EXPERIMENT NO. 09

Student Name and Roll Number: Piyush Gambhir – 21CSU349
Semester /Section: Semester-V – AIML-V-B (AL-3)
Link to Code: NCU-Lab-Manual-And-End-Semester-Projects/NCU-CSL347-AAIES-Lab_Manual at main · Piyush-Gambhir/NCU-Lab-Manual-And-End-Semester-Projects (github.com)
Date: 28.10.2023
Faculty Signature:
Grade:

Objective(s):

- Understand and study Simple Expert Systems.

Outcome:

Students will be familiarized with *Simple Expert Systems.*

Problem Statement:

A manufacturing company is trying to improve their quality control process. They have a dataset of historical quality data, which includes the results of quality tests, as well as the specifications for each product. Implement a Simple Expert system to identify potential quality problems.

The dataset is:

Product ID	Test 1 Result	Test 2 Result	Test 3 Result	Specification 1	Specification 2	Specification 3
1	95	20	8	100	25	10
2	98	22	9	100	20	10
3	93	18	7	95	15	8
4	100	24	10	100	25	12
5	96	21	8	98	20	10
6	92	19	6	95	15	8
7	90	17	5	95	15	8

Background Study:

A simple expert system is a type of artificial intelligence that uses a set of predefined rules or knowledge to make decisions or solve problems in a specific domain. It is designed to mimic the decision-making process of a human expert in a particular field. The system consists of a knowledge base, which stores the rules and facts about the domain, and an inference engine, which uses logical reasoning to derive conclusions from the available knowledge.

Question Bank:

1. What are Expert Systems?
   Expert Systems: Expert systems are AI programs designed to mimic the decision-making abilities of human experts in specific domains. They use knowledge, rules, and reasoning mechanisms to provide advice, solve problems, or make recommendations within their specialized areas.

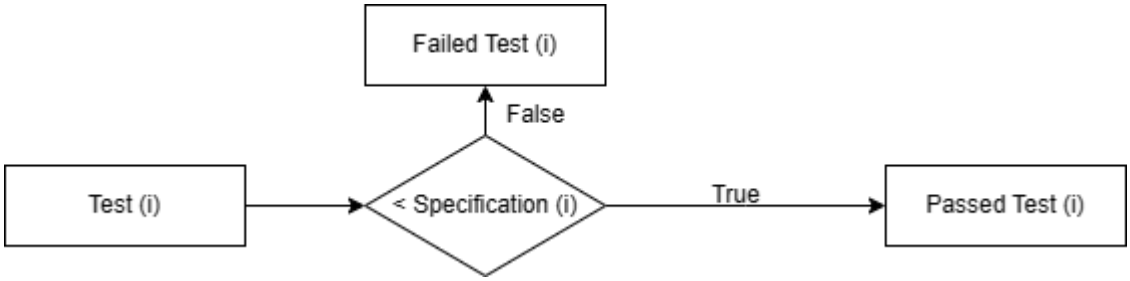
2. How is a knowledgebase used in an expert system?
   Expert Systems: Expert systems are AI programs designed to mimic the decision-making abilities of human experts in specific domains. They use knowledge, rules, and reasoning mechanisms to provide advice, solve problems, or make recommendations within their specialized areas.

3. What are limitations of Expert systems?
   - Narrow Focus: Expert systems are confined to the domain they are designed for and lack generalization to other areas.
   - Knowledge Acquisition: Gathering accurate and comprehensive expert knowledge can be time-consuming and challenging.
   - Maintenance: Regular updates are necessary to keep the knowledge base up-to-date, which can be resource-intensive.
   - Lack of Common Sense: Expert systems may struggle with understanding context and common-sense reasoning.
   - Scalability: Adapting and scaling expert systems to handle a wide range of complex scenarios can be difficult.
   - Ethical Concerns: The decisions made by expert systems can have ethical implications, and ensuring fairness and accountability is a challenge.
   - Limited Learning: Expert systems often lack the ability to learn from new data or adapt to evolving situations.
   - High Development Costs: Building, maintaining, and fine-tuning expert systems can be expensive and time-consuming.
   - Human Dependency: Overreliance on expert systems might reduce human decision-making skills and critical thinking.

