CSE 643: Artificial Intelligence
Assignment 3

Meetakshi Setiya, 2019253

Implementation Details

I wrote a python script using durable rules engine that smartly recommends possible career paths to IIITD students based on the following parameters:

- Branch
- Interest
- Internship(s) done
- Core courses done
- Grades obtained
- BTP/IP done

The possible career paths the program has in its knowledge base are:

- Software Developer
- Security Engineer
- AI/ML Engineer
- Data Scientist
- Consultant
- Computational Biologist
- Graphic Designer
- Hardware Engineer
- Researcher

The program first shows the user the list of possible career paths and asks the number of paths they would be interested in exploring further. Once the user gets done with entering the number and names of the career tracks they are interested in, the program also takes information about the number and tracks of the internships the user has done. Besides this, I have also defined a mapping that matches the branch at IIITD with the possible careers open to people who would be graduating from that branch. The mapping is as follows:

- **CSE:** software developer, security engineer, ai/ml engineer, researcher
- **CSAM:** software developer, data scientist, researcher
- **CSD:** software developer, graphic designer, researcher
- **CSAI:** software developer, data scientist, ai/ml engineer, researcher
- **CSSS:** software developer, consultant, researcher

- **CSB:** software developer, computational biologist, researcher
- **ECE:** hardware engineer, researcher

Once all this information is collected, the program assigns priority to each career path by giving them pre-defined weights based on the user's involvement in these fields. These weights will be mentioned later. Since the initial priorities are assigned now, the program asserts these facts (containing all career paths and their corresponding initial priority) for the "preferred_courses" ruleset.

In the "preferred_courses" ruleset, the program further narrows down the options by showing a list of courses at IIITD to the user that are core to a certain career path. The program also asks for the average GPA of the user in the courses they have taken. For the research track, I have asked for BTPs, IPs , grades in the said projects and the number of publications the student has. All of this new information is asserted for the "update_priorities" ruleset. The priorities are updated again, keeping the previous priorities and the new information in mind. The final facts are once again asserted to the "update_priorities" ruleset. The final path-priority facts are caught and the paths and their priorities are pushed into a dictionary. This dictionary is sorted in descending order per the priority values and the top n paths with priority value>=3 are reported as the most suitable choices for the candidate.

Here are the scoring rules for assigning priorities:

- interest: +2
- preferred course depending on the branch: +1
- courses taken: +1 per course if avg gpa >= 8.5 (for > 2 courses)
- courses taken: +0.5 per course if avg gpa < 8.5 and >= 7 (for > 2 courses)
- internship done: +5 per track
- btp publication and grade>=9: +5 per publication
- ip publication and grade>=9: +3 per publication
- grade>=9 but no btp publication: +2
- grade>=9 but no ip_publication: +1

(No research points for grade <9 because the student would not likely get an LOR if the professor has given them a poor grade).

If none of the paths have a priority score greater than or equal to 3, the AI informs the user that enough data is not available to make relevant recommendations.

Run the Code

1. Make sure that durable-rules is installed on your system.
2. Run the code by running: python3 main.py
3. Enter the requisite inputs the program asks

The preferred career paths would be displayed at the end of the execution.
Just for information, the program uses assert statements to check the validity of inputs.

