
	Marathwada Mitra Mandal's	
	Institute of Technology, Lohgaon Pune - 47	
	Department of Artificial Intelligence and Data Science	

Semester -I A.Y.2025-26 Sub.: - Artificial Intelligence Lab

Class: SE

Assignment 02: Building an Expert System Using Rule-Based Systems

Objective: Develop an Expert System that provides simple decision-making.

Problem Statement: Creating a simple Expert System that can be demonstrated to introduce Artificial Intelligence, decision-making algorithms, and rule-based systems.

"Expert System for Career Path Suggestion Based on Student

Interests" What is an Expert System?

An **Expert System** mimics the decision-making ability of a human expert. It uses a set of rules and a knowledge base to make decisions or solve problems in a specific domain.

Tools and Technologies:

- **Language:** Python
- **Interface:** CLI
- **Logic Engine:** PyKnow (Python library for Expert Systems)

DR. GAIKWAD KIRAN P.

Knowledge Base (Sample Rules):

IF student_likes == "Maths" AND student_likes == "Physics" THEN suggest
"Mechanical Engineering"

IF student_likes == "Programming" AND student_likes == "Maths" THEN
suggest "Computer Engineering"

IF student_likes == "Biology" AND student_likes == "Chemistry" THEN
suggest "Biotechnology"

IF student_likes == "Circuits" AND student_likes == "Maths" THEN suggest
"Electronics Engineering"

IF student_likes == "Programming" AND student_likes == "Statistics" THEN
suggest "Artificial Intelligence and Data Science"

IF student_likes == "Programming" AND student_likes == "AI Concepts" THEN
suggest "Artificial Intelligence and Machine Learning Engineering"

Students will develop the expert system/decision making using if else in python and then can go for the following implementation using “**experta**” library.

NOTE: For following code execution, your system needs an “**experta**” python library installed. -----Following is implementation using “experta” python library----- **Code:**

```
from experta import *  
class StudentFacts(Fact):
```

pass

class CareerExpertSystem(KnowledgeEngine):

DR. GAIKWAD KIRAN P.

@Rule(StudentFacts(likes='Maths'), StudentFacts(likes='Physics'))

def mechanical(self):

print("Suggested Career Path: Mechanical Engineering")

@Rule(StudentFacts(likes='Programming'),

StudentFacts(likes='Maths')) def computer(self):

print("Suggested Career Path: Computer Engineering")

@Rule(StudentFacts(likes='Biology'),

StudentFacts(likes='Chemistry')) def biotech(self):

print("Suggested Career Path: Biotechnology")

@Rule(StudentFacts(likes='Circuits'), StudentFacts(likes='Maths'))

def electronics(self):

print("Suggested Career Path: Electronics Engineering")

def main():

engine = CareerExpertSystem()

engine.reset()

print("Welcome to the Career Path Expert System!")

interests = input("Enter your interests separated by commas (e.g., Maths, Physics, Programming): ").split(',')

