Machine Learning Homework 1\_Report

2017112167 이수진

**1. Write a Python program to count the number of strings where the string length is 3 or more and the first and last character are same from a given list of strings.**

**Input List : ['cabc', 'xyza', 'abbc', '13221']**

**Output : 2**

<PROGRAM CODE>

#1

input\_list =['cabc', 'xyza','abbc','13221']

count=0

for x in range(len(input\_list)):

if len(input\_list)>=3:

if input\_list[x][0]==input\_list[x][-1]:

count=count+1

print(count)

<RESULT>

스크린샷이(가) 표시된 사진

자동 생성된 설명

**2. Write a Python program to get a list, sorted in increasing order by the first element in each tuple (inner list) from a list.**

Input List : [[2, 6], [1, 2], [3, 4], [5, 3], [4, 1]]

Output : [[1, 2], [2, 6], [3, 4], [4, 1], [5, 3]]

<PROGRAM CODE>

#2

input\_list = [[2, 6], [1, 2], [3, 4], [5, 3], [4, 1]]

input\_list.sort()

print(input\_list)

<RESULT>

스크린샷이(가) 표시된 사진

자동 생성된 설명

**\* Select ONE arff file from e-class. Change it to csv file. The csv file must contain numbers and/or strings only, each of which is separated by commas. In dong so, you have to modify arff file by removing header part (% and @ part) of the data.**

**3. Write Python code for the following tasks**

**1) read csvfile into a two dimension list (called “a\_list”)**

**e.g.: csvfile=**

**1 0 2 3 1**

**0 1 1 2 0**

**0 1 0 1 1**

**0 0 2 3 1**

**a\_list=[[1,0,2,3,1], [0,1,1,2,0], [0,1,0,1,1], [0,0,2,3,1]]**

<PROGRAM CODE>

#3-1

import pandas as pd

import numpy as np

flags\_data = pd.read\_csv('/Users/soojinlee/python/flags2.csv',header=0)

a\_list = np.array(flags\_data)

a\_list

<RESULT>

스크린샷이(가) 표시된 사진

자동 생성된 설명

**2) show the number of columns(attributes) and number of rows(records), respectively.**

<PROGRAM CODE>

#3-2

column\_num = len(a\_list[0])

row\_num = len(a\_list)

print("the number of columns :",column\_num ,"\nthe number of rows :",row\_num)

<RESULT>

스크린샷이(가) 표시된 사진

자동 생성된 설명

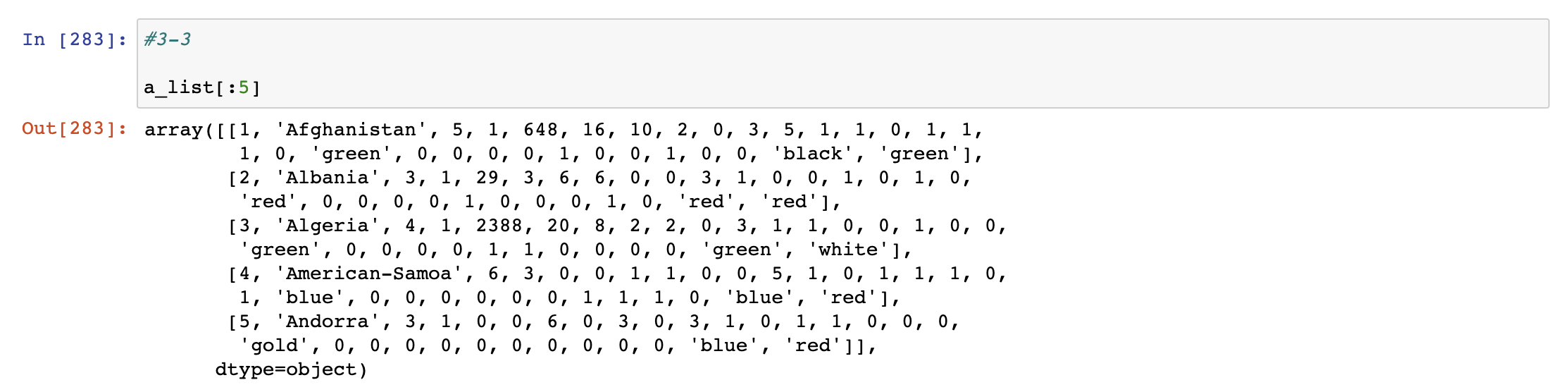
**3) write a Python program that shows the first 5 rows from the “a\_list”.**

