Hacettepe Univercity

Department Of Computer Engineering

BBM 103 Assignment 2 Report

Rıza ÇAKIR – 2220356059

24.11.2022

CONTENTS

ANALYSIS

In the problem, it was requested to establish a hospital system.

First command, patient name and if needed more patient info will be written to the imput file by the user. And the file will be read by the system.

With create command a list will be created for each patient in the system that includes patient name, diagnosis accuracy, disease name, disease incidence, treatment name and treatment risk.

After creating lists for patients other commands will be available to use by the user. Other commands are: remove, probability, recommendation and list.

With the remove command patient's info and list fort he patient will be removed from the system.

With the probability command probability of the patient actually has the disease will be calculated using Bayes' theorem.

With the recommendation command system will recommand or will not recommand to have the treatment depending on comparison of probability of the patient actually has the disease and risk of treatment.

With the list command information of registered patients will be listed and printed out to the output file.

DESIGN

First I have created needen variables: text = [] is for assining every line as an element of a list after reading input file ,patient_data_list = [] as the main list for patients containin every patients information , outputs = " " is for writing every situations outputs to the output file using writing() function, patients = [] is for containing every patients name for checkinsg for some situations , prob = 0 is for the probability of the patient actually has the disease and probPatient = [] is for getting information about patient for calculating probability of the patient actually has the disease.

For getting input from input file I have created reading() function.With reading() function I have opened the input file for reading and I have splited every line for assigning to the varibale named “text” as a list.

Next I have opened output file for writing and created writing() function. With writing() function I am able to write outputs for every situation.

In the main loop I have seperated first word in every line and made every line a list containing 2 elements. First element being the command and the other(s) being infomation about patients.

After that I have checked first word seperted from the line for every line using if elif clause. With that system was able to find which command to operate.

If first word of the line is create, system removes create from the list. And seperates second element from “, ”. And runs the create() function.

If first word of the line is remove, system first check if second word is in the patient[] list to find out that patient has been created. And runs remove() function. Else systems writes out negative message using writing() function.

If first word of the line is probability, system first check if second word is in the patient[] list to find out that patient has been created. And runs probability() function. Else systems writes out negative message using writing() function.

If first word of the line is recommendation, system first check if second word is in the patient[] list to find out that patient has been created. And runs recommendation() function. Else systems writes out negative message using writing() function.

If first word of the line is list, system will run listing() function.

PROGRAMMER CATALOG

Functions:

1) Reading function:

```
def reading():
    inputFile = open("doctors_aid_inputs.txt", "r")
    global text
    text = inputFile.read().splitlines()
```

I have opened “doctors_aid_inputs.txt” file from same directory with .py file for reading with open command and assigned to the inputFile variable as TextIO. I have used global for text to use out of function. “.read()” function reads file and assigns to “text” and “.splitlines()” function separates every line and turns them to elements of a list without hidden “\n”.

2) Writing function:

```
def writing():
    outputFile.write(outputs)
```

I have already opened outputfile before. With “.write()” command system writes variable named “outputs” to the outputfile. Variable “outputs” is defined in other commands.

3) Create function:

```
def create():
    patient_data_list.append(patient)
    patients.append(patient[0])
    global outputs
    outputs = "Patient {0} is recorded.\n".format(patient[0])
    writing()
```

Before calling create function I have removed “create” from the line and seperated every word with “, ”. This way every info about patients became an element for a list named patient. And then I have add patient[] list ino the patient_data_list[] list as an element. Next line I have add first element of patient[] list (this element is the name of the patient) to tane patients[] list as an element. And then assigned “outputs” variable “Patient {0} is recorded” with format making {0} patients name. And run writing() function to write out the output of the create function.

4) Remove function:

```
def remove():
    global outputs
    outputs = "Patient {0} is removed.\n".format(patient)
    patient_data_list.pop(ind)
    patients.pop(ind)
    writing()
```

Before running remove() function I have searched for second item in the line's list's (patient's name) index number in the patients[] (list contains patient's names) and run the function()And then assigned "outputs" variable "Patient {0} is removed" with format making {0} patient's name. After I have popped patient's list from patient_data_list[] and patient's name from patients[] lists using it's index number. And runned writing() function.