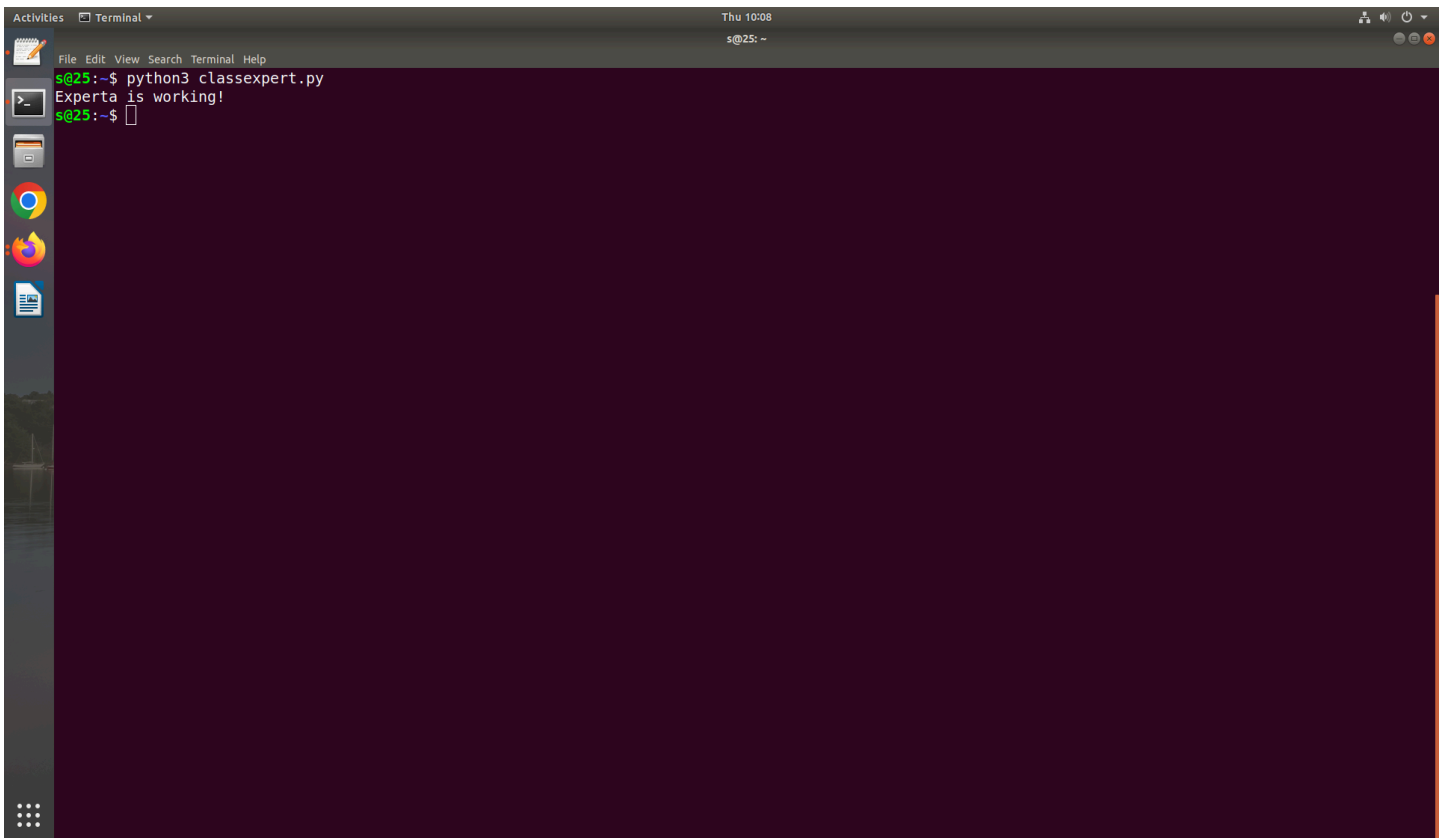
for interest in interests:

engine.declare(StudentFacts(likes=interest.strip()))

engine.run()

if __name__ == "__main__":

main()

A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Thu 10:08, s@25: ~). The terminal shows a command prompt 's@25:~\$' followed by the command 'python3 classexpert.py'. The output is 'Experta is working!' followed by another prompt 's@25:~\$'. The terminal has a dark purple background. On the left, there is a vertical dock with icons for a file manager, terminal, web browser, and other applications. The top of the window shows system status icons like network, volume, and power.

```
Activities Terminal Thu 10:08 s@25: ~
File Edit View Search Terminal Help
s@25:~$ python3 classexpert.py
Experta is working!
s@25:~$
```

```
class ExpertSystem:
    def __init__(self):
        self.knowledge_base = []
        self.facts = set()

    def ask(self, question):
        answer = input(question + " (yes/no): ").strip().lower()
        return answer == "yes"

    def add_rule(self, condition, conclusion):
        self.knowledge_base.append((condition, conclusion))

    def evaluate(self):
        applied = True
```

```
while applied:
    applied = False
    for condition, conclusion in self.knowledge_base:
        if condition(self.facts) and conclusion not in self.facts:
            print(f"Inferred: {conclusion}")
            self.facts.add(conclusion)
            applied = True
```

```
def run(self):
    print("🌿 Welcome to the Plant Doctor Expert System!")
    # Initial user facts
    if self.ask("Are the leaves yellow?"):
        self.facts.add("yellow_leaves")
    if self.ask("Is the soil wet?"):
        self.facts.add("wet_soil")
    if self.ask("Is the plant in direct sunlight?"):
        self.facts.add("direct_sunlight")

    self.evaluate()

    print("\n✅ Final conclusions:")
    for fact in self.facts:
        if fact.startswith("diagnosis:"):
            print("-", fact.replace("diagnosis:", ""))
```

Define the system

```
system = ExpertSystem()
```

Add rules (condition, conclusion)

```
system.add_rule(lambda facts: "yellow_leaves" in facts and "wet_soil" in facts,
                 "diagnosis:Overwatering")
```

