CIS 419/519 Introduction to Machine Learning Assignment 1

Kuan-Cheng Chiu

Hello world!
This is a **bold** text. This is a underline. And this is a ***composed*** style.

PART I: PROBLEM SET

1 Chapter 1

1.1 chapter 1 sub

Some text for chapter one.

Figure 1: a figure

2 Chapter 2

According to Figure 1, the space is.

```
'''
    TEMPLATE FOR MACHINE LEARNING HOMEWORK
```

```
    AUTHOR Eric Eaton, Chris Clingerman
'''

import numpy as np
import matplotlib.pyplot as plt

from sklearn import tree
from sklearn.metrics import accuracy_score



def evaluatePerformance():
    '''
    Evaluate the performance of decision trees,
    averaged over 1,000 trials of 10-fold cross validation

    Return:
      a matrix giving the performance that will contain the
    following entries:
      stats[0,0] = mean accuracy of decision tree
      stats[0,1] = std deviation of decision tree accuracy
      stats[1,0] = mean accuracy of decision stump
      stats[1,1] = std deviation of decision stump
      stats[2,0] = mean accuracy of 3-level decision tree
      stats[2,1] = std deviation of 3-level decision tree

    ** Note that your implementation must follow this API**
    '''

    # Load Data
    filename = 'data/SPECTF.dat'
    data = np.loadtxt(filename, delimiter=',')
    X = data[:, 1:]
    y = np.array([data[:, 0]]).T
    n,d = X.shape

    # shuffle the data
    idx = np.arange(n)
    np.random.seed(13)
    np.random.shuffle(idx)
    X = X[idx]
    y = y[idx]

    # split the data
    Xtrain = X[1:101,:]  # train on first 100 instances
    Xtest = X[101:,:]
    ytrain = y[1:101,:]  # test on remaining instances
    ytest = y[101:,:]

    # train the decision tree
    clf = tree.DecisionTreeClassifier()
    clf = clf.fit(Xtrain,ytrain)

    # output predictions on the remaining data
    y_pred = clf.predict(Xtest)

    # compute the training accuracy of the model
```

```
    meanDecisionTreeAccuracy = accuracy_score(ytest, y_pred)


    # TODO: update these statistics based on the results of your
    experiment
    stddevDecisionTreeAccuracy = 0
    meanDecisionStumpAccuracy = 0
    stddevDecisionStumpAccuracy = 0
    meanDT3Accuracy = 0
    stddevDT3Accuracy = 0

    # make certain that the return value matches the API
    specification
    stats = np.zeros((3,2))
    stats[0,0] = meanDecisionTreeAccuracy
    stats[0,1] = stddevDecisionTreeAccuracy
    stats[1,0] = meanDecisionStumpAccuracy
    stats[1,1] = stddevDecisionStumpAccuracy
    stats[2,0] = meanDT3Accuracy
    stats[2,1] = stddevDT3Accuracy
    return stats



# Do not modify from HERE...
if __name__ == "__main__":

    stats = evaluatePerformance()
    print "Decision Tree Accuracy = ", stats[0,0], " (", stats
    [0,1], ")"
    print "Decision Stump Accuracy = ", stats[1,0], " (", stats
    [1,1], ")"
    print "3-level Decision Tree = ", stats[2,0], " (", stats[2,1],
     ")"
# ...to HERE.
```

$A_x + 1$