Hacettepe University

Department Of Artificial Intelligence

BBM 103 Assignment 2 Report

Yusuf Emir Cömert – 2220765023

24.11.2022

Contents

Doctor's Aid

1 – Analysis:

In this assignment, It is requested us to code a python program of a hospital system which takes inputs from .txt file and gives outputs to another .txt file. There are some commands that we obliged to do. The commands are create, probability, recommendation, list, remove.

Firstly I created two empty lists named patient_information[] and names[]. It's because I"ll append the inputs later.

For example when user enters "create [name] [Diagnosis Accuracy], [Disease Name], [Disease Incidence], [Treatment Name], [Treatment Risk]" as an input, program will create a patient to the system and gives output us like "Patient xxx is recorded"

When user enters "Probability [name]", program will calculate the probability of the name and lowercase or uppercase doesn't matter. Program will work either way without an error. I will talk about calculating the probability in the probability function() section.

Here comes the findPatient() function which is the most important function for me because it solved all of my questions. It was not mandatory but it worked for my code very well so I thougt I should mention about it. It is basically does not have a job by itself but helps with most of my function. Shortly its usage is to finding the name, or the diseases.

Read() function is a short function and it's job is as you can imagine, reading the my input file.

Remove() function is for if you want to remove someone that you created before, you should just input remove xxx and the program will give you output like "Patient xxx is removed". It is basic function like read function.

Calculate() function is for calculating the probability. I didn't have to make another function for this. I could have done it in the probability function but with this way program would look more tidy and readable. I'll explain the logic of this function in the calculate function() section of this pdf.

Recommendation function is for an answer of the question "Should xxx take the treatment or not?". It has basic logic and calculation. The output would be "System suggests xxx to have the treatment" or "System suggests xxx NOT to have the treatment". If you want to get an output, you should just enter the input "recommendation [name]" inside of the input file.

Saving output function is for writing the result to the .txt file. It has a two line short codes because I called the save_output() function end of the every functions.

Lastly I have list() function and it's job is apparently creating a table and see the patient's names, their diagnosis accuracy, diseases names, diseases incidences, treatment names and treatments risks. To run this list function, you basically write "list" to the input file.

2 – Design:

I tried to make this program as simpler as I can. I have 85 line and it is very little number for this program. It's because I don't have much lines outside of the functions and it makes code more readable.

I think I have enough comment and if somebody who has low knowledge python like me can read the program easily. I also used meaningful naming.

I used a lot of "line.startswith" command because inputs would start with the command that I mention like create xxx …. You can see it started with a command.

```
if command.startswith("remove "):
    save_output(("Patient " + patient_information[find][0] + " is removed."))
```

You might see that "line.startswith" notation a few more times in my code.

Using the ".capitalize()" command works for when we want to print something to output file, it makes our string's first letter to uppercase because sentences should start with uppercases.

Used .format notation for printing the list and .format's job in Python(str. format()) is technique of the string category permits you to try and do variable substitutions and data formatting. It enables you to concatenate parts of a string at desired intervals through point data format

One of the most important phrase is ".split()" notation. The name of this phrase gives a hint because it's job is the split sentences to a words. For example our input has some informations separated by "," so we can use this code:

```
actualCommand = command.split()[0]
name = command.split()[1]
```

You can't see the full command right now so it might not make sense but for example in second line is for splitting the names from input. There are little bit more split examples in my code. I will explain them while explaining the functions. The big problem was there were no "," before create command so I used .split notation but different way. If you want to take a look at it you can go create() function page.

I have to explain this part of my code because it is not in any function and I will pass to the part of explaining the functions after this page of the pdf.

```
while True:
    line = input_file.readlines()
    for say in range(len(line)):
        if line[say].startswith("create "):
            create(line[say])
        elif line[say].startswith("list"):
            list()
        else:
            findPatient(line[say])
    if not line:
        break
```

It is basically for create and list commands. Firstly I defined “line” and made it equal to “readlines()” which is the command that I used for reading one line. You can see the “.startswith” notation again, it is really useful for this assignment. “If not” line is for preventing from an error when we enter empty input.

```
patient_information.pop(find)
```

I used “.pop()” notation in remove() function for removing information that we want. Python list pop() is an inbuilt function in Python that removes and returns the last value from the List or the given index value. I used this inbuilt function for removing, not returning the last value.

