

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.

```
1 '''  
2 TEMPLATE FOR MACHINE LEARNING HOMEWORK
```

```

3     AUTHOR Eric Eaton, Chris Clingerman
4     '''
5
6     import numpy as np
7     import matplotlib.pyplot as plt
8
9     from sklearn import tree
10    from sklearn.metrics import accuracy_score
11
12
13
14    def evaluatePerformance():
15        '''
16        Evaluate the performance of decision trees,
17        averaged over 1,000 trials of 10-fold cross validation
18
19        Return:
20        a matrix giving the performance that will contain the
21        following entries:
22            stats[0,0] = mean accuracy of decision tree
23            stats[0,1] = std deviation of decision tree accuracy
24            stats[1,0] = mean accuracy of decision stump
25            stats[1,1] = std deviation of decision stump
26            stats[2,0] = mean accuracy of 3-level decision tree
27            stats[2,1] = std deviation of 3-level decision tree
28
29        ** Note that your implementation must follow this API**
30        '''
31
32        # Load Data
33        filename = 'data/SPECTF.dat'
34        data = np.loadtxt(filename, delimiter=',')
35        X = data[:, 1:]
36        y = np.array([data[:, 0]]).T
37        n,d = X.shape
38
39        # shuffle the data
40        idx = np.arange(n)
41        np.random.seed(13)
42        np.random.shuffle(idx)
43        X = X[idx]
44        y = y[idx]
45
46        # split the data
47        Xtrain = X[1:101,:] # train on first 100 instances
48        Xtest = X[101:,:]
49        ytrain = y[1:101,:] # test on remaining instances
50        ytest = y[101:,:]
51
52        # train the decision tree
53        clf = tree.DecisionTreeClassifier()
54        clf = clf.fit(Xtrain,ytrain)
55
56        # output predictions on the remaining data
57        y_pred = clf.predict(Xtest)
58
59        # compute the training accuracy of the model

```

```

59     meanDecisionTreeAccuracy = accuracy_score(ytest, y_pred)
60
61
62     # TODO: update these statistics based on the results of your
        experiment
63     stddevDecisionTreeAccuracy = 0
64     meanDecisionStumpAccuracy = 0
65     stddevDecisionStumpAccuracy = 0
66     meanDT3Accuracy = 0
67     stddevDT3Accuracy = 0
68
69     # make certain that the return value matches the API
        specification
70     stats = np.zeros((3,2))
71     stats[0,0] = meanDecisionTreeAccuracy
72     stats[0,1] = stddevDecisionTreeAccuracy
73     stats[1,0] = meanDecisionStumpAccuracy
74     stats[1,1] = stddevDecisionStumpAccuracy
75     stats[2,0] = meanDT3Accuracy
76     stats[2,1] = stddevDT3Accuracy
77     return stats
78
79
80
81 # Do not modify from HERE...
82 if __name__ == "__main__":
83
84     stats = evaluatePerformance()
85     print "Decision Tree Accuracy = ", stats[0,0], " (", stats
        [0,1], ")"
86     print "Decision Stump Accuracy = ", stats[1,0], " (", stats
        [1,1], ")"
87     print "3-level Decision Tree = ", stats[2,0], " (", stats[2,1],
        ")"
88 # ...to HERE.

```

$A_x + 1$