CIS 419/519 Introduction to Machine Learning Assignment 1

Kuan-Cheng Chiu

Hello world!

This is a **bold** text. This is a $\underline{\text{underline}}$. And this is a $\underline{\text{composed}}$ style.

PART I: PROBLEM SET

1 Chapter 1

1.1 chapter 1 sub

Some text for chapter one.

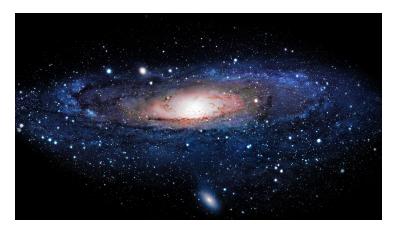


Figure 1: a figure

2 Chapter 2

According to Figure 1, the space is.

TEMPLATE FOR MACHINE LEARNING HOMEWORK

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

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

 $A_x + 1$