School of Data and Computer Science, Sun Yat-sen University

E09 Bayesian Network

17341111 Xuehai Liu

November 7, 2019

Contents

1 Pomegranate Installation

Under Linux:

1. Install `python` first (**python 2**, not python 3).
2. Run `sudo apt-get install python-pip` to install `pip`.
3. Run `sudo pip install pomegranate` to install `pomegranate`.

Under Windows

You can also run `pip install pomegranate` if you have installed `pip`. If you don't know how to install `pip`, please click `https://jingyan.baidu.com/article/e73e26c0d94e0524adb6a7ff.html`.

For more, please click the homepage of Pomegranate - `https://github.com/jmschrei/pomegranate` for help.

2 Building Bayesian Network

Please refer to `Tutorial_4_Bayesian_Networks.pdf`. I will explain it in class.

3 Tasks

3.1 Burglary

Please code to calculate:

1. $P(A)$
2. $P(J\overline{M})$
3. $P(A|J\overline{M})$
4. $P(B|A)$
5. $P(B|J\overline{M})$
6. $P(J\overline{M}|\overline{B})$

3.2 Diagnosing

Variables and their domais

```
(1)PatientAge:['0-30','31-65','65+']
(2)CTScanResult:['Ischemic Stroke','Hemmorraghic Stroke']
(3)MRIScanResult: ['Ischemic Stroke','Hemmorraghic Stroke']
(4)StrokeType: ['Ischemic Stroke','Hemmorraghic Stroke', 'Stroke Mimic']
(5)Anticoagulants: ['Used','Not used']
(6)Mortality:['True', 'False']
(7)Disability: ['Negligible', 'Moderate', 'Severe']
```

CPTs

Note: [CTScanResult, MRIScanResult,StrokeType] means:

P(StrokeType='...' | CTScanResult='...' ∧ MRIScanResult='...')

```
(1)
[PatientAge]

['0-30', 0.10],
['31-65', 0.30],
['65+', 0.60]


(2)
[CTScanResult]

['Ischemic Stroke',0.7],
[ 'Hemmorraghic Stroke',0.3]


(3)
[MRIScanResult]

['Ischemic Stroke',0.7],
[ 'Hemmorraghic Stroke',0.3]
```

```
(4)
[Anticoagulants]

[Used',0.5],
['Not used',0.5]

(5)
[CTScanResult, MRIScanResult,StrokeType])

['Ischemic Stroke','Ischemic Stroke','Ischemic Stroke',0.8],
['Ischemic Stroke','Hemmorraghic Stroke','Ischemic Stroke',0.5],
[ 'Hemmorraghic Stroke','Ischemic Stroke','Ischemic Stroke',0.5],
[ 'Hemmorraghic Stroke','Hemmorraghic Stroke','Ischemic Stroke',0],

['Ischemic Stroke','Ischemic Stroke','Hemmorraghic Stroke',0],
['Ischemic Stroke','Hemmorraghic Stroke','Hemmorraghic Stroke',0.4],
[ 'Hemmorraghic Stroke','Ischemic Stroke','Hemmorraghic Stroke',0.4],
[ 'Hemmorraghic Stroke','Hemmorraghic Stroke','Hemmorraghic Stroke',0.9],

['Ischemic Stroke','Ischemic Stroke','Stroke Mimic',0.2],
['Ischemic Stroke','Hemmorraghic Stroke','Stroke Mimic',0.1],
[ 'Hemmorraghic Stroke','Ischemic Stroke','Stroke Mimic',0.1],
[ 'Hemmorraghic Stroke','Hemmorraghic Stroke','Stroke Mimic',0.1],

(6)
[StrokeType, Anticoagulants, Mortality]

['Ischemic Stroke', 'Used', 'False',0.28],
['Hemmorraghic Stroke', 'Used', 'False',0.99],
['Stroke Mimic', 'Used', 'False',0.1],
['Ischemic Stroke','Not used', 'False',0.56],
['Hemmorraghic Stroke', 'Not used', 'False',0.58],
['Stroke Mimic', 'Not used', 'False',0.05],
['Ischemic Stroke',  'Used' ,'True',0.72],
['Hemmorraghic Stroke', 'Used', 'True',0.01],
['Stroke Mimic', 'Used', 'True',0.9],
['Ischemic Stroke',  'Not used' ,'True',0.44],
['Hemmorraghic Stroke', 'Not used', 'True',0.42 ],
['Stroke Mimic', 'Not used', 'True',0.95]

(7)
[StrokeType, PatientAge, Disability]

['Ischemic Stroke',   '0-30','Negligible', 0.80],
['Hemmorraghic Stroke', '0-30','Negligible', 0.70],
['Stroke Mimic',        '0-30', 'Negligible',0.9],
['Ischemic Stroke',     '31-65','Negligible', 0.60],
['Hemmorraghic Stroke', '31-65','Negligible', 0.50],
['Stroke Mimic',        '31-65', 'Negligible',0.4],
['Ischemic Stroke',     '65+'  , 'Negligible',0.30],
['Hemmorraghic Stroke', '65+'  , 'Negligible',0.20],
['Stroke Mimic',        '65+'  , 'Negligible',0.1],

['Ischemic Stroke',     '0-30' ,'Moderate',0.1],
['Hemmorraghic Stroke', '0-30' ,'Moderate',0.2],
['Stroke Mimic',        '0-30' ,'Moderate',0.05],
['Ischemic Stroke',     '31-65','Moderate',0.3],
['Hemmorraghic Stroke', '31-65','Moderate',0.4],
['Stroke Mimic',        '31-65','Moderate',0.3],
['Ischemic Stroke',     '65+'  ,'Moderate',0.4],
['Hemmorraghic Stroke', '65+'  ,'Moderate',0.2],
['Stroke Mimic',        '65+'  ,'Moderate',0.1],

['Ischemic Stroke',     '0-30' ,'Severe',0.1],
['Hemmorraghic Stroke', '0-30' ,'Severe',0.1],
['Stroke Mimic',         '0-30'  ,'Severe',0.05],
['Ischemic Stroke',      '31-65','Severe',0.1],
['Hemmorraghic Stroke', '31-65','Severe',0.1],
['Stroke Mimic',         '31-65','Severe',0.3],
['Ischemic Stroke',      '65+'   ,'Severe',0.3],
['Hemmorraghic Stroke', '65+'   ,'Severe',0.6],
['Stroke Mimic',         '65+'   ,'Severe',0.8]
```