There we have global thing;

```
global input_file
```

I made input_file global because I defined input_file in read() funcion’s scope. When I tried to call it again, it gives us an error. I used global method to make it general of the code and it didn’t returns an error now.

```
output_file = open('doctors_aid_outputs.txt', 'w')
```

This line was not in a scope of a function and it’s job is writing. I didn’t define it in a function because I didn’t want it to be in a scope because I want to call it whenever I want. I could’ve just make it global but I didn’t want it to be I don’t know why I didn’t. I think I did it without thinking.

I used lower() function for making output string lowercase.

```
has a probability of " + str(actualRisk) + "% of having " + patient_information[find][2].lower() + "."))
```

The output would be look like this:

```
has a probability of 31.82% of having colon cancer.
```

If I didn't use that function, the output of this code would look like "…… of having Colon Canser". It wouldn't be much problem but why would I make mistake of writing rules.

3-Programmer's Catalog:

3.01 – Functions:

In this presentation of python program, I used nine(9) functions. Their names are ;

"read(), create(), save_output(), findPatient(), calculate(), probability(), recommendation(), remove(), list()" functions.

3.1 – Why Did I Used Functions?:

1. They allow us to conceive of our program as a bunch of sub-steps. (Each sub-step can be its function. When any program seems too hard, break the overall program into sub-steps!)

2. They allow us to reuse code instead of rewriting it.

3. Functions allow us to keep our variable namespace clean (local variables only "live" as long as the function does). In other words, function_1 can use a variable called I, and function_2 can also use a variable called I, and there is no confusion. Each variable I only exists when the computer is executing the given function.

4. Functions allow us to test small parts of our program in isolation from the rest. This aid is especially true in interpreted languages, such as Python and Matlab, but it can be helpful in C, Java, and ActionScript.

5.They make our code look tidier and cleaner and this makes code more readable.

6.We can call them whenever we want.

7.Functions break long programs up into smaller components .

8. Functions can be shared and used by other programmers

3.2-Read() Function:

```
def read():
    global input_file
    input_file = open("doctors_aid_inputs.txt", 'r')
```

Read function is pretty short. I made input_file global because I thought I will need to call it later but later, figured that I didn't needed to. I used open() function to open the doctors_aid_inputs.txt file and the part that comes after it, I mean the 'r' part; 'r' is comes from the 'r'ead.

3.3-Create() Function:

```
def create(a):
    newPatient = a[7:].split(", ")
    patient_information.append(newPatient)
    save_output(str("Patient " + newPatient[0] + " is recorded."))
```

Create function has a lot work and it was one of the hardest part for me. For example

"create xxx ….." we have input like this. "c,r,e,a,t,e, " has 7 digits(including the white space) that's why I splitted after the 7'th index. The job of the split function is splitting the text from where you say it and after splitting it makes a new list. My list is called newPatient which make sense by its name because this list keeps the information of patients inputs. I have actual list that I record the patients information which name is "patient_information". I used .append function to add this informations to actual list.

Lastly I have save_output function which is write function. I'll explain it later of the functions. Output will be "Patient xxx is recorded.". I used newPatient[0] for the name because I explained that after I split from the create and white space, others would be the informations which the first index is the name. That's why I said 0'th index.

3.4-Save_Output() Function:

```
def save_output(text):
    output_file.writelines((text))
    output_file.writelines(('\n'))
```

Save output function is the second smallest function of my program (After read function).

I defined the output file actually which is;

```
output_file = open('doctors_aid_outputs.txt', 'w')
```

I didn't call it in the function because I wanted it to be global. If I wanted to do it I could've just make it global it wouldn't be a problem. I could've used ".write" command not ".writelines" but writelines function writes the output line by line that's why it would be more efficient for me so I used writelines. Then here comes '\n' . It says skip on to next line command. I tried to do it without using backslash n but outputs were getting together so I used it to split them.

Of course I used open function again and I write 'w' because it comes from 'w'rite command.

3.5-FindPatient() Function:

```
def findPatient(command):                                                    #This is the main f
    actualCommand = command.split()[0]                                       #I was going to nam
    name = command.split()[1]                                                #Index 0 is create,
    for find in range(len(patient_information)):
        if name in patient_information[find]:
            probability(command, find)
            recommendation(command, find)
            remove(command, find)
            break
        else:
            pass

    else:
        save_output((actualCommand.capitalize(), " for ", name, " cannot be calculated due to absence."))
```

This function is probably the most complicated function of me and it is the hardest to explain. I couldn't find a new name so I called actualCommand and this is when we split the input takes first string as a command like probability, recommendation, remove etc.

Name = .split[1] because 0'th index is saying that what the program do like create. The first index is gives us a name of this patient. I defined the find function in this for loop. In the for loop, we have an if statement. I said that patient_information list is the list of every patients information.