Code

```
from durable.lang import *
from colors import *

'''
scores:

interest: +2
branch preferred: +1
courses taken: +1 per course if avg gpa >= 8.5 (for > 2 courses)
courses taken: +0.5 per course if avg gpa < 8.5 and >= 7 (for > 2 courses)
internship done: +5 per
btp publication and grade>=9: +5 per publication
ip publication and grade>=9: +3 per publication
grade>=9 but no btp publication: +2
grade>=9 but no ip_publication: +1
'''

recommendations = {}

career_stream_mapping = {
    "cse": ["software developer", "security engineer", "ai/ml engineer", "researcher"],
    "csam": ["software developer", "data scientist", "researcher"],
    "csd": ["software developer", "graphic designer", "researcher"],
    "csai": ["software developer", "data scientist", "ai/ml engineer", "researcher"],
    "csss": ["software developer", "consultant", "researcher"],
    "csb": ["software developer", "computational biologist", "researcher"],
    "ece": ["hardware engineer", "researcher"],
}

career_courses_mapping = {
    "software developer": ["software development using open-source", "data structures
and algorithms", "advanced programming", "fundamentals of database management
systems"],
    "security engineer": ["foundations of computer security", "security engineering",
"computer networks", "network security"],
    "ai/ml engineer": ["machine learning", "artificial intelligence", "deep learning",
"human-centered ai"],
    "data scientist": ["data science", "machine learning", "big data analytics", "data
mining"],
    "graphic designer": ["design processes and prespectives", "human computer
interaction", "inclusive design & accessibility", "introduction to animation and
graphics"],
```

```
    "consultant": ["game theory", "foundation of finance", "market design", "industrial
organisation"],
    "computational biologist": ["introduction to quantitative biology", "computer aided
drug design", "algorithms in computational biology", "cell biology and biochemistry"],
    "hardware engineer": ["embedded logic design", "memory testing and design",
"digital hardware design", "digital circuits"],
}


def give_recommendations():
    global recommendations
    count = 0
    sorted_recommendations = sort_dictionary(recommendations)
    print(color("\n\nand done! here are future career paths the ai suggests for you: ",
          fg="blue", style="bold"))
    print(color("----------------------------------------------------------------",
fg="blue"))
    for k, v in sorted_recommendations.items():
        if v >= 3:
            count += 1
            print(color(f"preference {count}: {k}", fg="green"))
    if count == 0:
        print(color("enough data not available to make relevant recommendation.",
fg="red"))


def pathwise_course_gpa(path, priority):
    print(color(f"\n==== evaluating career path {path} ====", fg="cyan"))
    print(
        color(f"path relevant courses: {career_courses_mapping[path]}", fg="purple"))
    num_courses = int(input(
        "\nenter the number of courses you have done out of the ones shown above? "))
    assert num_courses >= 0 and num_courses < 5, "wrong number of courses"
    avg_gpa = float(input(
        "enter your average gpa in the courses you had taken out of the above (0 if not
applicable): "))
    assert avg_gpa >= 0 and avg_gpa <= 10, "invalid gpa"
    fact = {"courses_taken": num_courses, "course_path": path,
            "avg_gpa": avg_gpa, "path_priority": priority}
    return fact


def sort_dictionary(dictionary):
    sorted_dictionary = []
    values = list(dictionary.values())
    values.sort(reverse=True)
    for i in range(len(values)):
```

```
        for key, value in dictionary.items():
            if value == values[i] and (key, value) not in sorted_dictionary:
                sorted_dictionary.append((key, value))
    sorted_dictionary = dict(sorted_dictionary)
    return sorted_dictionary


with ruleset('preferred_courses'):

    @when_all((m.path == "software developer"))
    def softwareDev(c):
        fact = pathwise_course_gpa(c.m.path, c.m.priority)
        c.assert_fact("update_priorities", fact)

    @when_all((m.path == "security engineer"))
    def securityEngg(c):
        fact = pathwise_course_gpa(c.m.path, c.m.priority)
        c.assert_fact("update_priorities", fact)

    @when_all((m.path == "data scientist"))
    def dataScientist(c):
        fact = pathwise_course_gpa(c.m.path, c.m.priority)
        c.assert_fact("update_priorities", fact)

    @when_all((m.path == "ai/ml engineer"))
    def aimlEngg(c):
        fact = pathwise_course_gpa(c.m.path, c.m.priority)
        c.assert_fact("update_priorities", fact)

    @when_all((m.path == "consultant"))
    def consultant(c):
        fact = pathwise_course_gpa(c.m.path, c.m.priority)
        c.assert_fact("update_priorities", fact)

    @when_all((m.path == "hardware engineer"))
    def hardwareEngg(c):
        fact = pathwise_course_gpa(c.m.path, c.m.priority)
        c.assert_fact("update_priorities", fact)

    @when_all((m.path == "computational biologist"))
    def compBiologist(c):
        fact = pathwise_course_gpa(c.m.path, c.m.priority)
        c.assert_fact("update_priorities", fact)

    @when_all((m.path == "graphic designer"))
    def graphicDes(c):
        fact = pathwise_course_gpa(c.m.path, c.m.priority)
```

