Date 22/12/2020

Expt. No. 05

Page No. 11

Milind Bijukumar - 4MT17CS058

* Write a program to implement the naive Bayesian classifier for a simple training dataset stored as a .csv file Compute the accuracy of the classifier, considering few test dataset.

*

```
import csv, random, math
import statistic as st
def loadcsv(filename):
    line = csv.reader(open(filename, "r"))
    dataset = list(line)
    for i in range (len(dataset)):
        dataset[i] = [float(x) for x in dataset[i]]
    return dataset

def splitDataset (dataset, splitRatio):
    testsize = int (len) (dataset) * splitRatio );
    trainset = list(dataset)
    testset = []
    while len(testset) < testsize:
        index = random. randrange (len(trainset))
        testSet. append (trainset, pop(index))
    return [trainset, testset]

def seperate By class (dataset):
    seperated = { }
    for i in range (len(dataset)) :
        x = dataset[i]
```

Teacher's Signature _____

```
        if x[-1] not in seperated:
            seperated[x[-1]] = []
        Seperated[x[-1]].append(x)
    return Seperated

def Compute_mean_std(dataset):
    mean_std = [(st.mean(attribute), st.stdev(attribute))
    for attribute in zip(*dataset)]
    del mean_std[-1]
    return mean_std

def SummarizeByClass(dataset):
    Seperated = seperateByClass(dataset),
    Summary = {}
    for classValue, instance in Seperated.items():
        Summary[classValue] = Computer_mean_std(instance)
    return Summary.

def estimateProbability(x, mean, stdev):
    exponent = math.exp(-(math.pow(x-mean,2)/(2*math.
                pow(stdev,2))))

def CalculateClassProbabilities(summaries, testVector):
    p={}
    for classValue, classSummaries in Summaries.items():
        p[classValue] = 1
        for i in range(len(classSummaries)):
            mean, stdev = classSummaries[i],
            x=testVector[i]
            p[classValue] *= estimateProbability(x, mean, stdev);
    return p
```

Teacher's Signature _____

```
def predict(summaries, testVector):
    all_p = calculateClassProbabilities(summaries, testVector)
    bestLabel, bestProb = None, -1
    for lbl, p in all_p.items():
        if bestLabel is None or p > bestProb:
            bestProb = p
            bestLabel = lbl
    return bestLabel.
def performClassification(summaries, testSet):
    predictions = []
    for i in range(len(testSet)):
        result = predict(summaries, testSet[i])
        predictions.append(result)
    return predictions.
def getAccuracy(testSet, predictions):
    correct = 0
    for i in range(len(testSet)):
        if testSet[i][-1] == predictions[:]:
            correct += 1
    return (correct/float(len(testSet))) * 100.0

dataset = local csv('C:/Users/Muli/OneDrive/Desktop/Mike/4MT17CS058-
                    Milind/diabetes.csv')
print('Pima Indians Diabetes Data set loaded...')
print('Total instance available :', len(dataset))
print('Total attribute present :', len(dataset[0]-1))
print('First Five instance of data set')
for i in range(5):
    print(i+1, ':', dataset[i])
```

Teacher's Signature _____

```
splitRatio = 0.2
trainingset, testset = splitDataset(dataset, splitRatio)
print('\n Dataset is split into training set and testing set');
print('Training example = {0} \n Testing example = {1}'.format
                    (len(trainingset), len(testset)))
summaries = SummarizeByClass(trainingset);
predictions = performClassification(summaries, testset)
accuracy = getAccuracy(testset, predictions)
print('\n Accuracy of the Naive Bayesian classifier is:',
                    accuracy)
```

Teacher's Signature

output

Pima Indian Diabetes Dataset loaded ...

Total Instance available : 768

Total Attribute present : 8

First Five Instance of dataset :

1: [6.0, 148.0, 72.0, 35.0, 0.0, 33.6, 0.627, 50.0, 1.0]

2: [1.0, 85.0, 66.0, 29.0, 0.0, 26.6, 0.351, 31.0, 0.0]

3: [8.0, 183.0, 640, 0.0, 0.0, 23.3, 0.674, 32.0, 1.0]

4: [1.0, 89.0, 66.0, 23.0, 94.0, 28.1, 0.167, 21.0, 0.0]

5: [0.0, 137.0, 40.0, 35.0, 168.0, 43.1, 2.288, 33.0, 1.0]

Dataset is Split into training and testing set

Training example = 615

Testing example = 153

Accuracy of the Naive Bayesian Classifier is

75.16339869281046.