Calculation

Please code to calculate the following probability value:

p1 = P(Mortality='True' | PatientAge='31-65' ∧ CTScanResult='Ischemic Stroke')

p2 = P(Disability='Moderate' | PatientAge='65+' ∧ MRIScanResult='Hemmorraghic Stroke')

p3 = P(StrokeType='Stroke Mimic' | PatientAge='65+' ∧ CTScanResult='Hemmorraghic Stroke' ∧ MRIScanResult='Ischemic Stroke')

p4 = P(Anticoagulants='Not used' | PatientAge='0-30')

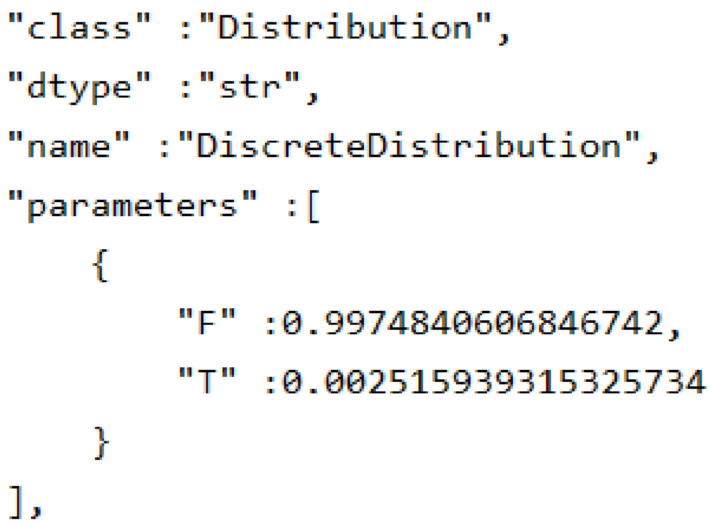
Please solve the 2 tasks and hand in a file named E09_YourNumber.pdf, and send it to ai_201901@foxmail.com

