CSE 4108: Artificial Intelligence Lab, Spring 2018 Spring 2018

Assignment 6

Gaussian Naive Bayes

Submitted By:
Sarwar Saif
ID – 15.01.04.091
Email: saifmu6@gmail.com
Year: 4th | Semester: 1st | Section:B2
Date: 27-October-2018

Submitted To:
Mr. Md. Wasi Ul Kabir
Assistant Professor
Mir Imtiaz Mostafiz
Lecturer

```
# -*- coding: utf-8 -*-
"""
Created on Fri Oct 26 19:56:43 2018

@author: saif
"""
import csv #for csv files
import pandas as pd #for reading data from csv file and operating on them
import random as rd
from matplotlib import pyplot as plt
import random
from random import shuffle
import numpy as np
data_file = "banknote.csv"



def splitDataset(dataset, splitRatio):
    trainSize = int(len(dataset) * splitRatio)
    trainSet = []
    copy = list(dataset)
    while len(trainSet) < trainSize:
        index = random.randrange(len(copy))
        trainSet.append(copy.pop(index))
    return [trainSet, copy]


def calculate_gaussian_probability(x,mu,sigma):
    result = np.exp(-((x-mu)**2)/(2.00*(sigma**2)))
    result /= ((2.00*3.1416*(sigma**2))**0.5)
    return result


def calculate_mu_sigma(dataset,cls):
    m1=[]
    m2=[]
    m3=[]
    m4=[]
```

```
for data in dataset:
    if data[4]==cls:
      m1.append(data[0])
      m2.append(data[1])
      m3.append(data[2])
      m4.append(data[4])


  return [np.mean(np.array(m1)) ,
np.mean(np.array(m2)),
np.mean(np.array(m3)),
np.mean(np.array(m4))],[np.std(np.array(m
1)) , np.std(np.array(m2)),
np.std(np.array(m3)), np.std(np.array(m4))]


if __name__ =='__main__':
  """
  Reading data from file and plotting them.
  """
  df = pd.read_csv(data_file)
  datalist= df.values.tolist()


  """
  Split the dataset in 80%-20% proportion
randomly
  """
  splitRatio = 0.80
  train, test = splitDataset(datalist, splitRatio)
  mean1=[]
  std1=[]
  mean2=[]
  std2=[]
  mean1,std1=calculate_mu_sigma(train,0)
  mean2,std2=calculate_mu_sigma(train,1)
```

```
  result=[]
  for data in test:
    prob1=1.0
    prob2=1.0
    for x in range(0,3):

prob1*=calculate_gaussian_probability(int(data[x]
),mean1[x],std1[x])

prob2*=calculate_gaussian_probability(int(data[x]
),mean2[x],std2[x])


    if prob1>prob2:
      result.append(0)
    else:
      result.append(1)


  mm0=0
  mm1=0
  acc=0.0
  for i in range(len(test)):
    e=abs(int(test[i][4])-int(result[i]))


    if e==0:
      mm0+=1
    else:
      mm1+=1


  acc=(mm0/(mm0+mm1)) * 100


  print("Accuracy ")
  print(acc) #More than 80%
```