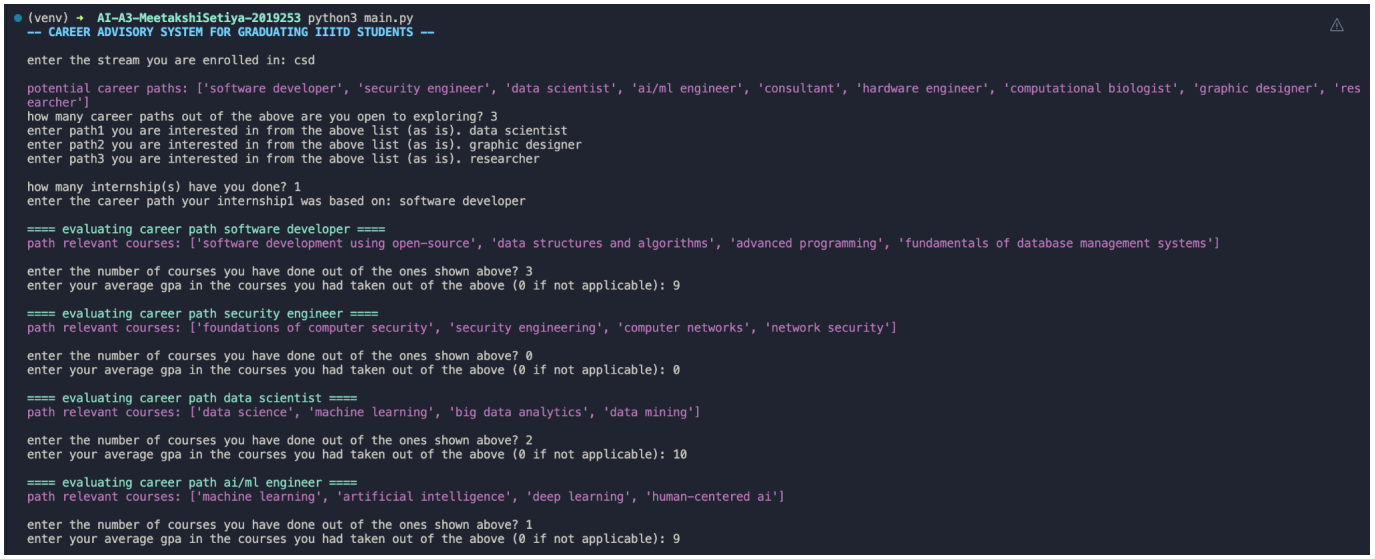
```
        c.assert_fact("update_priorities", fact)

    @when_all((m.path == "researcher"))
    def researcher(c):
        print(
            color(f"\n==== evaluating career path {c.m.path} ====", fg="cyan"))
        btp_grade = 0
        ip_grade = [0, 0]
        btp_publications = 0
        ip_publications = [0, 0]
        btp = input("did you take any btp? (y/n) ")
        if btp == "y":
            btp_grade += int(input("what was your btp grade? "))
            btp_publications += int(
                input("how many publications did you get out of your btp? "))
        ips = int(input("how many ip(s) did you do? (0/1/2) "))
        for i in range(ips):
            ipg = int(input(f"what was your grade in ip{i+1}? "))
            ip_grade[i] = ipg
            ipp = int(
                input(f"how many publications did you get out of this ip{i+1}? "))
            ip_publications[i] = ipp
        fact = {"course_path": c.m.path, "path_priority": c.m.priority, "btp_grade":
btp_grade, "btp_pub": btp_publications,
                "ip_grade1": ip_grade[0], "ip_pub1": ip_publications[0], "ip_grade2":
ip_grade[1], "ip_pub2": ip_publications[1]}
        c.assert_fact("update_priorities", fact)


with ruleset('update_priorities'):

    @when_all((m.courses_taken > 2) & (m.avg_gpa >= 8.5))
    def update(c):
        new_priority = c.m.courses_taken + c.m.path_priority
        fact = {"final_path": c.m.course_path, "final_priority": new_priority}
        c.assert_fact("update_priorities", fact)

    @when_all((m.courses_taken > 2) & (m.avg_gpa < 8.5) & (m.avg_gpa >= 7))
    def update(c):
        new_priority = 0.5*c.m.courses_taken + c.m.path_priority
        fact = {"final_path": c.m.course_path, "final_priority": new_priority}
        c.assert_fact("update_priorities", fact)

    @when_all((m.courses_taken <= 2) | (m.avg_gpa < 7))
    def update(c):
        fact = {"final_path": c.m.course_path,
                "final_priority": c.m.path_priority}
```