4 Burglary Alarm problem

- **Code:**

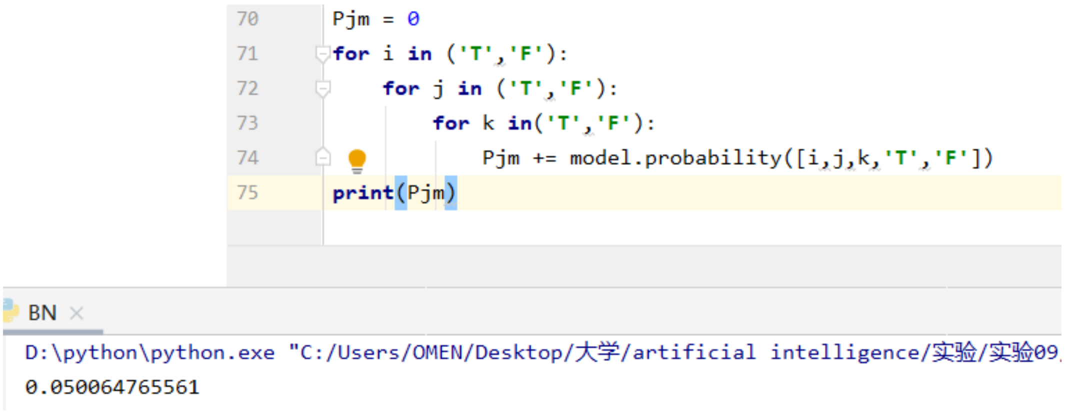
```
from pomegranate import  *


Burglary = DiscreteDistribution ({'T':0.001,'F':0.999})

Earthquake = DiscreteDistribution ({'T':0.002, 'F':0.998})

Alarm = ConditionalProbabilityTable(
    [
        ['T','T','T',0.95],
        ['T','T','F',0.05],
        ['T','F','T',0.94],
        ['T','F','F',0.06],
        ['F','T','T',0.29],
        [’F’,’T’,’F’,0.81],
        [’F’,’F’,’T’,0.001],
        [’F’,’F’,’F’,0.999]
    ],
    [Burglary,Earthquake]
)

JohnCalls = ConditionalProbabilityTable(
    [
        [’T’,’T’, 0.90],
        [’T’,’F’, 0.10],
        [’F’,’T’, 0.05],
        [’F’,’F’, 0.95]
    ],
    [Alarm]
)

MaryCalls = ConditionalProbabilityTable(
    [
        [’T’,’T’, 0.70],
        [’T’,’F’, 0.30],
        [’F’,’T’, 0.01],
        [’F’,’F’, 0.99]
    ],
    [Alarm]
)

s1 = State (Burglary, name = ’Burglary’)
s2 = State (Earthquake , name = ’Burglary’)
s3 = State (Alarm, name = ’Alarm’)
s4 = State (JohnCalls ,name =’JohnCalls’)
s5 = State (MaryCalls ,name =’MaryCalls’)

model = BayesianNetwork ("Burglary Alarm problem")

model.add_states(s1,s2,s3,s4,s5)

model.add_transition(s1,s3)
model.add_transition(s2,s3)
model.add_transition(s3,s4)
model.add_transition(s3,s5)
model.bake()

#print (model.predict_proba({}) )

#print (model.predict_proba({'JohnCalls':'T','MaryCalls':'F'}) )
marginals = model.predict_proba({})
# q1 print(marginals[2].parameters[0])

Pjm = 0
for i in ('T','F'):
    for j in ('T','F'):
        for k in('T','F'):
            Pjm += model.probability([i,j,k,'T','F'])
# q2 print(Pjm)

marginals = model.predict_proba({'JohnCalls':'T','MaryCalls':'F'})
# q3 print(marginals[2].parameters[0])
# q5 print(marginals[0].parameters[0])

marginals = model.predict_proba({'Alarm':'T'})
# q4 print(marginals[0].parameters[0])



marginals = model.predict_proba({})
PfeiB = marginals[0].parameters[0]['F'] # P(~B)

PjmfeiB = 0 # P(J & ~M & ~B)
for i in ('T','F'):
    for j in('T','F'):
        PjmfeiB += model.probability (['F',i,j,'T','F'])

ans = PjmfeiB / PfeiB #P(J&~M|~B)
# q6 print(ans)
```

- **Result**



```
"class" :"Distribution",
"dtype" :"str",
"name" :"DiscreteDistribution",
"parameters" :[
    {
        "F" :0.9974840606846742,
        "T" :0.002515939315325734
    }
],
```

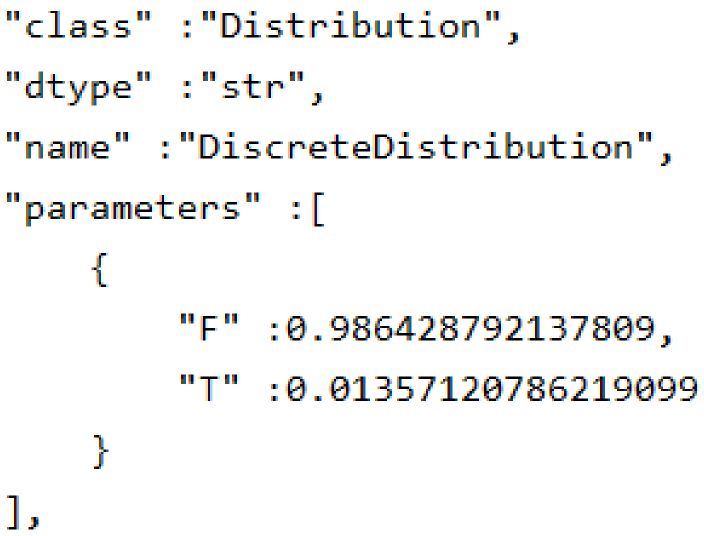
Figure 1: Q1



```
Pjm = 0
for i in ('T','F'):
    for j in ('T','F'):
        for k in('T','F'):
            Pjm += model.probability([i,j,k,'T','F'])
print(Pjm)
```

