Machine Learning Homework 1 (Python Exercise)

Apr. 3, 2020

∗ Please note that <u>all homework should be your own work.</u> You should also not copy answers from other person's, books or internet resources.
∗ I didn't proofread the questions. If you find any typos/errors, let me know.

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

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]]

∗ 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]]

2) show the number of columns(attributes) and number of rows(records), respectively.
3) write a Python program that shows the first 5 rows from the "a_list".
4) write a Python program which randomly shuffles 'a_list' data

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.
2) show the reversed elements of q. 1) (We don't actually change the values a_list)

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)
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

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):
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]
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]]
4) run equ_interval 2 times by using different values of list and number of intervals.

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.
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
3) for every attribute in "a_list", calculate the number of values and distinct values, respectively, using q 1) and q 2).
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.

<u>Hand In</u>
1) In your report
 ...
 6. .Write Python code for the tasks...
 1) define a function "min_max" ...
 <PROGRAM CODE> <== This is title
 put your program code segment for Q 6 1) here
 <RESULT> <== This is title
 put the screen dump of your program run for Q 6 1) here

 2) randomly generate ...
 <PROGRAM CODE> <== This is title
 put your program code segment for Q 6 2) here
 <RESULT> <== This is title
 put the screen dump of your program run here
 ...

2) upload the following files at e-class.
 i) report file, ii) program source file

Due: 4/17(Fri) 11:59PM