T.C.
DOKUZ EYLUL UNIVERSTY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

2023 – 2024
SPRING SEMESTER

CME 3202
CONCEPTS OF PROGRAMMING LANGUAGES

ASSIGNMENT 1:
DIABETES PROBABILITY CALCULATION SOFTWARE

DUE DATE:
23:55 – 21.05.2024

In this assignment you are asked to write a software that will take patient data from the user and checks if the inputted data is similar enough to diabetes patients given in diabetes.csv file.

You are required to use Python programming language and create a desktop application for this assignment. You are free to use any python library and framework for Graphical User Interface (GUI) but you are strongly advised to use something simple, popular and part of standard Python installation to make your development and execution easy and to make it work with standard Python installation in both Windows and Linux operating systems without the requirement of installing specific Python packages and libraries.

You are given a Comma Separated Values (CSV) file named “diabetes.csv”. This file contains 768 different records (rows) of data, divided into 9 different columns. The first 8 columns are related patient information (inputs of our program) and last column “Outcome” contains is patient is diabetic or not (where 1 means is True and 0 is False for diabetes diagnosis).

Your program should contain 8 text boxes for patient information, corresponding to first 8 columns of data. Your program should check before execution if all 8 text boxes contains a value and if one or some of them are not empty.

After all inputs are given, you should check if the given data in text boxes are between the smallest and largest data in diabetes.csv file. For example, for the first column (Pregnancies), the minimum value is 0 and maximum value is 17. If the values entered are not within the values given in dataset, you should give an error, warn the user and request input again.

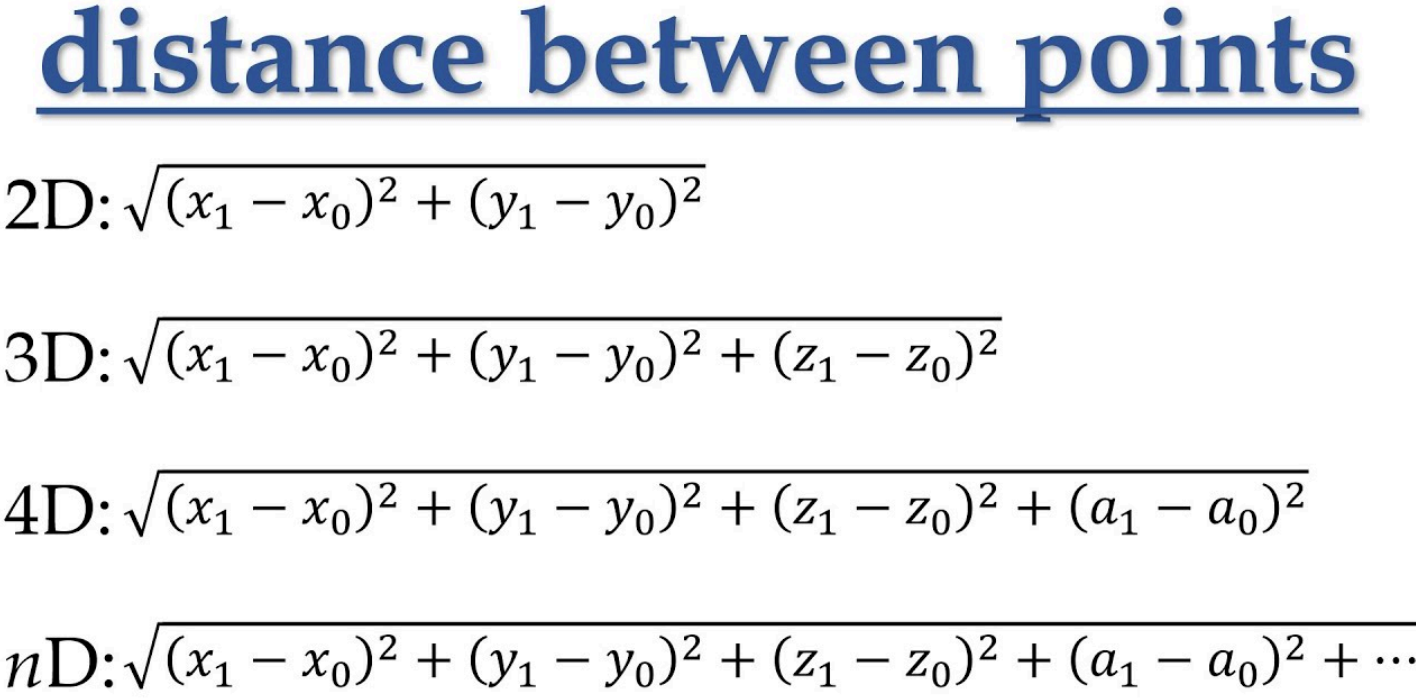
At the start of the execution, your program should read all values from given CSV file and perform data standardization (preprocessing) on all columns. This means mapping all integer or float values inside a column, between 0 and 1. For example, for the first column (Pregnancies), means changing 0 to 0 (because it is the minimum value) and 17 to 1 (because it is the maximum value). For values between minimum and maximum, you should use the following formula.

$$N_{new} = \frac{N_{old} - N_{minimum}}{N_{maximum} - N_{minimum}}$$

For example, again for the first column (Pregnancies), this means value 1 is changed to 0.058823529 and value 16 is changed to 0.941176471. This data transformation allow our data mining and machine learning algorithms to work correctly, on same given data range (between 0 and 1).

