National University of Computer and Emerging Sciences

School of Computing **Spring 2021** **Islamabad Campus**

CS218: Data Structures (Spring 2021)

Assignment 1

(Deadline: 31st March, 2021 11:59 PM)

Deadline: Deadline to submit assignment is **31st March, 2021 11:59 PM**. Correct and timely submission of assignment is responsibility of every student; hence no relaxation will be given to anyone.
Plagiarism: **-50% marks** in the assignment if any significant part of assignment is found plagiarized. A code is considered plagiarized if **more than 20%** code is not your own work.
Comments: Comment your code properly. **Bonus marks (maximum 10%)** will be awarded to well commented code.
Naming Convention: Submit .h file named '19ixxxx_DSAss_01.h"

Predict Your Data Structures Grade

You are given data for students' grade in first three semesters of one of the best universities in the world. Your task is to read the data in any suitable data structure and then predict your grade in Data Structures course during current semester. Note that you are NOT allowed to use ANY library/built-in data structures during Data Structures course. You are here to learn how to develop them!

This assignment would give you basic insight into prediction algorithms. Prediction algorithms usually read history of a given problem and predict future outcomes, based on the patterns occurring in the past. For the above problem to predict your Data Structures grade, you will be using a couple of very simple yet effective algorithms. In simple words, our prediction algorithms state that you would perform exactly like the student from past who has similar behavior and learning like you in first two semesters. It means that you have to look into history data and find a student whose performance was just like you in first two semesters. Finding such a student from past would require some treatment of data according to some given rules.

An interesting fact is that some (read VERY FEW) students perform exceptionally different from their past behavior. It means that the data might contain some outlier cases as well. If your algorithm somehow looks into data and (unfortunately) picks that outlier student then this would not be correct prediction of your grade. As it is stated that such students are rare, so we assume that you are also not the one! Our algorithm should predict what maximum number of students like you did in the past. To cater for the issue, we would use probability (you have finally figured it out why we offer Calculus, Statistics, Probability, and other irrelevant courses during Computer Science degree).

Another interesting fact is that some students (read MAJORITY) perform well in one type of courses and fail to perform equally well in others. Like one student might score "A" grades in both

ITC and CP, but could not score even B grades in Calculus-I and Basic Electronics. This situation intrigues us to find the courses that relate to Data Structures more than other courses. To do so, we would like to give some weightage to courses. These weights would be used to calculate probability for prediction. I would like to assign a higher weight to the courses that give me correct prediction about Data Structures grade and low for the others.

The following features carry marks independent of the overall working of the algorithm, so you are required to strictly follow the code structure provided to you. In case of failure to follow the structure, you might lose marks.

Step 1. Preprocessing of the dataset

Consider the following dataset:

Sr. No	Semester	Course Co	Course Tit	Credit Hou	Grade	Grade Poi	SGPA	CGPA	Warning
1	Fall 2016	MT104	Linear Alg	3	B	3	3.27	3.27	0
1	Fall 2016	MT101	Calculus -	3	B+	3.33	3.27	3.27	0
1	Fall 2016	CS101	Introducti	3	A	4	3.27	3.27	0
1	Fall 2016	CL101	Introducti	1	A+	4	3.27	3.27	0
1	Fall 2016	EE182	Basic Elec	3	C-	1.67	3.27	3.27	0
1	Fall 2016	SL101	English La	1	A-	3.67	3.27	3.27	0
1	Fall 2016	SS101	English La	3	A+	4	3.27	3.27	0
1	Fall 2017	CS201	Data Struc	3	A	4	3.75	3.57	0
1	Spring 2017	EE227	Digital Log	3	A+	4	3.71	3.49	0
1	Spring 2017	SS122	English Co	3	A	4	3.71	3.49	0
1	Spring 2017	MT115	Calculus -	3	A-	3.67	3.71	3.49	0
1	Spring 2017	SS111	Islamic an	3	B	3	3.71	3.49	0
1	Spring 2017	CS103	Computer	3	A	4	3.71	3.49	0
1	Spring 2017	EL227	Digital Log	1	B	3	3.71	3.49	0
1	Spring 2017	CL103	Computer	1	A+	4	3.71	3.49	0

Table 1 The sample record of the student having roll number (Sr. No) 1.

