AI-Assignment 3

Ritisha Gupta, MT22056

In this assignment, we are making a Career Advisory System restricted for MTech System which is used to suggest careers based on their branch, CGPA, and interest. For that, we are using durable rules and forward chaining.

This is my main function which takes input: Name, Branch, CGPA, and Interest.

In interest, I am giving 2 options: a) Higher Studies; b) Job

```
print("Welcome to Career Advisory System - by IIITD")
print("Please Enter your name.")
name = input()
print('\nHi {}! I am here to help with your career'. format(name))
print("Lets start with asking some Questions.")
print("\n1. Please Enter your branch.")
print("a. CSE\nb. ECE \nc. CB")
branch = input()
print("\n2. Please Enter your current CGPA.")
cgpa = float(input())
print("\n3. What are your future goals?")
print("\na. Looking for job. (Enter JOB) \nb. Thinking to pursue Higher Studies.
(Enter HS)")
interest = input()
print('\n*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
*-*-*-*-*-*\n')
print('According to the inputs given, the best results are as follows. ')
assert_fact('recommend',{'field':branch, 'cg':cgpa, 'goal':interest})
print('\n*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
*-*-*-*-*-*\n')
```

According to the input given: one of the cases would be selected.

```
# based on inputs by user
with ruleset('recommend'):
  #for cse branch + higher studies
  @when_all((m.field == 'CSE') & (m.cg>=8.0) & (m.goal == 'HS'))
  def func(c):
        c.assert_fact('suggestion_HS', {'suggest': 'HigherStudies'})
          c.assert_fact('suggestion_institute',{'suggest_institute':  'higher-cg-
hs'})
```

```
  @when_all((m.field == 'CSE') & (m.cg<8.0) & (m.goal == 'HS'))
  def func(c):
        c.assert_fact('suggestion_HS', {'suggest': 'HigherStudies'})
       c.assert_fact('suggestion_institute',{'suggest_institute': 'lower-cg-hs'})


  #for cse branch + job
  @when_all((m.field == 'CSE') & (m.cg>=8.0) & (m.goal == 'JOB'))
  def func(c):
        c.assert_fact('suggestion_Jobs', {'suggest': 'cse-jobs'})
       c.assert_fact('suggestion_companies',{'suggest_company': 'cse-companies'})

  @when_all((m.field == 'CSE') & (m.cg<8.0) & (m.goal == 'JOB'))
  def func(c):
        c.assert_fact('suggestion_Jobs', {'suggest': 'cse-jobs'})
       c.assert_fact('suggestion_companies',{'suggest_company': 'cse-companies'})


  #for ece branch + higher studies
  @when_all((m.field == 'ECE') & (m.cg>=8.0) & (m.goal == 'HS'))
  def func(c):
        c.assert_fact('suggestion_HS', {'suggest': 'HigherStudies'})
          c.assert_fact('suggestion_institute',{'suggest_institute':  'higher-cg-
hs'})

  @when_all((m.field == 'ECE') & (m.cg<8.0) & (m.goal == 'HS'))
  def func(c):
        c.assert_fact('suggestion_HS', {'suggest': 'HigherStudies'})
       c.assert_fact('suggestion_institute',{'suggest_institute': 'lower-cg-hs'})

  #for ece branch + job
  @when_all((m.field == 'ECE') & (m.cg>=8.0) & (m.goal == 'JOB'))
  def func(c):
        c.assert_fact('suggestion_Jobs', {'suggest': 'ece-jobs'})
       c.assert_fact('suggestion_companies',{'suggest_company': 'ece-companies'})

  @when_all((m.field == 'ECE') & (m.cg<8.0) & (m.goal == 'JOB'))
  def func(c):
        c.assert_fact('suggestion_Jobs', {'suggest': 'ece-jobs'})
       c.assert_fact('suggestion_companies',{'suggest_company': 'ece-companies'})



  #for cb branch + higher studies
```

```
  @when_all((m.field == 'CB') & (m.cg>=8.0) & (m.goal == 'HS'))
  def func(c):
        c.assert_fact('suggestion_HS', {'suggest': 'HigherStudies'})
        c.assert_fact('suggestion_institute',{'suggest_institute': 'cb-higher-cg-
hs'})

  @when_all((m.field == 'CB') & (m.cg<8.0) & (m.goal == 'HS'))
  def func(c):
        c.assert_fact('suggestion_HS', {'suggest': 'HigherStudies'})
         c.assert_fact('suggestion_institute',{'suggest_institute': 'cb-lower-cg-
hs'})


  #for cb branch + job
  @when_all((m.field == 'CB') & (m.cg>=8.0) & (m.goal == 'JOB'))
  def func(c):
        c.assert_fact('suggestion_Jobs', {'suggest': 'cb-jobs'})
        c.assert_fact('suggestion_companies',{'suggest_company': 'cb-companies'})

  @when_all((m.field == 'CB') & (m.cg<8.0) & (m.goal == 'JOB'))
  def func(c):
        c.assert_fact('suggestion_Jobs', {'suggest': 'cb-jobs'})
        c.assert_fact('suggestion_companies',{'suggest_company': 'cb-companies'})
```

In every selected call 2 forward chaining is used by calling a different ruleset. On a broader scale, this is shown.