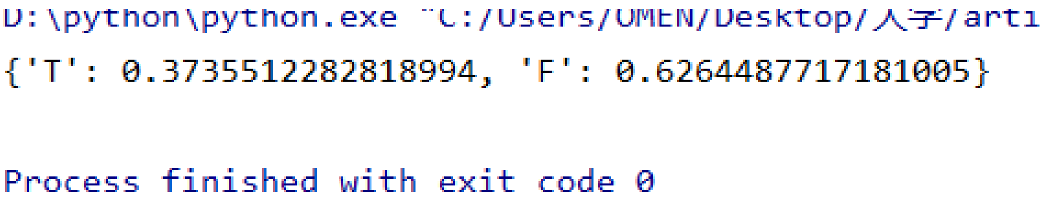
```
D:\python\python.exe "C:/Users/OMEN/Desktop/大学/artificial intelligence/实验/实验09
0.050064765561
```

Figure 2: Q2



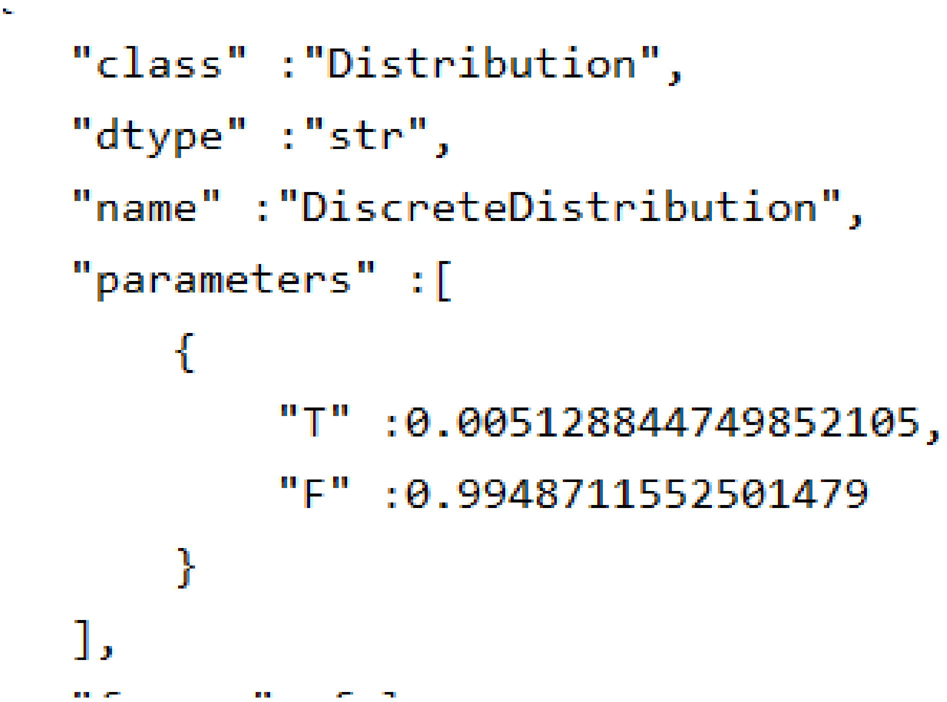
```
"class" :"Distribution",
"dtype" :"str",
"name" :"DiscreteDistribution",
"parameters" :[
    {
        "F" :0.986428792137809,
        "T" :0.01357120786219099
    }
],
```

Figure 3: Q3



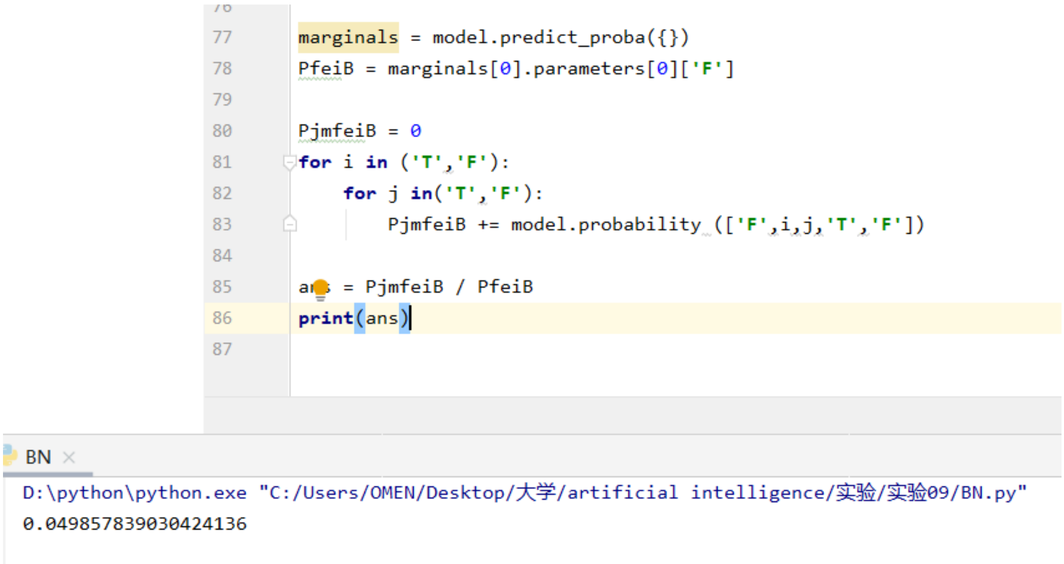
```
D:\python\python.exe "C:/Users/OMEN/Desktop/大学/arti
{'T': 0.3735512282818994, 'F': 0.6264487717181005}

Process finished with exit code 0
```