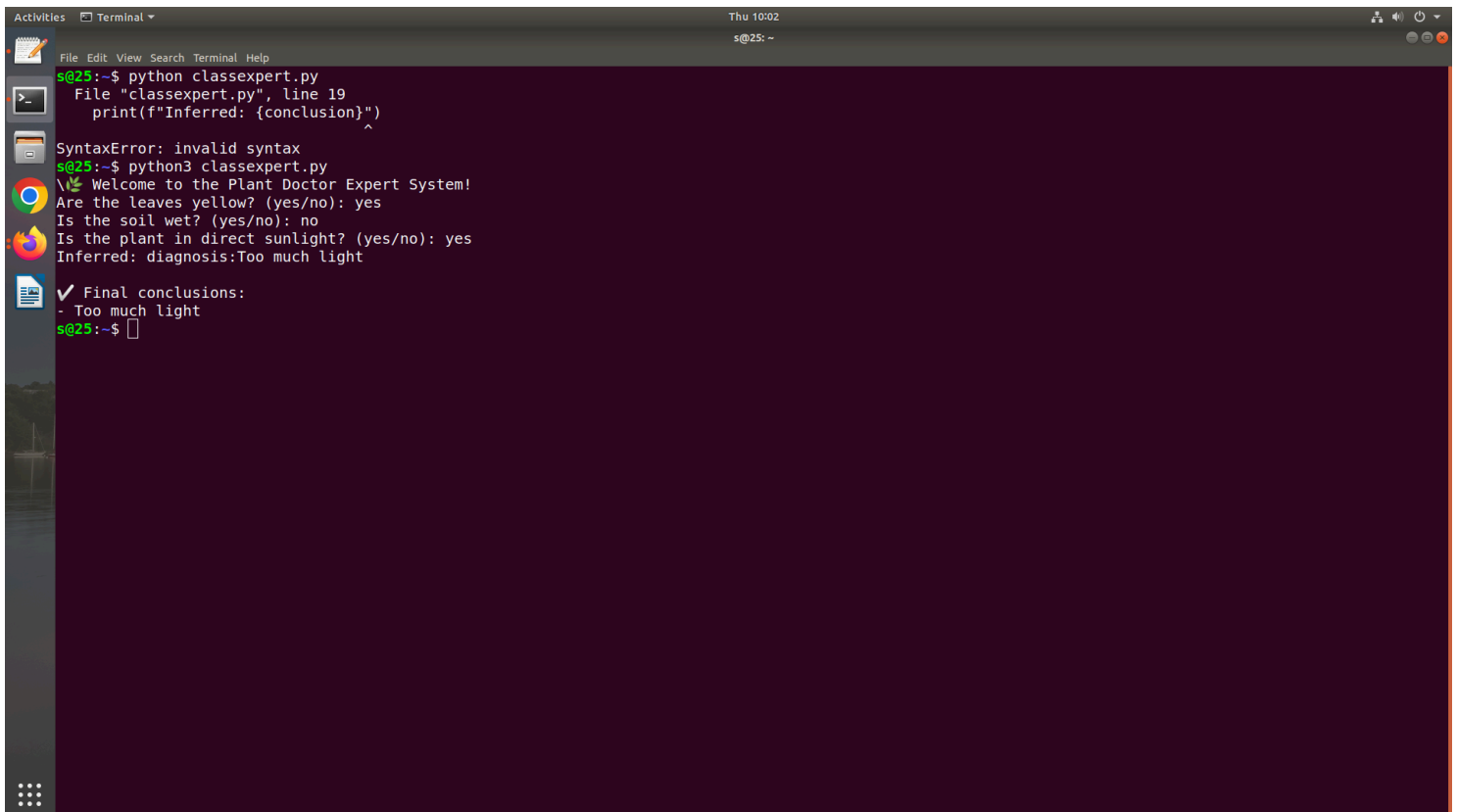
```
system.add_rule(lambda facts: "yellow_leaves" in facts and "direct_sunlight" in facts,
                 "diagnosis:Sunburn")
```

```
system.add_rule(lambda facts: "yellow_leaves" in facts and "wet_soil" not in facts,
```

```
        "diagnosis:Underwatering")
system.add_rule(lambda facts: "yellow_leaves" not in facts and "direct_sunlight" in facts,
                "diagnosis:Too much light")
```

```
# Run the system
```

```
if __name__ == "__main__":
    system.run()
```



```
Activities Terminal Thu 10:02
s@25: ~
s@25:~$ python classexpert.py
File "classexpert.py", line 19
    print(f"Inferred: {conclusion}")
          ^
SyntaxError: invalid syntax
s@25:~$ python3 classexpert.py
\\ Welcome to the Plant Doctor Expert System!
Are the leaves yellow? (yes/no): yes
Is the soil wet? (yes/no): no
Is the plant in direct sunlight? (yes/no): yes
Inferred: diagnosis:Too much light

✓ Final conclusions:
- Too much light
s@25:~$
```