1) If higher studies are chosen then, possible options, as well as institutes suitable for that, will be suggested.
2) If the Job is selected then, possible roles you can apply for + the companies you can think of will be shown.

This all will be based on the CGPA you have scored in Master, So all the possible cases and permutations are handled. Given below are all the permutations.

- Ruleset for Suggestion of Higher Study.

```
with ruleset('suggestion_HS'):
    @when_all((m.suggest == 'HigherStudies'))
    def mathc(d):
        d.assert_fact({ 'advice': '- PHD' })
```

```
        d.assert_fact({ 'advice': '- MS Abrod' })
        d.assert_fact({ 'advice': '- Research' })
        d.assert_fact({ 'advice': 'For Higher Studies, you can opt either
of the options:' })

    @when_all(+m.advice)
    def output(d):
        print(d.m.advice)
```

- Ruleset for Suggestion of Institutes

```

with ruleset('suggestion_institute'):
    @when_all((m.suggest_institute == 'higher-cg-hs'))
    def mathc(d):
        d.assert_fact({ 'advice': '- IITs, IIITs..' })
        d.assert_fact({ 'advice': '- MIT, Stanford, Harvard..' })
        d.assert_fact({ 'advice': '- IISC Banglore, IIIT Delhi..' })
        d.assert_fact({ 'advice': '\nThe top Institutions you can opt:' })

    @when_all((m.suggest_institute == 'lower-cg-hs'))
    def mathc(d):
        d.assert_fact({ 'advice': '- BITS, VIT Vellore, Thappar..' })
        d.assert_fact({ 'advice': '- Mcgill, Waterloo, Carleton, Queens
Mary University...' })
        d.assert_fact({ 'advice': '- IIST Shibpur..' })
        d.assert_fact({ 'advice': '\nThe top Institutions you can opt:' })

    @when_all((m.suggest_institute == 'cb-higher-cg-hs'))
    def mathc(d):
        d.assert_fact({ 'advice': '- IIITD, University of Hyderabad,
Indian Institute of Science, Banglore..' })
        d.assert_fact({ 'advice': '- The University of Queensland, The
State University of New York,George Mason University..' })
        d.assert_fact({ 'advice': '\nThe top Institutions you can opt:' })

    @when_all((m.suggest_institute == 'cb-lower-cg-hs'))
    def mathc(d):
        d.assert_fact({ 'advice': '- Amity Institute of Biotechnology,
Noida; University School of Biotechnology, Dwarka, New Delhi..' })
        d.assert_fact({ 'advice': '- National University of Singapore,The
University of Melbourne..' })
        d.assert_fact({ 'advice': '\nThe top Institutions you can opt:' })
```

```
•
•       @when_all(+m.advice)
•       def output(d):
•           print(d.m.advice)
```

- Ruleset for Suggestion of Job Roles

```
•   with ruleset('suggestion_Jobs'):
•       @when_all((m.suggest == 'cse-jobs'))
•       def mathc(d):
•           d.assert_fact({ 'advice': '- Web Developer' })
•           d.assert_fact({ 'advice': '- Android Developer' })
•           d.assert_fact({ 'advice': '- Data Scientist' })
•           d.assert_fact({ 'advice': '- Business Analyst' })
•           d.assert_fact({ 'advice': '- Data Engineer' })
•           d.assert_fact({ 'advice': '- SDE (Software Development Engineer'
    })
•           d.assert_fact({ 'advice': 'Jobs roles you can apply for:' })
•
•       @when_all((m.suggest == 'ece-jobs'))
•       def mathc(d):
•           d.assert_fact({ 'advice': '- Electronic Design Engineer' })
•           d.assert_fact({ 'advice': '- PCB Designer' })
•           d.assert_fact({ 'advice': '- Hardware/IOT Engineer' })
•           d.assert_fact({ 'advice': '- VLSI Design Engineer' })
•           d.assert_fact({ 'advice': 'Jobs roles you can apply for:' })
•
•       @when_all((m.suggest == 'cb-jobs'))
•       def mathc(d):
•           d.assert_fact({ 'advice': '- Bioinformatics Engineer' })
•           d.assert_fact({ 'advice': '- Biophysicists' })
•           d.assert_fact({ 'advice': '- Medical Scientists' })
•           d.assert_fact({ 'advice': 'Jobs roles you can apply for:' })
•
•       @when_all(+m.advice)
•       def output(d):
•           print(d.m.advice)
```

