Data mining 4th Report  
(Student Number: 1651046, Name: Daiki Kubo)

Practice "Decision Tree", "Random Forest" and "Naive Bayes Classifier" using 3 different data and tools of your choice (R or other) and make some reports within maximum 6 pages (3 sheets).

**Answer.**

1. Decision Tree (tool: python, data: iris dataset)

Source Code

----------------------------------------------------------------------------

#coding:utf-8

from sklearn.datasets import load\_iris

from sklearn import tree

from graphviz import Digraph

from sklearn.externals.six import StringIO

import pydot

def main():

# iris dataset

iris = load\_iris()

# make a model (decision tree)

dtc = tree.DecisionTreeClassifier(max\_depth=3)

dtc = dtc.fit(iris.data, iris.target)

# output

dot\_data = StringIO()

tree.export\_graphviz(dtc, out\_file=dot\_data,

class\_names=iris.target\_names,

feature\_names=iris.feature\_names)

graph = pydot.graph\_from\_dot\_data(dot\_data.getvalue())

print graph

# visualization with a graph

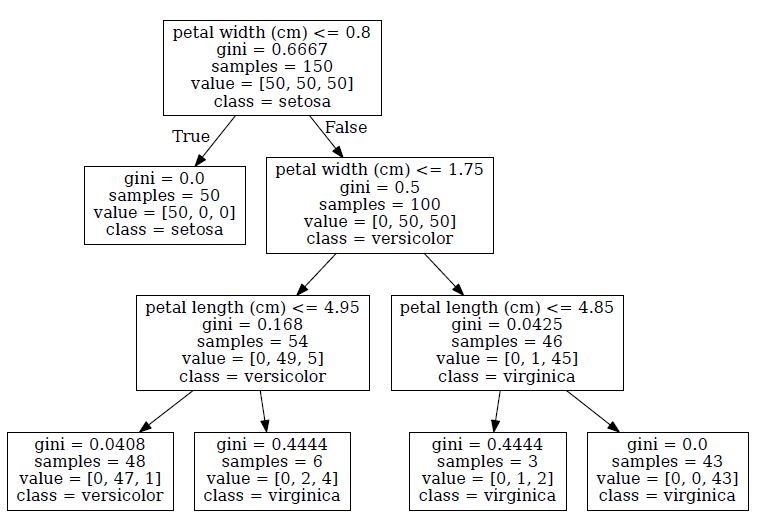
graph[0].write\_pdf("graph.pdf")

if \_\_name\_\_ == '\_\_main\_\_':

main()

----------------------------------------------------------------------------

Result



1. Random Forest (tool: python, data: mnist dataset)

Source Code

----------------------------------------------------------------------------

#coding:utf-8

import numpy as np

from sklearn.datasets import fetch\_mldata

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report

from sklearn.metrics import accuracy\_score

def main():

# load the data

mnist = fetch\_mldata('MNIST original')

mnist.data = mnist.data.astype(np.float32)

mnist.data /= 255

mnist.target = mnist.target.astype(np.int32)

N = 60000

x\_train, x\_test = np.split(mnist.data, [N])

y\_train, y\_test = np.split(mnist.target, [N])

# learn

rfc = RandomForestClassifier()

rfc.fit(x\_train, y\_train)

# test

x\_test, y\_test

pred = rfc.predict(x\_test)

print "classification report"

print classification\_report(y\_test, pred, target\_names=map(str, range(10)))

print "accuracy"

print accuracy\_score(y\_test, pred)

if \_\_name\_\_ == '\_\_main\_\_':

main()

----------------------------------------------------------------------------

Result

classification report

precision recall f1-score support

0 0.95 0.99 0.97 980

1 0.99 0.99 0.99 1135

2 0.92 0.96 0.94 1032

3 0.92 0.93 0.93 1010

4 0.94 0.95 0.95 982

5 0.94 0.93 0.93 892

6 0.97 0.96 0.96 958

7 0.97 0.95 0.96 1028

8 0.94 0.90 0.92 974

9 0.95 0.92 0.93 1009

avg / total 0.95 0.95 0.95 10000

accuracy

0.9482

1. Naive Bayes Classifier (tool: python, data: digits dataset)

Source Code

----------------------------------------------------------------------------

#coding:utf-8

from sklearn import datasets

from sklearn.naive\_bayes import GaussianNB

def main():

# load the data

digits = datasets.load\_digits()

# make a model

gnb = GaussianNB()

y\_pred = gnb.fit(digits.data, digits.target).predict(digits.data)

print("Number of mislabeled points out of a total %d points : %d"

% (digits.data.shape[0],(digits.target != y\_pred).sum()))

if \_\_name\_\_ == '\_\_main\_\_':

main()

----------------------------------------------------------------------------

Result  
 Number of mislabeled points out of a total 1797 points : 255