Figure 4: Q4



```
"class" :"Distribution",
"dtype" :"str",
"name" :"DiscreteDistribution",
"parameters" :[
    {
        "T" :0.005128844749852105,
        "F" :0.994871155250147 9
    }
],
```

Figure 5: Q5



```
marginals = model.predict_proba({})
PfeiB = marginals[0].parameters[0]['F']

PjmfeiB = 0
for i in ('T','F'):
    for j in('T','F'):
        PjmfeiB += model.probability ([ 'F',i,j,'T','F'])

ans = PjmfeiB / PfeiB
print(ans)
```

```
D:\python\python.exe "C:/Users/OMEN/Desktop/大学/artificial intelligence/实验/实验09/BN.py"
0.049857839030424136
```

Figure 6: Q6

5 Diagnosing Problem

- **Code:**

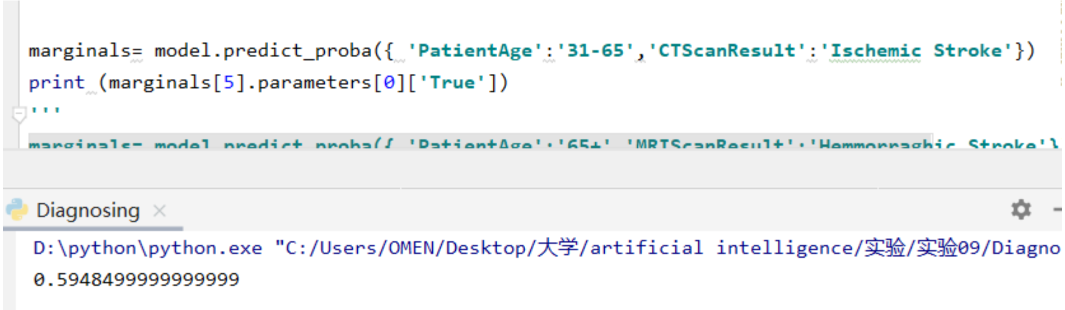
```
from pomegranate import *
```

```
PatientAge = DiscreteDistribution ({'0-30':0.10,'31-65':0.30,'65+':
                                    0.60})
CTScanResult = DiscreteDistribution ({'Ischemic Stroke':0.7 , '
                                    Hemmorraghic Stroke':0.3})
MRIScanResult = DiscreteDistribution ({'Ischemic Stroke':0.7 , '
                                    Hemmorraghic Stroke':0.3})
#StrokeType = DiscreteDistribution ({'Ischemic Stroke': , '
                                    Hemmorraghic Stroke':, 'Stroke
                                    Mimic':})
Anticoagulants = DiscreteDistribution ({'Used':0.5 , 'Not Used':0.5
                                    })
#Mortality = DiscreteDistribution ({'True': , 'False':})


StrokeType = ConditionalProbabilityTable(
    [
        [ 'Ischemic Stroke' , 'Ischemic Stroke' , 'Ischemic Stroke'
                                            , 0.8 ] ,
        [ 'Ischemic Stroke' , 'Hemmorraghic Stroke' , 'Ischemic
                                            Stroke' , 0.5 ] ,
        [ 'Hemmorraghic Stroke' , 'Ischemic Stroke' , 'Ischemic
                                            Stroke' , 0.5 ] ,
        [ 'Hemmorraghic Stroke' , 'Hemmorraghic Stroke' , 'Ischemic
                                            Stroke' , 0 ] ,

        [ 'Ischemic Stroke' , 'Ischemic Stroke' , 'Hemmorraghic
                                            Stroke' , 0 ] ,
        [ 'Ischemic Stroke' , 'Hemmorraghic Stroke' , 'Hemmorraghic
                                            Stroke' , 0.4 ] ,
        [ 'Hemmorraghic Stroke' , 'Ischemic Stroke' , 'Hemmorraghic
                                            Stroke' , 0.4 ] ,
        [ 'Hemmorraghic Stroke' , 'Hemmorraghic Stroke' , '
                                            Hemmorraghic Stroke' ,
                                            0.9 ] ,

        [ 'Ischemic Stroke' , 'Ischemic Stroke' , 'Stroke Mimic' ,
                                            0.2 ] ,
        [ 'Ischemic Stroke' , 'Hemmorraghic Stroke' , 'Stroke Mimic
                                            ' , 0.1 ] ,
        [ 'Hemmorraghic Stroke' , 'Ischemic Stroke' , 'Stroke Mimic
```

```
                                                ' , 0.1 ] ,
        [ 'Hemmorraghic Stroke' , 'Hemmorraghic Stroke' , 'Stroke
                                                Mimic' , 0.1 ]
    ],
    [CTScanResult,MRIScanResult]
)

Mortality = ConditionalProbabilityTable(
    [
        [ 'Ischemic Stroke' , 'Used' , 'False' , 0.28 ] ,
        [ 'Hemmorraghic Stroke' , 'Used' , 'False' , 0.99 ] ,
        [ 'Stroke Mimic' , 'Used' , 'False' , 0.1 ] ,
        [ 'Ischemic Stroke' , 'Not Used' , 'False' , 0.56 ] ,
        [ 'Hemmorraghic Stroke' , 'Not Used' , 'False' , 0.58 ] ,
        [ 'Stroke Mimic' , 'Not Used' , 'False' , 0.05 ] ,
        [ 'Ischemic Stroke' , 'Used' , 'True' , 0.72 ] ,
        [ 'Hemmorraghic Stroke' , 'Used' , 'True' , 0.01 ] ,
        [ 'Stroke Mimic' , 'Used' , 'True' , 0.9 ] ,
        [ 'Ischemic Stroke' , 'Not Used' , 'True' , 0.44 ] ,
        [ 'Hemmorraghic Stroke' , 'Not Used' , 'True' , 0.42 ] ,
        [ 'Stroke Mimic' , 'Not Used' , 'True' , 0.95 ]
    ],
    [StrokeType,Anticoagulants]
)

Disability = ConditionalProbabilityTable(
    [
