Hacettepe University

BBM103

Assignment 2

Deniz Kutay Açıcı-2220356022

24.11.2022

1.Analysis

For this assignment we have to make a probability-based decision system that makes a list from patient's data(Patient name, diagnosis accuracy, disease name, disease incidence, treatment name, treatment risk).In this program we can list the details about the patients,calculate the patient's probability of having disease with Bayes's Theorem.System can make a comparison between the result of Bayes's Theorem and treatment risk and suggest to the user that patient should have the treatment or not.

System reads every line one by one and define the what function is written at the begin of the line.

2.Design

First i create a function that open "doctors_aid_input.txt" file and after that i made a code that can read this file line by line and splits the line as function and user's input.For example in "create Hayriye, 0.999, Breast Cancer, 50/100000, Surgery, 0.40" line the code splits the line as "create" and " Hayriye, 0.999, Breast Cancer, 50/100000, Surgery, 0.40 ".

As a function only "create, remove, list, probability, recommendation" strings defined.

Create function takes the user's input and it splits it and adding to the patient data list. And it takes name data and adds to the names list for the other functions.

Remove function is used as "remove name". If name exists in names list function takes that name's index in list of names and removes name from names list and the same index of patients data list.If name doesn't exist on names list it gives output as "name doesn't exist."

Probability function, rearrange the patients data a bit and then it returns the value of the Bayes's Theorem.

Recommend function, compare the value of Bayes's Theorem and treatment risk which we know from the input.If Bayes's Theorem's value is bigger than treatment risk recommend function recommends that patients should have the treatment or not.

List function lists all the patients data's properly.

Programmer's Catalogue

```
output = open("doctors_aid_outputs.txt", "w")

def input():        #input function
    return open("doctors_aid_inputs.txt", "r")
```

Input Function: it opens the “doctors_aid_inputs.txt” file

```
output = open("doctors_aid_outputs.txt", "w")
def write(file, text):
    file.write(text)   #output function
```

Output Function: It prints the text you want to “doctors_aid_outputs.txt” file. For using this function you need to write “write(file.name(which is i defined as output), “string you want to write”)”

```
lines = input().readlines()
for row in lines:
    space = row.find(' ')

    function = (row[:space])                #finds what the fuction is
    line = (row[space + 1:])                #strip the function
    line = line.replace('\n', '')           #strip the \n at the end
    name = line.split(",")[0]               #finds the name

    '''
```

This code is reading the lines of the input file which is “doctors_aid_list.txt”. for row in lines command reads all the characters in the line. row.find(‘ ‘) function finds the index of the first space in the line which i defined it as space.Then i defined 2 more variables which is function and line.Function variable starts from beginning of the line and end of the first space.Line variable contains rest of the line.

For example in “create Ateş, 0.99, Thyroid Cancer, 16/100000, Chemotherapy, 0.02” linet he function variable will be “create” and line variable will be “create Ateş, 0.99, Thyroid Cancer, 16/100000, Chemotherapy, 0.02 “.I added line = line.replace(‘\n’, ‘’) command because in text file you have to write commands line by line and when you press enter at it adds “\n” at the end of the line. Variable name is needed for create, remove, recommendation and probability functions.

```
if space == -1:
    if line == "list":
        list_patients()
else:
    pass

if function == "create":
    create_patient(line)

if function == "remove":
    remove_patients(line)

if function == "probability":
    try:
        output.write("Patient {} has a probability of {}% having {}.\n".format(name, (float(probability(name)) * 100), patient_data_list[names.index(name)][2]))
    except:
        pass

if function == "recommendation":
    recommendation(line)
```

As I mentioned before this assignment has 5 functions (create, remove, list, probability, recommendation).This part defines what the function is.Function variable was defined before the upper part.