5) Probability function:

```
def probability():
    ind = patients.index(lines[1])
    global probPatient
    probPatient = patient_data_list[ind]
    accuracy = float(probPatient[1])
    rate = probPatient[3].split("/")
    sick = float(rate[0])
    healthy = float(rate[1])
    global prob
    prob = 100*((sick*accuracy)/((sick*accuracy)+healthy*(1-accuracy)))
    prob = round(prob , 2)
    global outputs
    outputs = "Patient {0} has a probability of %{1} having {2}.\n".format(probPatient[0] , str(prob) , probPatient[2])
```

First I have searched for second item in the line's list's (patient's name) index number in the patients[] (list contains patient's names). And assigned patient's list in the patients_data_list[] to the variable "probPatient". And turned diagnosis accuracy of patient to float and assigned it to "accuracy" variable. After I have splitted disease's incidence from "/", and assign first element to "sick" variable , second element to "healthy" variable and turned them to float. And then I have calculated probability of the patient actually has the disease using Bayes' theorem. And assigned it to "prob" variable. And I have rounded prob to two decimal places. And then assigned "outputs" variable "Patient {0} has a probability of %{1} having {2}.\n" with format making {0} patient's name ,{1 } probability of the patient actually has the disease and {2}disease'name.

6) Recommendation function:

```
def recommendation():
    probability()
    risk = (float(probPatient[5]))*100
    global outputs
    if prob > risk:
        outputs = "System suggest {} to have the treatment.\n".format(probPatient[0])
    else:
        outputs = "System suggest NOT {} to have the treatment.\n".format(probPatient[0])
```

First I have calculated probability of the patient actually has the disease. And got 6th element from probPatient as risk of treatment turned it to a float , multiplied by 100 and assigned it to variable “risk”. Then compared “prob” and “risk”. If prob is greater then risk I have assigned “outputs” variable “System suggest {} to have the treatment.\n” with format making {} patient’s name. Else I have assigned “outputs” variable “System suggest NOT {} to have the treatment.\n” with format making {} patient’s name.

7) Listing function:

```
def listing():
    global outputs
    outputs = f"{'Patient':<16} {'Diagnosis':<16} {'Disease':<16} {'Diseases':<16} {'Treatment':<16} {'Treatment':<16}\n"
    writing()
    outputs = f"{'Name':<16} {'Accuracy':<16} {'Name':<16} {'Incidence':<16} {'Name':<16} {'Risk':<16}\n"
    writing()
    outputs = "-"*95+"\n"
    writing()
    for patient in patient_data_list:
        outputs = f"{patient[0]:<16} {str(100*float(patient[1]))+'%':<16} {patient[2]:<16} {patient[3]:<16} {patient[4]:<16} {str(int(100*(float(patient[5]))))+'%':<16}\n"
        writing()
```

First I have assigned “outputs” variable f"{'Patient':<16} {'Diagnosis':<16} {'Disease':<16} {'Diseases':<16} {'Treatment':<16} {'Treatment':<16}\n” to set the first row of the table. I have use align and gave every column 16 spaces. And runned writing() function. And I have assigned “outputs” variable f" {'Name':<16} {'Accuracy':<16} {'Name':<16} {'Incidence':<16} {'Name':<16} {'Risk':<16}\n” to set the second row of the table. I have use align and gave every column 16 spaces. And runned writing() function. And I have assigned “outputs” variable “-”*95+“\n” for the line between categories and patients. And I have used for cycle to get every patients list from patient_data_list. And then I have assigned “outputs” variable f"{patient[0]:<16} {str(100*float(patient[1]))+'%':<16} {patient[2]:<16} {patient[3]:<16} {patient[4]:<16} {str(int(100*(float(patient[5]))))+'%':<16}\n" fort he patients info. I have use align and gave every column 16 spaces. And runned writing() function.

USER CATALOG

User Manual/Tutorial:

First open doctor_aid_inputs.txt file. And write your commands. If u use create command type create give a space and write patient's name, diagnosis accuracy, disease name, disease incidence, treatment name, treatment risk. After every info about patient put a "," and give a space and write the other info until last one. Example line for create command: create Hayriye, 0.999, Breast Cancer, 50/100000, Surgery, 0.40

If u use remove, probability or recommendatiton command write your command give a space and write patient's names. Examples: Remove Hayriye , probability Hayriye , recommendation Hayriye

If u use list command just write list.

Restrictions:

Anything written other than given format will cause error or gives wrong output. Switching patient's information's palces for create command etc.

Project .py file input file and output file have to be in the same directory. Unless code doesn't work .

Any of the info about patient being longer than 16 characters causes error.

Evaluation	Points	Guess Grading
Indented and Readable Codes	5	5
Using Meaningful Naming	5	5
Using Explanatory Comments	5	5
Efficiency (avoiding unnecessary actions)	5	4
Function Usage	25	23
Correctness	35	35
Report	20	15