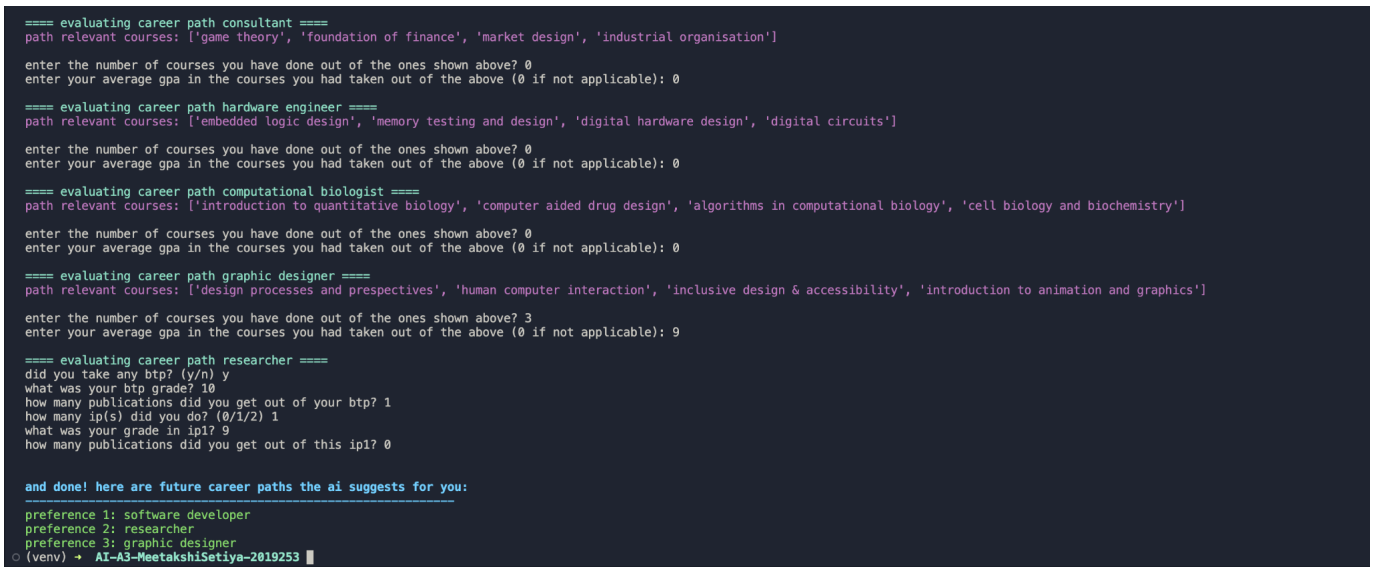
```
        c.assert_fact("update_priorities", fact)

    @when_all((m.course_path == "researcher"))
    def update(c):
        add_priority = 0
        if c.m.btp_grade >= 9:
            add_priority += c.m.btp_pub*5 if c.m.btp_pub > 0 else 2
        if c.m.ip_grade1 >= 9:
            add_priority += c.m.ip_pub1*3 if c.m.ip_pub1 > 0 else 1
        if c.m.ip_grade2 >= 9:
            add_priority += c.m.ip_pub2*3 if c.m.ip_pub2 > 0 else 1
        new_priority = c.m.path_priority + add_priority
        fact = {"final_path": c.m.course_path,
                "final_priority": new_priority}
        c.assert_fact("update_priorities", fact)

    @when_all((+m.final_path))
    def push_to_dict(c):
        recommendations[c.m.final_path] = c.m.final_priority


def main():
    priorities = {}
    internships = []
    interests = []

    print(color("-- CAREER ADVISORY SYSTEM FOR GRADUATING IIITD STUDENTS --",
          style="bold", fg="blue"))

    stream = input("\nenter the stream you are enrolled in: ")
    assert stream in ["cse", "csam", "ece", "csd",
                      "csss", "csb", "csai"], "incorrect stream"

    paths = ["software developer", "security engineer", "data scientist", "ai/ml
engineer", "consultant", "hardware engineer",
             "computational biologist", "graphic designer", "researcher"]
    print(color(f"\npotential career paths: {paths}", fg="purple"))
    num_interests = int(
        input("how many career paths out of the above are you open to exploring? "))
    assert num_interests >= 0 and num_interests <= 9, "invalid number of paths"

    for i in range(num_interests):
        interest = input(
            f"enter path{i+1} you are interested in from the above list (as is). ")