When you call list function you write just “list” to the input text file so it doesn’t have a space.Because of that space value would be equal to -1.So if space value is equal to -1 it must be list function or an invalid function.

```
patient_data_list = []
names = []


def create_patient(line):
    if name in names:                                           #if name exists
        write(output, "{} is already recorded\n".format(name))
    else:                                                       #if name doesn't exit
        write(output, "Patient {} is recorded\n".format(name))
        details = line.split(", ")
        patient_data_list.append(details)
        names.append(name)
```

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

Create Function: At first this function checks that name exists or not.If name exists before it prints name is already recorded.If name doesn’t exist before it splits the commas and make every data index of a details list.Then it appends details list to another list patient_data_list which is systems main list.Also it appends name in names.

```
def remove_patients(line):
    if line in names:       #when name exists
        a = (names.index(line))
        patient_data_list.pop(a)
        names.pop(a)
        write(output, "{} is removed.\n".format(line))

    else:                   #when name doesn't exist
        write(output, "Patient {} cannot be removed due to absence.\n".format(line))
```

```
remove Ateş
```

Remove Function: This function firstly checks name exists or not. If name doesn't exist it prints "Patient 'name' cannot be removed due to absence . If name exists it takes name index in names list. Because code adds datas patient_data_list and names at the same time name index and details index must be the same.So this part pops the index of name on names list on patient_data_list.Also it pops name in names so the indexes name and details remains same.

```
from copy import deepcopy
```

```
def list_patients():

    write(output, "Patient\tDiagnosis\tDisease\t\t\tDisease\t\tTreatment\t\tTreatment\nName\tAccuracy\tName\t\t\tIncidence\tName\t\t\tRisk")
    write(output, "\n-------------------------------------------------------------------------")
    patient_data_list_copy = deepcopy(patient_data_list)    #for list function

    for list_row in patient_data_list_copy:
        while not ("%" in list_row[1]):
            """"""
            add "%", mul with 100, arrange the decimals of diognosis accuracy and treatment risk
            """"""
            list_row[1] = (float(list_row[1]) * 100)
            list_row[1] = ("%.2f" %float(list_row[1]))
            list_row[1] = str(list_row[1]) + str("%")

            list_row[5] = str(float(list_row[5]) * 100)
            list_row[5] = ("%.0f" %float(list_row[5]))
            list_row[5] = str(list_row[5]) + str("%")
```

```
Patient Diagnosis   Disease          Disease     Treatment        Treatment
Name    Accuracy    Name             Incidence   Name             Risk
-------------------------------------------------------------------------
Hayriye 99.90%      Breast Cancer    50/100000   Surgery          40%
Deniz   99.99%      Lung Cancer      40/100000   Radiotherapy     50%
Ateş    99.00%      Thyroid Cancer   16/100000   Chemotherapy     2%
Toprak  98.00%      Prostate Cancer  21/100000   Hormonotherapy   20%
Hypatia 99.75%      Stomach Cancer   15/100000   Immunotherapy    4%
Pakiz   99.97%      Colon Cancer     14/100000   Targeted Therapy30%
```

List Function: This function prints patients_data_list as a list.Firstly it prints the top part which shows datas feature(patient name, diagnosis accuracy, disease name, disease incidence, treatment name, treatment risk). Then it gets copy of patients_data_list.Because the inputs are like "Toprak, 0.98, Prostate Cancer, 21/100000, Hormonotherapy, 0.20" and in the assignment diagnosis accuracy and treatment risk should be listed as "%98.00" and