In this if statement if this name is in this main list, it calls the functions that will calculate the probability, recommendation, also the remove function; then I used break to get out of this loop. This if has no else only the pass command to not have a problem but for loop's else gives us an output. İf input be like "probability Hayriye" , the output would look like for example:

"Probability for Hayriye cannot be calculated due to absence" this output occurs when we do not define the name or if we deleted(removed) the name program will delete it from its memory and gives output like this.

3.6-Calculate() Function:

```
def calculate(find):
    global risk
    global actualRisk
    incidence, total = (patient_information[find][3]).split("/")
    incidence = float(incidence)
    total = float(total)
    accuracy = float(patient_information[find][1])
    predicted = (total - incidence) * (1 - accuracy) + incidence
    risk = incidence / predicted
    actualRisk = round(100 * risk, 2)
```

We have calculate function. This function was not obligatory but I wanted to use it because it was really useful for me. I used this function to find the probability calculation and firstly I made the global risk and actualRisk . I'll come to explain the difference between risk and actualRisk. I splitted from third index because 3'rd index is a number like 50/100000 which makes 50 people has this disease from 100000 people to find nominators and denominators. I turned the incidence and total to float. Accuracy is a number like 0.999 which gives us an information of accuracy of this disease because humans can make mistakes and this disease diagnosis might be wrong. That's the problem of calculating the probability function. I watched a video to understand how to calculate probability function and as to that video I understand how to calculate it. It shows how to calculate probability it is hard to explain. Risk is incidence/predicted and the actuaRisk is hundred times risk and rounding it to 2 digit after the comma(,). For example if the probability calculates number like 95.6734513242414….. the output shows it like 95.68 and this looks more clean

3.7-Probability() Function:

```
def probability(command, find):
    if command.startswith("probability "):
        calculate(find)
        save_output(("Patient " + patient_information[find][0] + " has a probability of " + str(actualRisk) + "% of having " + patient_information[find][2].lower() + "."))
    else:
        pass
```

This probability is short for vertical size but it is the longest second command of the program for horizontal axis. It basically comes if out input starts with probability like code says. There comes the beautiful part of the code which I called the calculate function and it calculates it. That's why the probability function is short but the calculate function is pretty long . Then I called save_output function as you can see which calls the write function named save_output .As you can see end of the code has .lower() function and it makes output in lower case words. If I didn't used the lower() function the output would be like "Patient Hayriye has a probability of 33.34% of having Breast Cancer" because we entered input as "Breast Cancer". When I used lower function the output turns into "breast cancer" with lower letters.

3.8-Recommendation() Function:

```
def recommendation(command, find):
    if command.startswith("recommendation "):
        calculate(find)
        if risk < float(patient_information[find][-1]):                    #If risk < probability,
            save_output(("System suggests " + patient_information[find][0] + " NOT to have the treatment."))
        else:
            save_output(("System suggests " + patient_information[find][0] + " to have the treatment."))
```

Recommendation function used for finding the answer of "if the system suggest to take the risk or not?" question. As you can see I started with .startswith notation to find recommendation command. Here comes again the beautiful part which is calculate function again. Careful I used risk, not actualRisk. actualRisk gives us probability of rounded risk. Program basically says if risk < treatment risk system does not suggests this patient to have the treatment because ratio of the failing treatment is too high. Otherwise system suggests to have the treatment. I used [-1] -1'st index because it gives us the last index which is the treatment risk. For example input of the Hayriye is "create Hayriye, 0.999, Breast Cancer, 50/100000, Surgery, 0.40" in this case 0.40 is going to be our treatment risk .

3.9—Remove() Function:

```
def remove(command, find):
    if command.startswith("remove "):
        save_output(("Patient " + patient_information[find][0] + " is removed."))
        patient_information.pop(find)
```

As you can see we have again startswith command and I have to explain it again if the input starts with remove, we will come to this function. Program basically removes the person we want with .pop() command inside of the brackets we have find. Pop command is for extract the person we want from our list. Find basically finds the person we ask for and again I called the save_output function to write it to txt file. The output would look like “Patient xxxx is removed.”