        assert interest in paths, "path does not exist in the knowledge base"
        assert interest not in interests, "please enter a path only once"
        interests.append(interest)
```

```
    num_internships = int(input("\nhow many internship(s) have you done? "))
    for i in range(num_internships):
        internship_path = input(
            f"enter the career path your internship{i+1} was based on: ")
        assert internship_path in paths, "path does not exist in the knowledge base"
        internships.append(internship_path)

    for i in paths:
        priority = 0
        if i in interests:
            priority += 2
        if i in career_stream_mapping[stream]:
            priority += 1
        if i in internships:
            priority += 5
        priorities[i] = priority

    post('preferred_courses', {
        "path": "software developer", "priority": priorities["software developer"]})
    post('preferred_courses', {
        "path": "security engineer", "priority": priorities["security engineer"]})
    post('preferred_courses', {
        "path": "data scientist", "priority": priorities["data scientist"]})
    post('preferred_courses', {
        "path": "ai/ml engineer", "priority": priorities["ai/ml engineer"]})
    post('preferred_courses', {
        "path": "consultant", "priority": priorities["consultant"]})
    post('preferred_courses', {
        "path": "hardware engineer", "priority": priorities["hardware engineer"]})
    post('preferred_courses', {
        "path": "computational biologist", "priority": priorities["computational
biologist"]})
    post('preferred_courses', {
        "path": "graphic designer", "priority": priorities["graphic designer"]})
    post('preferred_courses', {
        "path": "researcher", "priority": priorities["researcher"]})
    give_recommendations()


main()
```