        [ 'Ischemic Stroke' , '0-30' , 'Negligible' , 0.80 ] ,
        [ 'Hemmorraghic Stroke' , '0-30' , 'Negligible' , 0.70 ] ,
        [ 'Stroke Mimic' , '0-30' , 'Negligible' , 0.9 ] ,
        [ 'Ischemic Stroke' , '31-65' , 'Negligible' , 0.60 ] ,
        [ 'Hemmorraghic Stroke' , '31-65' , 'Negligible' , 0.50 ] ,
        [ 'Stroke Mimic' , '31-65' , 'Negligible' , 0.4 ] ,
        [ 'Ischemic Stroke' , '65+' , 'Negligible' , 0.30 ] ,
        [ 'Hemmorraghic Stroke' , '65+' , 'Negligible' , 0.20 ] ,
        [ 'Stroke Mimic' , '65+' , 'Negligible' , 0.1 ] ,
        [ 'Ischemic Stroke' , '0-30' , 'Moderate' , 0.1 ] ,
        [ 'Hemmorraghic Stroke' , '0-30' , 'Moderate' , 0.2 ] ,
        [ 'Stroke Mimic' , '0-30' , 'Moderate' , 0.05 ] ,
        [ 'Ischemic Stroke' , '31-65' , 'Moderate' , 0.3 ] ,
        [ 'Hemmorraghic Stroke' , '31-65' , 'Moderate' , 0.4 ] ,
        [ 'Stroke Mimic' , '31-65' , 'Moderate' , 0.3 ] ,
        [ 'Ischemic Stroke' , '65+' , 'Moderate' , 0.4 ] ,
        [ 'Hemmorraghic Stroke' , '65+' , 'Moderate' , 0.2 ] ,
        [ 'Stroke Mimic' , '65+' , 'Moderate' , 0.1 ] ,
        [ 'Ischemic Stroke' , '0-30' , 'Severe' , 0.1 ] ,
        [ 'Hemmorraghic Stroke' , '0-30' , 'Severe' , 0.1 ] ,
        [ 'Stroke Mimic' , '0-30' , 'Severe' , 0.05 ] ,
        [ 'Ischemic Stroke' , '31-65' , 'Severe' , 0.1 ] ,
        [ 'Hemmorraghic Stroke' , '31-65' , 'Severe' , 0.1 ] ,
        [ 'Stroke Mimic' , '31-65' , 'Severe' , 0.3 ] ,
        [ 'Ischemic Stroke' , '65+' , 'Severe' , 0.3 ] ,
        [ 'Hemmorraghic Stroke' , '65+' , 'Severe' , 0.6 ] ,
        [ 'Stroke Mimic' , '65+' , 'Severe' , 0.8 ]
    ],
    [StrokeType,PatientAge]
)

'''
Q1 : P(S6|S1 = '31-65' ^ S2 = 'i' )
Q2 : P(s7 = 'M'| s1 = '65+' ^ s3 = 'H')
p1 = P(Mortality=   True     | PatientAge=  31  - 65
                                    CTScanResult=  Ischemic
                                    Stroke )
p2 = P(Disability=   Moderate     | PatientAge=  65  +
                                    MRIScanResult=  Hemmorraghic
                                    Stroke )
p3 = P(StrokeType=  Stroke   Mimic  | PatientAge=  65  +
                                    CTScanResult=  Hemmorraghic
                                    Stroke
    MRIScanResult=  Ischemic   Stroke )
p4 = P(Anticoagulants=  Not   used  | PatientAge=  0  - 30  )

'''

s1 = State (PatientAge, name = 'PatientAge')
s2 = State (CTScanResult , name = 'CTScanResult')
s3 = State (MRIScanResult, name = 'MRIScanResult')
s4 = State (StrokeType ,name = 'StrokeType')
s5 = State (Anticoagulants ,name ='Anticoagulants')
s6 = State (Mortality,name = 'Mortality')
s7 = State (Disability ,name = 'Disability')
```

```
model = BayesianNetwork ("Diagnosing problem")

model.add_states(s1,s2,s3,s4,s5,s6,s7)

model.add_transition(s2,s4)
model.add_transition(s3,s4)
model.add_transition(s4,s6)
model.add_transition(s5,s6)
model.add_transition(s1,s7)
model.add_transition(s4,s7)

model.bake()

marginals= model.predict_proba({ 'PatientAge':'31-65','CTScanResult
                                    ':'Ischemic Stroke'})
#print (marginals[5].parameters[0]['True'])

marginals= model.predict_proba({ 'PatientAge':'65+','MRIScanResult'
                                    :'Hemmorraghic Stroke'})
#print(marginals[6].parameters[0]['Moderate'])

marginals= model.predict_proba({ 'PatientAge':'65+','CTScanResult':
                                    'Hemmorraghic Stroke'\
                               ,'MRIScanResult':'Hemmorraghic
                                                                   Stroke
                                                                   '
                                                                   }

                                                                   )

#print(marginals[3].parameters[0]['Stroke Mimic'])

marginals= model.predict_proba({ 'PatientAge':'0-30'})
#print(marginals[4].parameters[0]["Not Used"])
```