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

Flowchart



Code:

Experiment 9

Problem Statement:

A manufacturing company is trying to improve their quality control process. They have a dataset of historical quality data, which includes the results of quality tests, as well as the specifications for each product. Implement a Simple Expert system to identify potential quality problems.

Product ID	Test 1 Result	Test 2 Result	Test 3 Result	Specification 1	Specification 2	Specification 3
1	95	20	8	100	25	10
2	98	22	9	100	20	10
3	93	18	7	95	15	8
4	100	24	10	100	25	12
5	96	21	8	98	20	10
6	92	19	6	95	15	8
7	90	17	5	95	15	8

Manual Rules:

Rule for Test 1:

- If the 'Test 1 Result' is less than 'Specification 1', flag as 'Test 1 Issue'.

Rule for Test 2:

- If the 'Test 2 Result' is less than 'Specification 2', flag as 'Test 2 Issue'.

Rule for Test 3:

- If the 'Test 3 Result' is less than 'Specification 3', flag as 'Test 3 Issue'.

Code:

[26]
```python
# importing required libraries
import numpy as np
import pandas as pd
```
Python

Load the dataset into a DataFrame

```python
# Load the dataset into a DataFrame
def load_dataset():
    # TODO: Load the dataset from a CSV file or define it here as a dictionary and create a DataFrame.
    data = {
        'Product ID': [1, 2, 3, 4, 5, 6, 7],
        'Test 1 Result': [95, 98, 93, 100, 96, 92, 90],
        'Test 2 Result': [20, 22, 18, 24, 21, 19, 17],
        'Test 3 Result': [8, 9, 7, 10, 8, 6, 5],
        'Specification 1': [100, 100, 95, 100, 98, 95, 95],
        'Specification 2': [25, 20, 15, 25, 20, 15, 15],
        'Specification 3': [10, 10, 8, 12, 10, 8, 8]
    }

```

```
    df = pd.DataFrame(data)
    return df

```

[27] Python

Identify Quality problems for a single product

```
# Function to identify quality problems for a single product
def identify_quality_problems_for_product(product_id, test_results, specifications):
    problems = []
    for i in range(3):
        if test_results[i] < specifications[i]:
            problems.append(f"Product {product_id} failed Test {i+1}. Expected: {specifications[i]}, Got: {test_results[i]}")
    return problems
```

[28] Python

Identify Quality problems for entire dataset

```
# Function to identify quality problems in the entire dataset
def identify_quality_problems(df):
    problems = []
    for index, row in df.iterrows():
        product_id = row['Product ID']
        test_results = [row['Test 1 Result'], row['Test 2 Result'], row['Test 3 Result']]
        specifications = [row['Specification 1'], row['Specification 2'], row['Specification 3']]

        product_problems = identify_quality_problems_for_product(product_id, test_results, specifications)
        problems.extend(product_problems)
    return problems
```

[29] Python

Main function to solve the problem

```
# Main function to run the expert system
def main():
    df = load_dataset()
    quality_problems = identify_quality_problems(df)

    if len(quality_problems) == 0:
        print("No quality problems detected.")
    else:
        print("Quality problems detected:")
        for problem in quality_problems:
            print(problem)

if __name__ == "__main__":
    main()
```

[30] Python

```
Quality problems detected:
Product 1 failed Test 1. Expected: 100, Got: 95
Product 1 failed Test 2. Expected: 25, Got: 20
Product 1 failed Test 3. Expected: 10, Got: 8
Product 2 failed Test 1. Expected: 100, Got: 98
Product 2 failed Test 3. Expected: 10, Got: 9
Product 3 failed Test 1. Expected: 95, Got: 93
Product 3 failed Test 3. Expected: 8, Got: 7
Product 4 failed Test 2. Expected: 25, Got: 24
Product 4 failed Test 3. Expected: 12, Got: 10
Product 5 failed Test 1. Expected: 98, Got: 96
Product 5 failed Test 3. Expected: 10, Got: 8
Product 6 failed Test 1. Expected: 95, Got: 92
Product 6 failed Test 3. Expected: 8, Got: 6
Product 7 failed Test 1. Expected: 95, Got: 90
Product 7 failed Test 3. Expected: 8, Got: 5
```