Screenshots

Example 1:





Example 2:

```
(venv) ➜  AI-A3-MeetakshiSetiya-2019253 python3 main.py
-- CAREER ADVISORY SYSTEM FOR GRADUATING IIITD STUDENTS --

enter the stream you are enrolled in: cse

potential career paths: ['software developer', 'security engineer', 'data scientist', 'ai/ml engineer', 'consultant', 'hardware engineer', 'computational biologist', 'graphic designer', 'researcher']
how many career paths out of the above are you open to exploring? 0

how many internship(s) have you done? 1
enter the career path your internship1 was based on: researcher

==== evaluating career path software developer ====
path relevant courses: ['software development using open-source', 'data structures and algorithms', 'advanced programming', 'fundamentals of database management systems']

enter the number of courses you have done out of the ones shown above? 4
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 8

==== evaluating career path security engineer ====
path relevant courses: ['foundations of computer security', 'security engineering', 'computer networks', 'network security']

enter the number of courses you have done out of the ones shown above? 1
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 10

==== evaluating career path data scientist ====
path relevant courses: ['data science', 'machine learning', 'big data analytics', 'data mining']

enter the number of courses you have done out of the ones shown above? 2
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 7

==== evaluating career path ai/ml engineer ====
path relevant courses: ['machine learning', 'artificial intelligence', 'deep learning', 'human-centered ai']

enter the number of courses you have done out of the ones shown above? 0
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 0
```

```
==== evaluating career path consultant ====
path relevant courses: ['game theory', 'foundation of finance', 'market design', 'industrial organisation']

enter the number of courses you have done out of the ones shown above? 0
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 0

==== evaluating career path hardware engineer ====
path relevant courses: ['embedded logic design', 'memory testing and design', 'digital hardware design', 'digital circuits']

enter the number of courses you have done out of the ones shown above? 0
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 0

==== evaluating career path computational biologist ====
path relevant courses: ['introduction to quantitative biology', 'computer aided drug design', 'algorithms in computational biology', 'cell biology and biochemistry']

enter the number of courses you have done out of the ones shown above? 3
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 9

==== evaluating career path graphic designer ====
path relevant courses: ['design processes and prespectives', 'human computer interaction', 'inclusive design & accessibility', 'introduction to animation and graphics']

enter the number of courses you have done out of the ones shown above? 0
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 0

==== evaluating career path researcher ====
did you take any btp? (y/n) n
how many ip(s) did you do? (0/1/2) 1
what was your grade in ip1? 10
how many publications did you get out of this ip1? 0


and done! here are future career paths the ai suggests for you:
------------------------------------------------------------
preference 1: researcher
preference 2: software developer
preference 3: computational biologist
(venv) ➜  AI-A3-MeetakshiSetiya-2019253
```

Example 3:

```
(venv) ➜  AI-A3-MeetakshiSetiya-2019253 python3 main.py
-- CAREER ADVISORY SYSTEM FOR GRADUATING IIITD STUDENTS --

enter the stream you are enrolled in: csai

potential career paths: ['software developer', 'security engineer', 'data scientist', 'ai/ml engineer', 'consultant', 'hardware engineer', 'computational biologist', 'graphic designer', 'res
earcher']
how many career paths out of the above are you open to exploring? 0

how many internship(s) have you done? 0

==== evaluating career path software developer ====
path relevant courses: ['software development using open-source', 'data structures and algorithms', 'advanced programming', 'fundamentals of database management systems']

enter the number of courses you have done out of the ones shown above? 0
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 0

==== evaluating career path security engineer ====
path relevant courses: ['foundations of computer security', 'security engineering', 'computer networks', 'network security']

enter the number of courses you have done out of the ones shown above? 0
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 0

==== evaluating career path data scientist ====
path relevant courses: ['data science', 'machine learning', 'big data analytics', 'data mining']

enter the number of courses you have done out of the ones shown above? 0
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 0

==== evaluating career path ai/ml engineer ====
path relevant courses: ['machine learning', 'artificial intelligence', 'deep learning', 'human-centered ai']

enter the number of courses you have done out of the ones shown above? 0
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 0
```

```
==== evaluating career path consultant ====
path relevant courses: ['game theory', 'foundation of finance', 'market design', 'industrial organisation']

enter the number of courses you have done out of the ones shown above? 0
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 0

==== evaluating career path hardware engineer ====
path relevant courses: ['embedded logic design', 'memory testing and design', 'digital hardware design', 'digital circuits']

enter the number of courses you have done out of the ones shown above? 0
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 0

==== evaluating career path computational biologist ====
path relevant courses: ['introduction to quantitative biology', 'computer aided drug design', 'algorithms in computational biology', 'cell biology and biochemistry']

enter the number of courses you have done out of the ones shown above? 0
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 0

==== evaluating career path graphic designer ====
path relevant courses: ['design processes and prespectives', 'human computer interaction', 'inclusive design & accessibility', 'introduction to animation and graphics']

enter the number of courses you have done out of the ones shown above? 0
enter your average gpa in the courses you had taken out of the above (0 if not applicable): 0

==== evaluating career path researcher ====
did you take any btp? (y/n) 0
how many ip(s) did you do? (0/1/2) 0


and done! here are future career paths the ai suggests for you:
------------------------------------------------------------
enough data not available to make relevant recommendation.
(venv) ➜  AI-A3-MeetakshiSetiya-2019253
```