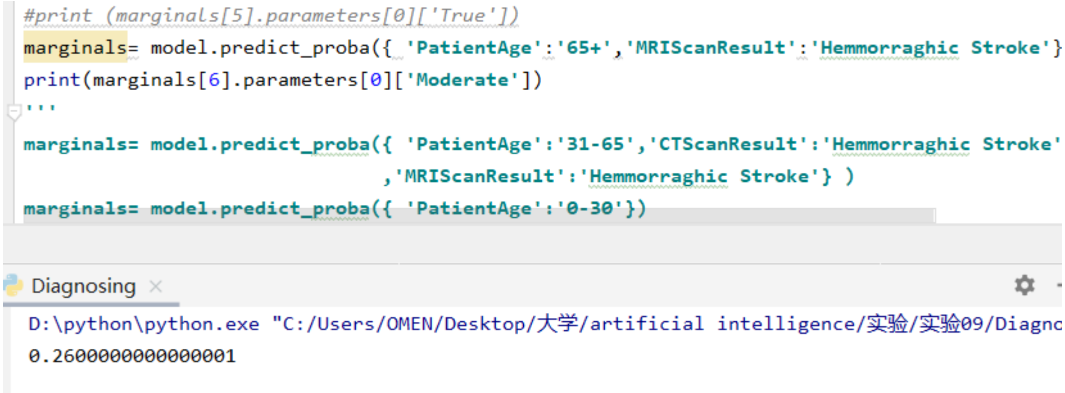
- **Result**



```
marginals= model.predict_proba({ 'PatientAge':'31-65','CTScanResult':'Ischemic Stroke'})
print (marginals[5].parameters[0]['True'])
'''
marginals= model.predict_proba({ 'PatientAge':'65+','MRIScanResult':'Hemmorraghic Stroke'}
```

```
D:\python\python.exe "C:/Users/OMEN/Desktop/大学/artificial intelligence/实验/实验09/Diagno
0.5948499999999999
```