<PROGRAM CODE>

#3-3

a\_list[:5]

<RESULT>



**4) write a Python program which randomly shuffles ‘a\_list’ data**

<PROGRAM CODE>

#3-4

import numpy as np

import copy

shuffled\_list = copy.copy(a\_list)

np.random.shuffle(shuffled\_list)

shuffled\_list

<RESULT>

스크린샷이(가) 표시된 사진

자동 생성된 설명

4. Using the “a\_list” in question 3. write Python code for the following tasks

1) given a column(attribute) number, write a program that shows the values of the column.

<PROGRAM CODE>

#4-1

import numpy as np

column\_x=input("put a column number : ")

column\_values=a\_list[:,int(column\_x)-1:int(column\_x)]

column\_values\_1d = column\_values.ravel()

column\_values\_1d

<RESULT>

스크린샷이(가) 표시된 사진

자동 생성된 설명

**2) show the reversed elements of q. 1) (We don’t actually change the values a\_list)**

<PROGRAM CODE>

#4-2

column\_values\_1d[::-1]

<RESULT>

스크린샷이(가) 표시된 사진

자동 생성된 설명

**5. Using the “a\_list”, write Python code for the following tasks**

**1) define a function “divide\_train\_test(in\_list, prop)” function where**

**input: 1) in\_list: a 2D list, 2) prop: proportion of training data**

**output: train\_data (first “prop” percent of in\_list), test\_data (the rest of in\_list)**

<PROGRAM CODE>

#5-1

def divide\_train\_test(in\_list,prop):

train\_ind = int(len(in\_list)\*prop)

train\_data = in\_list[:train\_ind]

test\_data = in\_list[train\_ind:]

return [train\_data, test\_data]

<RESULT>

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

**2) run divide\_train\_test(a\_list, prop) TWO times using prop=0.7, 0.9, respectively, and show the result.**

**e.g.: divide\_train\_test([[1,2,3], [5,1,8], [8,5,2], [0,3,6], [1,7,3]], 0.8)**

**returns [ [ [1,2,3], [5,1,8], [8,5,2] ], [ [0,3,6], [1,7,3] ] ]**

**# train\_data test\_data**

<PROGRAM CODE>

#5-2

print(divide\_train\_test(a\_list,0.7))

print(divide\_train\_test(a\_list,0.9))

<RESULT>

스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

**6. Write Python code for the tasks.**

**1) define a function “min\_max\_avg” which takes a list of numbers and returns**

**[minimum, maximum, average] of the list**

**e.g.: def min\_max\_avg(in\_list):**

<PROGRAM CODE>

#6-1

def min\_max\_avg(in\_list):

minmaxavg=[min(in\_list),max(in\_list),sum(in\_list)/len(in\_list)]

return minmaxavg

<RESULT>



**2) randomly generate 10 numbers and, calculate the average, minimum, and**

**maximum values using above “min\_max\_avg” function**

**e.g.: mean\_min\_max([1,6,2,8,3,5,-4,2]) returns [-4, 8, 2.875]**

<PROGRAM CODE>

#6-2

random\_list=np.random.randint(-100,100,10)

print(random\_list)

print(min\_max\_avg(random\_list))