3.10-List() Function:

```
def list():                 #Creating the table
    save_output(("\n{:<15}{:<15}{:<20}{:<15}{:<20}{:<5}".format("Patient", "Diagnosis", "Disease", "Disease", "Treatment", "Treatment")))
    save_output(("{:<15}{:<15}{:<20}{:<15}{:<20}{:<5}".format("Name", "Accuracy", "Name", "Incidence", "Name", "Risk")))
    save_output(("{:<94}".format(94*"-",)))

    for info in range(len(patient_information)):
        save_output(("{:<15}{:<15}{:<20}{:<15}{:<20}{:<5}".format(patient_information[info][0], patient_information[info][1],patient_information[info][2],
```

The continuation of the line of save_output is like:

```
, patient_information[info][3], patient_information[info][4], patient_information[info][5])))
```

This list function has longest code and even it didn’t fit to my screen to take screenshot. The list function is actually all about write function and as you can see I only used save_output function. In the first line I started with “backslash n” because in the output, list and the other sentences were getting into each other so I wanted to split them with this way. I used .format function because this method attaches the thing we asked inside to the curly braces “{]}”. The smaller than symbol(<) means align to the left and the numbers are the space numbers. In the first line we have “Patient, Diagnosis, Disease, Treatment, Treatment”. In the second line we have something like …. “Name, Accuracy, Name, Incidence, Name, Risk”. In the first look you might not see why did I write this but if you read them vertically you can see the “Patient Name, Diagnosis Accuracy etc.” which make senses. In the third line you might not see why did I used 94* “-” . Actually it is for the line that splits the titles and contents. You can see it in the output of the list.

In the longest part of the code is attaching the files to their places 0’th index to first place… The output of the list function is this:

```
Patient Pakiz is recorded.

Patient        Diagnosis      Disease            Disease       Treatment          Treatment
Name           Accuracy       Name               Incidence     Name               Risk
-------------------------------------------------------------------------------------------
Hayriye        0.999          Breast Cancer      50/100000     Surgery            0.40

Deniz          0.9999         Lung Cancer        40/100000     Radiotherapy       0.50

Ates           0.99           Thyroid Cancer     16/100000     Chemotherapy       0.02

Toprak         0.98           Prostate Cancer    21/100000     Hormonotherapy     0.20

Hypatia        0.9975         Stomach Cancer     15/100000     Immunotherapy      0.04

Pakiz          0.9997         Colon Cancer       14/100000     Targeted Therapy   0.30

Patient Ates is removed.
```

You can see the “------“ line that splits titles from contents and you can also see the benefit of using “\n” . If I wouldn’t use “\n” stuff the output would look like:

```
 9  Patient Pakiz is recorded.
10  Patient        Diagnosis      Disease            Disease       Treatment          Treatment
11  Name           Accuracy       Name               Incidence     Name               Risk
12  -------------------------------------------------------------------------------------------
13  Hayriye        0.999          Breast Cancer      50/100000     Surgery            0.40
14
15  Deniz          0.9999         Lung Cancer        40/100000     Radiotherapy       0.50
```

You see that there is no spaces between line-9 and line-10 and it looks ugly, complicated.

‘’’Actually in the Assingment2 pdf file says that we should use tabs to make the spaces but I made it with spaces like you can see in the eleventh page I used .format method because when I coded this program there was no information about using spaces or tabs method. I didn’t want to broke my code after finished it so I leave it be. ‘’’

4-While Loop:

```
while True:
    line = input_file.readlines()
    for say in range(len(line)):
        if line[say].startswith("create "):
            create(line[say])
        elif line[say].startswith("list"):
            list()
        else:
            findPatient(line[say])
    if not line:
        break
```

This while loop is not in any function so I wanted to mention about it too but not in functions section. To turn this while loop on I started with while true: .readlines() reads the inputs line by line. As you can see I called input_file here that was the reason of making it global. This while loop does not have much of an work actually .You can see the .startswith function again here.

5-User's Catalog:

1-When user runs the program, there would be no output for console except this:

```
"C:\Users\Yusuf Emir Cömert\AppData\Local\Programs\Python\

Process finished with exit code 0
```

Also no response from terminal if you would start it from there.

2-Be careful about input and output file the input file's name should be = doctors_aid_inputs.txt

The output file's name should be = doctors_aid_outputs.txt

If you don’t want to use with this names you should change it from getting into code and change it manually.

3- The example of inputs must not be anything except type of this:

```
create Hayriye, 0.999, Breast Cancer, 50/100000, Surgery, 0.40
```

```
probability Hayriye
recommendation Ateş
```

```
list
remove Ateş
```

You can change values and names but you can not change the format of inputs or the uppercase or lowercase of the commands if you try it you can get errors.

6-Grading:

EVALUATION	Points	Grading
Indented and Readable Codes	5	5
Using Meaningful Naming	5	5
Using Explanatory Comments	5	5
Efficiency(avoiding unnecessary actions)	5	5
Function Usage	25	25
Correctness	35	35
Report	20	20
There are several negative evaluations	…	…