You should save this transformed dataset to “diabetes_preprocessed.csv” to check if your data manipulation was correct or not.

After you checked if the input data was correct or not and after you standardized (preprocessed) your data set, you should calculate the Euclidean Distance between your input data and all records in your dataset. How to do this operation is given in Figure 1 below. Since your data has 8 columns for input and 1 column for output, you should use a 8th dimensional Euclidean Distance calculation.



distance between points

$$2D: \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$$

$$3D: \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}$$

$$4D: \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2 + (a_1 - a_0)^2}$$

$$nD: \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2 + (a_1 - a_0)^2 + \cdots}$$

Figure 1: Euclidean Distance calculation from 2 dimensional to n dimensional space (image taken from https://www.youtube.com/watch?v=K6Eu0kRolmA).

After this operation is done, your program should find the closest 5 records to given input data using Euclidean Distance calculation and check how many of them are diabetic or not and give the user a probability of diabetes based on these results. For example, if we are looking for closest 5 patient records and 3 of them are diabetic, the probability of diabetes for your inputted patient is 60%. The number of closest records (in this case 5) is arbitrary and your program should work with different requested closest values (as an input on your GUI), between 1 and total number of records in your dataset (at an extreme case). This input should also be checked if it is in valid range or not.

This was the main part of your assignment. In part 2, you are asked to write a simple program in programming languages Go, Rust and Ruby, to calculate Euclidean Distance between given two points. You are not required to develop a GUI or getting an input from console for these applications, just showing us that you can write and execute Euclidean Distance formula with two example records from dataset. During your code control, you will be asked to try your program with alternative records from dataset, so make sure your variables are easily copied from dataset. You can use https://ideone.com/ or alternative website to develop and show your program during code control.

UPLOAD REQUIREMENTS:

You are required to work in groups of 2 or 3 for this assignment, not working individually or on more than 3 person groups. You will be asked of your group structure during class or as a text assignment upload in Sakai. Do not forget to form your groups in requested time period.

You are required to upload four different files for this assignment. One is the Python file (.py) with your assignment. The next three are your Euclidean Distance formula implementation in Go (.go), Rust (.rs) and Ruby (.rb) programming languages. Only one of the group members should make an upload, it is not required for all members to do a separate upload of same files.

Do not “zip” or “rar” requested files and upload them. It is not necessary and it makes it harder for us to evaluate your assignments. Please upload 3 files as they are without compressing them to a single file.

The naming of your files should follow the format below for 3 person groups. For 2 person groups just write 2 student numbers in ascending order. You should write your group numbers with one leading zero if it is between 1 and 9, normally for 10 and above.

FORMAT:

```
GROUP_<group_number>_<student_number_1>_<student_number_2>_<student_number_3>.py
GROUP_<group_number>_<student_number_1>_<student_number_2>_<student_number_3>.go
GROUP_<group_number>_<student_number_1>_<student_number_2>_<student_number_3>.rs
GROUP_<group_number>_<student_number_1>_<student_number_2>_<student_number_3>.rb
```

EXAMPLE:

```
GROUP_01_2023510123_2023510124_2023510125.py
GROUP_01_2023510123_2023510124_2023510125.go
GROUP_01_2023510123_2023510124_2023510125.rs
GROUP_01_2023510123_2023510124_2023510125.rb
```

You are required to come to your lab session (planned to be your last lab session in 23.05.2024) or previous day morning and show your code and execution. If you do not come to this code control, even if you made an upload, your assignment will be graded zero. The time table for code control will be done during following lab sessions.

You are expected to write your own code for algorithms instead of using an available method for calculations. If you use such as a method that makes this assignment trivial to code, your grade for this part of assignment will be zero.

Your uploaded source codes will be checked for cheating and plagiarism. If cheating is detected, your entire assignment will be graded zero. If you or other students copy your code from an online source rather than writing it yourself, it will be considered as cheating as well.

Make sure that you upload your correct assignment to correct upload. If you accidentally upload another assignment (from another class for example) or to an incorrect upload (other section’s upload), it will be considered as not turned in and it will be graded as zero. Worst of all, you will only realize it after grades are published and it will too late to fix it.

If you have any questions or problems regarding this assignment, you can ask about it in our lab sessions. If you wish, you can also ask it in class forums or assignment page comments. If you send an email and if your question is answered, please share this information with other students to prevent asking of the same question again and again.

Your assignment will be open for upload until 23:55, 23.05.2024. This is done to allow students who may experience extreme problems (no Internet or electricity, computer crash or failure, etc.) and miss the deadline as a result. This extension will allow them to upload. If you are still unable to upload, send us an email informing your situation with your upload files in attachment and at the same time, try everything you can to upload your assignment.

Lastly, please do not forget to click “Submit” button after you upload your assignment files. If you do not, even though your files are uploaded to Sakai, you are labeled as “No Submission” and ignored when we try to download your assignments, making your uploaded files invisible to us, leading us to assume you did not make an assignment submission.

GOOD LUCK TO YOU ALL!