**Machine Learning - assignment #3**

0416045 孫嘉妤

**程式運行結果**

本次作業附上的code分為兩個檔案：iris.py 以及 forestfires.py，兩者分別依序對其dataset建立decision tree/kNN/naïve Bayes的classification(iris)/  
regression(forestfires) model。其中數個xx\_result.txt是詳細的測試結果（如以下截圖）。可讓程式執行多次，每次重新隨機選取70% train set及30% test set，得到多次建立model的平均準確率。

**iris dataset**

|  |  |
| --- | --- |
|  |  |
| iris.py會顯示每次測試三種model的準確度及執行n次的平均準確度 | 若選擇顯示詳細結果，可看到每次測試三種model的預測狀況 |

**model 比較**

執行 1000 次的accuracy平均如下：

* decision tree ( 0.944689 )
  + CART (Classification and Regression Trees)
* kNN ( 0.962311 )
  + n = 5, distance metric：Euclidean metric
* naïve Bayes ( 0.952756 )
  + pdf：Gaussian Naive Bayes algorithm

基本上這個dataset用三種model預測的準確率都還不錯，約達95%。其中 kNN (n=5) 的表現為三者中最佳，或許是由於decision tree在建構時有時會受少數較特殊的資料影響，而naïve Bayes忽略了某些特徵間可能的相互依賴性，因此在這個dataset中參考多個鄰近點的kNN準確率略高於其他兩者。

**forest fires dataset**

|  |  |
| --- | --- |
|  |  |
| foresetfires.py會顯示每次測試三種model預測area與實際area的差，以及執行n次的平均 | 若選擇顯示詳細結果，可看到每次測試三種model的預測狀況（area為取log處理的結果） |

**model 比較**

執行 50 次的預測area與實際area之（log10）差距平均如下：

* decision tree（0.0808476，約等於面積差1.2倍）
  + CART (Classification and Regression Trees)
* kNN（1.28488，約等於面積差19倍）
  + n = 5
  + distance metric：Euclidean metric
* naïve Bayes ( 0.4133036，約等於面積差2.59倍)
  + pdf：Gaussian Naive Bayes
  + category features：multinomial Naive Bayes

這個dataset用三種model預測的結果，準確性大致上是decision tree > naïve Bayes > kNN，且decision tree明顯優於其他兩者。會有這樣的結果，我猜測是因為area取log處理的緣故，使得一旦沒有發生火災（燃燒面積＝0）便會與有發生火災的面積有一定差距（取log的原因如dataset說明所述）。而當test data為無發生火災時，三者中最有可能得到預測面積為零的decision tree便會得到較小的預測差距。計算data每維度整體機率的naïve Bayes則應該能相對得到一個不見得最準確但也不至於造成極大偏差的預測結果；而kNN只要neighbor中有發生燃燒面積較大的火災，預測結果便會得到較大的偏差。此外，此dataset可能是因維度較多，其執行時間（尤其是在naïve Bayes的部份）較iris data久上許多，因此測試的次數較iris data少。

**程式語言**

|  |  |
| --- | --- |
|  | python3 |

從讀取資料、建立model及預測皆使用 python3。其中三種model都是call library，若要執行需要先安裝數個package（主要是sklearn）。我自己的執行環境是3.5.2及系上工作站的3.6.2。

**使用程式庫**

iris.py：

|  |
| --- |
| // I/O  import sys  // 數學計算  import math  import numpy as np  from numpy import array  // model建立與 predict  from sklearn import tree  from sklearn.neighbors import KNeighborsClassifier  from sklearn.naive\_bayes import GaussianNB  // shuffle  from random import shuffle |

foresetfires.py：

|  |
| --- |
| // I/O  import csv  import sys  // 數學計算  import math  import numpy as np  from numpy import array  // model建立與 predict  from sklearn.tree import DecisionTreeRegressor // decision tree  from sklearn.preprocessing import normalize // kNN  from sklearn.neighbors import KNeighborsRegressor // kNN  from sklearn.naive\_bayes import GaussianNB // naïve Bayes  from sklearn.naive\_bayes import MultinomialNB // naïve Bayes  // shuffle  from random import shuffle |

