import numpy as np

from numpy import genfromtxt

import collections

from collections import Counter

import csv

def CalculateBayes(j, m, std, trainingSet, counts):

exponent = m[j] / 2 + (2 \* std[j])

print exponent

#Read in the file to an array

filename = "smallDataSet.csv"

dataset = genfromtxt(filename, delimiter=',', dtype=int)

print len(dataset)

#Calculates the number of attributes in the file

with open(filename, 'r') as f:

reader = csv.reader(f, delimiter=',', skipinitialspace=True)

firstRow = next(reader)

noOfColumns = len(firstRow)

noOfAttributes = noOfColumns

#Split the dataset into 60% training and 40% test data

split = len(dataset)\*0.6

trainingSet, testSet = np.split(dataset, [split])

#Calculate number of training and test data

totalTraining = len(trainingSet)

totalTest = len(testSet)

##NOT NEEDED

p = np.sum(trainingSet, axis = 0)

#Calculate the mean of each attribute

m = np.mean(trainingSet, axis = 0)

#Calculate the standard deviation of each attribute

std = np.std(trainingSet, axis = 0)

transposeTraining = np.transpose(trainingSet, axes=None)

print transposeTraining

j = 0

while j < (noOfAttributes):

unique, counts = np.unique(transposeTraining[j], return\_counts=True)

CalculateBayes(j, m, std, trainingSet, counts)

j = j + 1

print unique, counts

#c = trainingSet.Counter(1)

#print c

#p\_category = number of categories / trainingsize

#p\_category \*= how many of an attribute + m \* p / how many categories + m