART: /home/peiting/hw3_forest.py ====================
decision tree:
0.365384615385
knn:
0.423076923077
naive bayes:
0.237179487179
>>>
==================== RESTART: /home/peiting/hw3_forest.py ====================
decision tree:
0.320512820513
knn:
0.442307692308
naive bayes:
0.25641025641
>>>
==================== RESTART: /home/peiting/hw3_forest.py ====================
decision tree:
0.326923076923
knn:
0.410256410256
naive bayes:
0.346153846154
>>>
```

*中間[0 1 2 3 4 5]為測試時多印的

Used library:

Csv ,os ,sys,numpy
from sklearn import tree
from sklearn.cross_validation import train_test_split
from sklearn import metrics
from sklearn import neighbors
from sklearn.naive_bayes import GaussianNB
from sklearn.naive_bayes import MultinomialNB

code explanation:

knn/dctree : same as code in iris_data

naïve_bayes:

因為 forest_data 裡面 feature 0 1 2 3 為 categorical，而其他為 continuous，所以將離散資料用 multinomialnb fit (alpha = 1.0 -> laplace smooth)，而連續資料用 gaussian(calculate pdf) fit，之後用 predict_proba 得到 test_X 每一個測資下去預測會得到的各種結果的機率(mnb_p ,gnb_p)

根據講義條件機率算法

$$M(\mathbf{q}) = \underset{Y \in \mathbf{T}}{\operatorname{argmax}} P(Y) \prod_{i=1}^{m} P(X_i \mid Y)$$

由於 feature 分開來算，所以最後再乘回去就是 mix 過後的條件機率，因此我把 mnb_p[i] gnb_p[i]裡的各項相乘 (i 為 0~測資 test_X 總數)

也就是 mnb 預測 class[j]的機率*gnb 預測 class[j]的機率 (j 為 0~5： 代表 class 種類)

因為最後是找最大值所以忽略多乘的 p(Y)

因此相乘以後找機率最大的那一項 class 便為此測資的預測值

Analysis of 3 classifier:

執行共六次，可以發現各種分類準確率均比 iris_data 低很多，除了 forest_fires 複雜度比 iris_data 高之外，我們也可以推測 forest feature 與 target 關聯度應比 irisdata 來得低，因而造成準確率低很多。

而根據三種 classfier 的差異，大部分的結果中，均為 knn>dctree>naïve bayes，而在少數的結果中，naïve bayes 準確率高於 dctree，甚至有一筆高於 knn，由此看出 naïve bayes 預測結果較為不穩，並且大部分時候偏低

根據以上結果我推測原因:

1. naive bayes 假設各 feature 獨立去做分析，因此沒考慮到各 feature 之間的相關性，不論是 knn 或者是 dctree，都是分類完 feature a 再去 feature a 中的集合去分類下一個 feature，因此準確率可能會受測資分布所影響
   (knn>dctree 原因同 analysis of iris data)
2. 我可能寫錯了

不過如果我沒有將 gaussian & multinomial NB mix，而單純只用其中一種的話，準確率似乎更低。