- Ruleset for Suggestion of Companies.

```
•   with ruleset('suggestion_companies'):
```

```
    @when_all((m.suggest_company == 'cse-companies'))
    def mathc(d):
        d.assert_fact({ 'advice': '- Infosys, TCS, Capgemini..' })
        d.assert_fact({ 'advice': '- Dream11, WheelsEye, Adobe, Walmart..'
})
        d.assert_fact({ 'advice': '- Google, Facebook, Amazon, Netflix..'
})
        d.assert_fact({ 'advice': '\nThe top companies you can apply for:'
})

    @when_all((m.suggest_company == 'ece-companies'))
    def mathc(d):
      d.assert_fact({ 'advice': '- Mentor Graphics' })
      d.assert_fact({ 'advice': '- Synopsys' })
      d.assert_fact({ 'advice': '- STM Electronics' })
      d.assert_fact({ 'advice': '- Cisco' })
      d.assert_fact({ 'advice': '\nThe top companies you can apply for:'
})

    @when_all((m.suggest_company == 'cb-companies'))
    def mathc(d):
      d.assert_fact({ 'advice': '- MedGenome Labs, Corteva, Kyvor
Genomics..' })
      d.assert_fact({ 'advice': '- Astrazencea Pharma , Corteva, E-Merge
Tech..' })
      d.assert_fact({ 'advice': '- Cognizant , Jubilant Pharmova
Limited..' })
      d.assert_fact({ 'advice': '\nThe top companies you can apply for:'
})

    @when_all(+m.advice)
    def output(d):
        print(d.m.advice)
```

Sample Output 1:

```
Welcome to Career Advisory System - by IIITD
Please Enter your name.
Ritisha

Hi Ritisha! I am here to help with your career
Lets start with asking some Questions.
```

```
1. Please Enter your branch.
a. CSE
b. ECE
c. CB
CSE

2. Please Enter your current CGPA.
9.8

3. What are your future goals?

a. Looking for job. (Enter JOB)
b. Thinking to pursue Higher Studies. (Enter HS)
JOB

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
*-*-*-*-*

According to the inputs given, the best results are as follows.
Jobs roles you can apply for:
- SDE (Software Development Engineer
- Data Engineer
- Business Analyst
- Data Scientist
- Android Developer
- Web Developer

The top companies you can apply for:
- Google, Facebook, Amazon, Netflix..
- Dream11, WheelsEye, Adobe, Walmart..
- Infosys, TCS, Capgemini..

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
*-*-*-*-*
```

Sample Output 2:

```
Welcome to Career Advisory System - by IIITD
Please Enter your name.
Pranoy

Hi Pranoy! I am here to help with your career
Lets start with asking some Questions.

1. Please Enter your branch.
a. CSE
b. ECE
c. CB
ECE

2. Please Enter your current CGPA.
7.8
```

```
3. What are your future goals?

a. Looking for job. (Enter JOB)
b. Thinking to pursue Higher Studies. (Enter HS)
HS

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
*-*-*-*-*

According to the inputs given, the best results are as follows.
For Higher Studies, you can opt either of the options:
- Research
- MS Abrod
- PHD

The top Institutions you can opt:
- IIST Shibpur..
- Mcgill, Waterloo, Carleton, Queens Mary University...
- BITS, VIT Vellore, Thappar..

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
*-*-*-*-*
```

Sample Output 3:

```
Welcome to Career Advisory System - by IIITD
Please Enter your name.
Kirti

Hi Kirti! I am here to help with your career
Lets start with asking some Questions.

1. Please Enter your branch.
a. CSE
b. ECE
c. CB
CB

2. Please Enter your current CGPA.
8.75

3. What are your future goals?

a. Looking for job. (Enter JOB)
b. Thinking to pursue Higher Studies. (Enter HS)
JOB

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
*-*-*-*-*

According to the inputs given, the best results are as follows.
```

Jobs roles you can apply for:
- Medical Scientists
- Biophysicists
- Bioinformatics Engineer

The top companies you can apply for:
- Cognizant , Jubilant Pharmova Limited..
- Astrazencea Pharma , Corteva, E-Merge Tech..
- MedGenome Labs, Corteva, Kyvor Genomics..

*-
--*-*-*