<RESULT>

스크린샷이(가) 표시된 사진

자동 생성된 설명

**3) define a function “equ\_interval” which divides a value range into n equal**

**intervals.**

**input: 1) list [min, max] of range, 2) number of intervals**

**output: list of (equal distance) intervals**

**e.g.: equ\_interval([-4, 8], 3) returns [[-4,0], [0,4], [4,8]]**

<PROGRAM CODE>

#6-3

def equ\_interval(minmax,interval\_num):

minr=minmax[0]

maxr=minmax[1]

interval\_size=int((maxr-minr)/interval\_num)

out\_equ=[[0]\*2 for i in range(interval\_num)]

out\_equ[0][0]=0

for x in range(int(interval\_num)):

out\_equ[x][0]=minr+interval\_size\*x

out\_equ[x][1]=minr+interval\_size\*(x+1)

return out\_equ

<RESULT>

스크린샷이(가) 표시된 사진

자동 생성된 설명

**4) run equ\_interval 2 times by using different values of list and number of**

**intervals.**

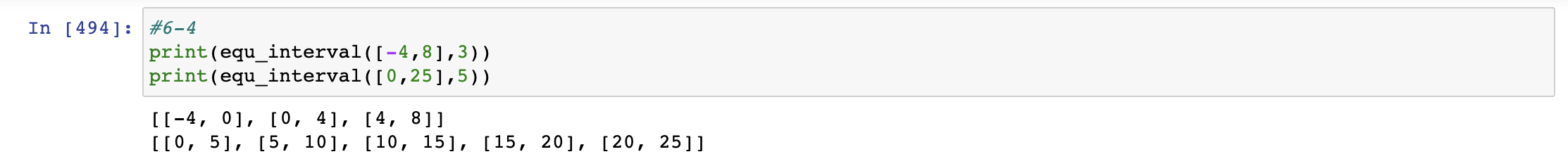
<PROGRAM CODE>

#6-4

print(equ\_interval([-4,8],3))

print(equ\_interval([0,25],5))

<RESULT>



**7. Write Python code for the following tasks.**

**1) define a function “no\_of\_values” which takes a list and returns the number of**

**values in the list.**

<PROGRAM CODE>

#7-1

def no\_of\_values(in\_list):

return len(in\_list) #1d

<RESULT>



**2) define a function “no\_of\_dis\_val” which takes a list and returns the number of**

**“distinct” values in the list.**

**e.g.: a\_list=[0,1,1,2,0]**

**no\_of\_dis\_val(a\_list) returns 3 ==> 3 unique values**

**This means a\_list contains 3 distinct values**

<PROGRAM CODE>

#7-2

from collections import Counter

def no\_of\_dis\_val(in\_list):

distinct\_list= (Counter(in\_list).keys())

return len(distinct\_list) #1d

<RESULT>

테이블이(가) 표시된 사진

자동 생성된 설명

**3) for every attribute in “a\_list”, calculate the number of values and distinct**

**values, respectively, using q 1) and q 2).**

<PROGRAM CODE>

#7-3

for x in range(1,len(a\_list[0])+1):

attributes\_val = a\_list[:,x-1:x]

into\_1d = attributes\_val.ravel()

# print(into\_1d)

print("attribute : ",x, ", val :", no\_of\_values(into\_1d),", dis :",no\_of\_dis\_val(into\_1d))

<RESULT>

스크린샷이(가) 표시된 사진

자동 생성된 설명

**4) plot a graphic table(e.g.: bar graph) by your favorite color using matplotlib as**

**follows: X axis: index of attribute, Y axix: number of distinct values.**

<PROGRAM CODE>

#7-4

import matplotlib.pyplot as plt

plot\_x=[]

plot\_y=[]

for x in range(1,len(a\_list[0])+1):

attributes\_val = a\_list[:,x-1:x]

into\_1d = attributes\_val.ravel()

# print(into\_1d)

plot\_x.append(x)

plot\_y.append(no\_of\_dis\_val(into\_1d))

plt.plot(plot\_x, plot\_y,'red')

plt.xlabel('index of attribute')

plt.ylabel('number of distinct values')

plt.show()

<RESULT>

스크린샷이(가) 표시된 사진

자동 생성된 설명