“%2”. So it changes 0.98 to %98.00 and 0.20 to “%2”.We don’t want to change the original patient_data_list so i create a copy of patient_data_list called patient_data_list_copy.While loop changes the diagnosis accuracy and treatment risk.There is `while not ("%" in list_row[1]):` code is necessary because we do not want to apply same changings to the same values.For diagnosis accuracy it multiples the value with 100. Then it turns the value with 2 decimal points.At the end it adds % to end of the value.For treatment risk it multiples the value with 100 and then it turns the value with 0 decimal points.At the end it adds % to end of the value.

```
    write(output, "\n")

    """
    manage how tabs will be used
    """
    # name
    if len(list_row[0]) <= 3:
        write(output, list_row[0])
        write(output, "\t\t")
    else:
        write(output, list_row[0])
        write(output, "\t")

    # diagnosis accuracy
    if len(list_row[1]) == 6:
        write(output, list_row[1])
        write(output, "\t\t")
    else:
        write(output, list_row[1])
        write(output, "\t\t")
    #disease name
    if len(list_row[2]) <= 7:
        write(output, list_row[2])
        write(output, "\t\t\t")
    elif len(list_row[2]) <= 11:
        write(output, list_row[2])
        write(output, "\t\t")
    else:
        write(output, list_row[2])
        write(output, "\t")

    #disease incidence
    if len(list_row[3]) == 9:
        write(output, list_row[3])
        write(output, "\t")

    #treatment name
    if len(list_row[4]) <=7:
        write(output, list_row[4])
        write(output, "\t\t\t")
    elif len(list_row[4]) < 16:
        write(output, list_row[4])
        write(output, "\t")
    else:
        write(output, list_row[4])

    #treatment risk
    write(output, list_row[5])

write(output, "\n")

pass
```

at the left side it manages how tabs will be used according to example output and at the end it goes to the new line with “\n”

Probability Function:

```
def probability(name):
    if line in names:
        split = patient_data_list[names.index(name)][3].split("/")                              #it splits count and total
        accuracy, count, total = float(patient_data_list[names.index(name)][1]), int(split[0]), int(split[1])
        return "{:.4f}".format((accuracy*count) / (accuracy*count + (total-count)*(1-accuracy)))   #Bayes theorem with 4 decimal
    else:
        write(output, "{} for {} cannot be calculated due to absence.\n".format(function, name))   #when name doesn't exist
        pass
```

```
if function == "probability":
    try:
        output.write("Patient {} has a probability of {}% having {}.\n".format(name, (float(probability(name)) * 100), patient_data_list[names.index(name)][2]))
    except:
        pass
```

Probability function a little different than others because when name exists it doesn't print anything.It just turns the value of Bayes's Theorem.If name doesn't exist it returns None but prints "probability for name cannot be calculated due to absence".At first probability function tries to write:

```
output.write("Patient {} has a probability of {}% having {}.\n".format(name, (float(probability(name)) * 100), patient_data_list[names.index(name)][2]))
```

If the name exists probabilty function will return as the value of Bayes's Theorem. So everything will be fine and that will give no error.But if name doesn't exist probability function return as None but will print "probability for name cannot be calculated due to absence." When probability returns None float(probability(name) will be an error because None can't be float.That is why i added try/except.

Recommendation Function:

```
def recommendation(line):
    if probability(name) != None:       #added this "if" because probability doesn't return when name doesn't exist
        if float(probability(name)) < float(patient_data_list[names.index(name)][5]):
            write(output, "System suggests {} NOT to have the treatment\n".format(name))

        else:
            write(output, "System suggests {} to have the treatment\n".format(name))
    else:
        pass
```

First it checks probability(name) is None or not.As i mentoned before probability function returns None when name doesn't exist.But it prints:

```
write(output, "{} for {} cannot be calculated due to absence.\n".format(function, name))   #when name doesn't exist
```

In recommendation function, function variable would be recommendation so it will print "recommendation for Su cannot be calculated due to absence.".

When name exists it basically makes compare between probability and treatment risk.If treatment risk is bigger than probability it prints patient should have the treatment or not.

User's Catalogue

1. Open "doctors_aid_inputs.txt" file
2. Write down your inputs:
   Inputs should be as:
   For create function, input must be:
   "create name, diagnosis accuracy, disease name, disease incidence, treatment name, treatment risk"

   NOTES: diagnosis accuracy must be in 0,xx format
   Disease incidence must be in x/100000 format
   Treatment risk must be in 0,xx format.

   For remove function, input must be:
   "remove name"
   For list function, input must be :
   "list"
   For probability function, input must be:
   "probability name",
   For recommendation function, input must be:
   "recommendation name"
3. Save all the inputs to the text
4. Run "Assignment2.py"file
5. Outputs will be on "doctors_aid_outputs.txt" file