**開發環境**

程式撰寫主要使用文字編輯器，完成後在 windows 10運行（Windows Subsystem for Linux）。

**code實作**

**Iris dataset**

這部份由於資料皆是連續的，整理好讀入的data後直接call library：

* decision tree ( [DecisionTreeClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html) )
  + scikit-learn的tree使用CART (Classification and Regression Trees) algorithm ( [# 1.10.6](http://scikit-learn.org/stable/modules/tree.html) )
* kNN ( [KNeighborsClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier) )
  + 參數：n=5, algorithm="kd\_tree"
  + distance metric：scikit-learn預設為Euclidean metric ( [# metric](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier) )
* naïve Bayes ( [GaussianNB](http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB) )
  + pdf：Gaussian Naive Bayes algorithm ( [# 1.9.1](http://scikit-learn.org/stable/modules/naive_bayes.html) )

**forest fire dataset**

先處理原本是category features的month及day：

day視為category features，由function day\_to\_int( ) 直接將星期一至日轉為 0~6 的數字；month我認為並不完全是category的feature，尤其在算距離時月份間其實是具有某種相關性的（像十二月其實離一月很近，若直接轉為1~12會得到相反的結果），但難以設計出一種能表現上述特質的月份數字轉換，考慮折衷過後仍是將1~12月轉為數字0~11。我覺得比起直接轉成數字，較適當的方法是在計算月份距離時不直接相減，另外定義其他的計算方式。

area：根據 dataset 的說明（the burned area of the forest is very skewed towards 0.0, thus it may make sense to model with the logarithm transform)，因此area的預測上以取log10處理。

normalize：在建kd-Tree前先做資料的normalize（使用此[normalize](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html)）。decision tree 和naïve Bayes並不受正規化影響，所以只有kd-Tree有做normalize處理。

naïve Bayes：先用gaussian NB model fit category features，以及用multinomial NB model fit continuous features，分別建出兩個naïve Bayes model，最後將這兩個model預測的結果再建成一個naïve Bayes model。這部分的code如下：

|  |
| --- |
| # naïve Bayes  # gaussian NB model on the continuous part  gNB = GaussianNB()  y\_train = np.asarray(forestfire\_nb['target'], dtype="|S6")  gNB.fit(forestfire\_nb['cont\_data'], y\_train)  # multinomial NB model on the categorical part with Laplace smoothing  mNB = MultinomialNB(alpha=1.0)  mNB.fit(forestfire\_nb['cat\_data'], y\_train)  # get predict for two NB model  hy\_pred = []  for i in range(len(forestfire\_nb['cont\_data'])):  gp = gNB.predict([forestfire\_nb['cont\_data'][i]])  mp = mNB.predict([forestfire\_nb['cat\_data'][i]])  g = float(gp[0].decode('UTF-8'))  m = float(mp[0].decode('UTF-8'))  p = [g, m]  hy\_pred.append(p)  hy\_pred = array(hy\_pred)  hNB = GaussianNB()  hNB.fit(hy\_pred, y\_train) |

* decision tree ( [DecisionTreeRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html) )
  + scikit-learn的tree使用CART (Classification and Regression Trees) algorithm ( [# 1.10.6](http://scikit-learn.org/stable/modules/tree.html) )
* kNN ( [KNeighborsRegressor](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html) )
  + 參數：n=10, algorithm="kd\_tree"
  + distance metric：scikit-learn預設為Euclidean metric ( [# metric](http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier) )
* naïve Bayes ( [GaussianNB](http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB)、[MultinomialNB](http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html" \l "sklearn.naive_bayes.MultinomialNB) )
  + laplace smooth：set the smoothing priors α = 1 is Laplace smoothing ( [# 1.9.2](http://scikit-learn.org/stable/modules/naive_bayes.html) )
  + pdf：Gaussian Naive Bayes algorithm ( [# 1.9.1](http://scikit-learn.org/stable/modules/naive_bayes.html) )