1.1. Instructions for Data Preprocessing/ Data Cleaning

You are required to read the “Students_Dataset.csv” file and restructure this dataset by arranging each student’s records in one line as shown below.

Sr. No	MT104	MT119	CS118	CL118	EE182	SL101	SS101	EE227	SS122	MT224	SS111	CS217	EL227	CL217	CGPA	Warning	CS201(Data Stratures)
1	3	3.33	4	4	1.67	3.67	4	4	4	3.67	3	4	3	4	3.49	0	4
2	4	4	3.67	4	3.33	3	4	4	4	4	3.33	4	2.33	4	3.66	0	4
3	3.33	2.67	2.67	3.33	2.33	3.33	3	3	4	3	3.67	3.67	3	4	3.21	0	3.33

Each student record (feature vector) consists of his scores of all subjects of the first two semesters, CGPA, and Warning, where we will treat CS218-Data Structures as a label or target class.

Algorithm 1: Implement the **MyBrothersInDS** algorithm which has the following implementation steps:

1) Load data into an appropriate data structure, and assume a variable k=10.

2) Take distance of your scores with the scores of each student. Distance is the absolute difference of values here i.e., for each course X, $\sum$ |your GPA in course CX – student i GPA in course CX| is your distance from student i. You are required to store this distance for all students i.e., i = 0 to N. When calculating distance, you are required to multiply the distance from CS courses with $\partial = 1.5$ to give more weightage to them.

3) Sort your distances from students and select k least distant students.

4) Assign yourself the grade that majority of students from these k selected students have scored in DS, e.g., if these k=10 students have scored B+, B+, B, A+, A, A, B, B, B-, and D in DS then your grade in DS would be B as 3 students have scored B.

Algorithm 2: Now here is the second algorithm, named **MyDSGroup**:

1) Load data into an appropriate data structure.

2) There are a total of thirteen (13) possible grades; A+, A, A-, B+, B, B-, C+, C, C-, D+, D, D-, and F. Set a variable equal to 13 (say k=13).

3) Initially pick 1st 13 students and assume that each belong to one group. There would be total 13 groups. Calculate score of these 13 students for the first two semesters as per following score:

- A/A+ = 13 points
- A - = 11 points
- B+=10 points
- B = 09 points
- B- = 08 points

… so on for remaining grades

Add up points for all courses to find total score. Call this score mean of the group. There would be 13 means, say M1, M2, M3, …, and M13. In next steps we would refine these groups based on the grades of all students in first two semesters.

4) Calculate score of each student based on above scoring method and assign him/her a group that has the closest mean to this score (initially calculated in step 2 and then refined as per algorithm). All students are now assigned to their closest group.

5) For every group of students in step 3, find the mean score of the group. Overwrite the previous mean with this new mean. Repeat step 3 and 4 with this new mean to update groups. Groups will be updated after every iteration.

Eventually, actual groups of students as per actual underlying distribution will be reached. In order to decide whether we have reached the actual distribution, we would calculate the

difference between two consecutive means of a group. If the difference for all groups is less than a small factor (say 0.001), algorithm would stop. This part is called **Model Training.**

6) Calculate your score of first two semesters and assign yourself the group that has closest mean to your score. Assign yourself the grade that majority of students in that group scored.

Your task is to implement both **MyBrothersInDS** and **MyDSGroup** and predict your grade.

Helping Note:

Link for understanding K-Mean clustering algorithm which is similar to your Algorithm (**MyBrothersInDS)** except you have to calculate distance according to above description:

https://www.youtube.com/watch?v=UqYde-LULfs

Link for understanding K-Mean clustering algorithm which is similar to your Algorithm (**MyDSGroup):**

https://www.youtube.com/watch?v=rESDfP2ik3k

Good Luck!