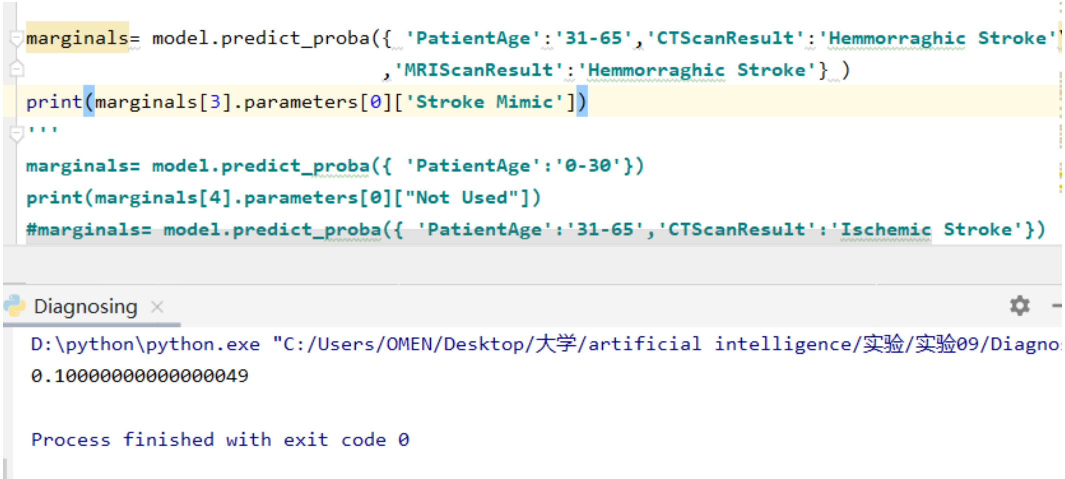
Figure 7: Q1



```
#print (marginals[5].parameters[0]['True'])
marginals= model.predict_proba({ 'PatientAge':'65+','MRIScanResult':'Hemmorraghic Stroke'}
print(marginals[6].parameters[0]['Moderate'])
'''
marginals= model.predict_proba({ 'PatientAge':'31-65','CTScanResult':'Hemmorraghic Stroke'
                                ,'MRIScanResult':'Hemmorraghic Stroke'} )
marginals= model.predict_proba({ 'PatientAge':'0-30'})
```

```
D:\python\python.exe "C:/Users/OMEN/Desktop/大学/artificial intelligence/实验/实验09/Diagnc
0.2600000000000001
```

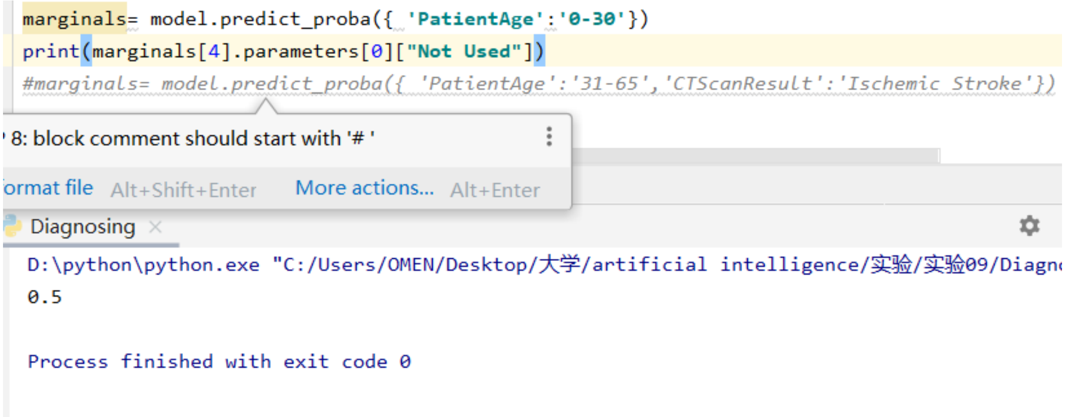
Figure 8: Q2



```
marginals= model.predict_proba({ 'PatientAge':'31-65','CTScanResult':'Hemmorraghic Stroke'
                                ,'MRIScanResult':'Hemmorraghic Stroke'} )
print(marginals[3].parameters[0]['Stroke Mimic'])
'''
marginals= model.predict_proba({ 'PatientAge':'0-30'})
print(marginals[4].parameters[0]["Not Used"])
#marginals= model.predict_proba({ 'PatientAge':'31-65','CTScanResult':'Ischemic Stroke'})
```

```
D:\python\python.exe "C:/Users/OMEN/Desktop/大学/artificial intelligence/实验/实验09/Diagno
0.10000000000000049

Process finished with exit code 0
```

Figure 9: Q3



```
marginals= model.predict_proba({ 'PatientAge':'0-30'})
print(marginals[4].parameters[0]["Not Used"])
#marginals= model.predict_proba({ 'PatientAge':'31-65','CTScanResult':'Ischemic Stroke'})
```

' 8: block comment should start with '# '

```
D:\python\python.exe "C:/Users/OMEN/Desktop/大学/artificial intelligence/实验/实验09/Diagn
0.5

Process finished with exit code 0
```

Figure 10: Q4