EXPERIMENT NO. 10

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: 11.11.2023
Faculty Signature:
Grade:

Objective(s):

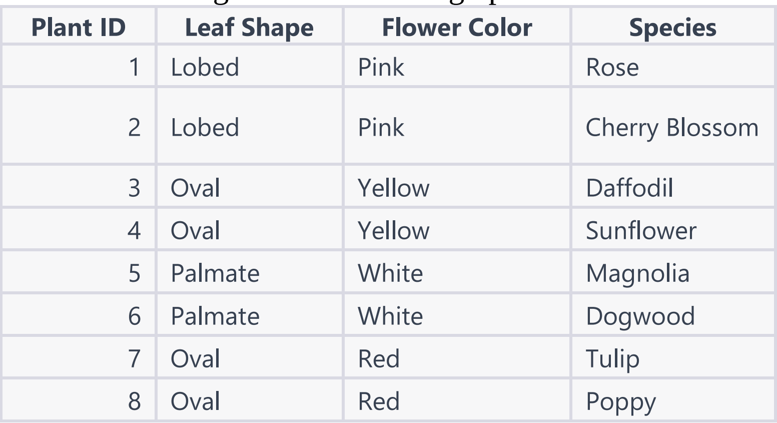
- Understand the widely known MYCIN and DART expert based systems.
- Implement an expert-based systems inspired by MYCIN and DART.

Outcome:

Students will be familiarized with basics of Fuzzy sets.

Problem Statement:

Create Python program for an expert-based systems inspired by MYCIN and DART for plant identification. The systems will take input from the user regarding observable characteristics of a plant and use a set of rules and certainty factors to identify the most likely plant species.
Use the following dataset for setting up the rules:



Plant ID	Leaf Shape	Flower Color	Species
1	Lobed	Pink	Rose
2	Lobed	Pink	Cherry Blossom
3	Oval	Yellow	Daffodil
4	Oval	Yellow	Sunflower
5	Palmate	White	Magnolia
6	Palmate	White	Dogwood
7	Oval	Red	Tulip
8	Oval	Red	Poppy

Background Study:

MYCIN, a medical diagnosis system, utilized rule-based reasoning with certainty factors to diagnose infectious diseases and suggest treatments based on patient symptoms. Meanwhile, DART excelled in diagnosing complex electronic circuit faults using model-based reasoning and causal analysis. These pioneering systems demonstrated the effectiveness of knowledge-based approaches in real-world problem-solving, setting the foundation for the development of the expert-based system aimed at identifying potential quality problems in the manufacturing company's quality control process for plant identification.

Question Bank:

1. How does MYCIN and DART based Expert system work?
   - **MYCIN:** MYCIN was an early expert system used for medical diagnosis of infectious diseases. It worked by collecting patient symptoms and medical data, applying a set of

rules based on expert knowledge in the medical field, and then providing a diagnosis and treatment recommendation.

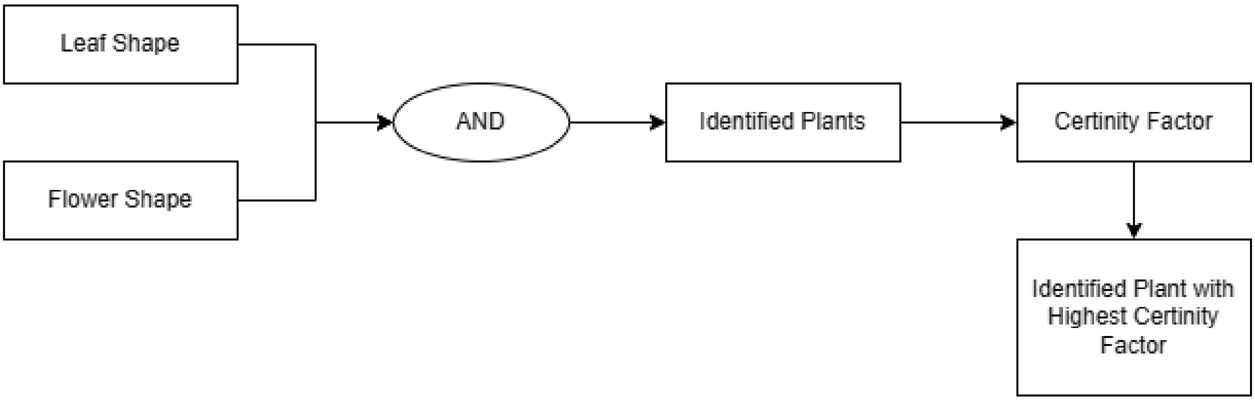
- **DART:** DART (Diagnostic and Reasoning Tool) is another medical expert system that focuses on diagnosing diseases based on clinical symptoms and laboratory results. It uses an inference engine to match symptoms with known patterns and suggest possible diagnoses.

2. What are some of the challenges faced while building such systems?
    - Knowledge Acquisition: Extracting expert knowledge and translating it into a machine-readable format can be difficult and time-consuming.
    - Knowledge Representation: Choosing the right representation for knowledge that captures both declarative and procedural aspects.
    - Rule Management: Handling large rule sets and ensuring consistency, accuracy, and maintainability.
    - Inference Mechanisms: Designing efficient and accurate inference engines for drawing conclusions from knowledge.
    - Domain Limitations: Expert systems are often limited to specific domains and may struggle with handling unfamiliar situations.
    - Common Sense Reasoning: Infusing systems with human-like common-sense reasoning remains a challenge.
    - User Interaction: Designing intuitive and effective user interfaces and explanation mechanisms.
    - Continuous Learning: Ensuring expert systems can adapt and learn from new data and experiences.
    - Ethical Considerations: Addressing ethical concerns, accountability, and biases in decision-making.

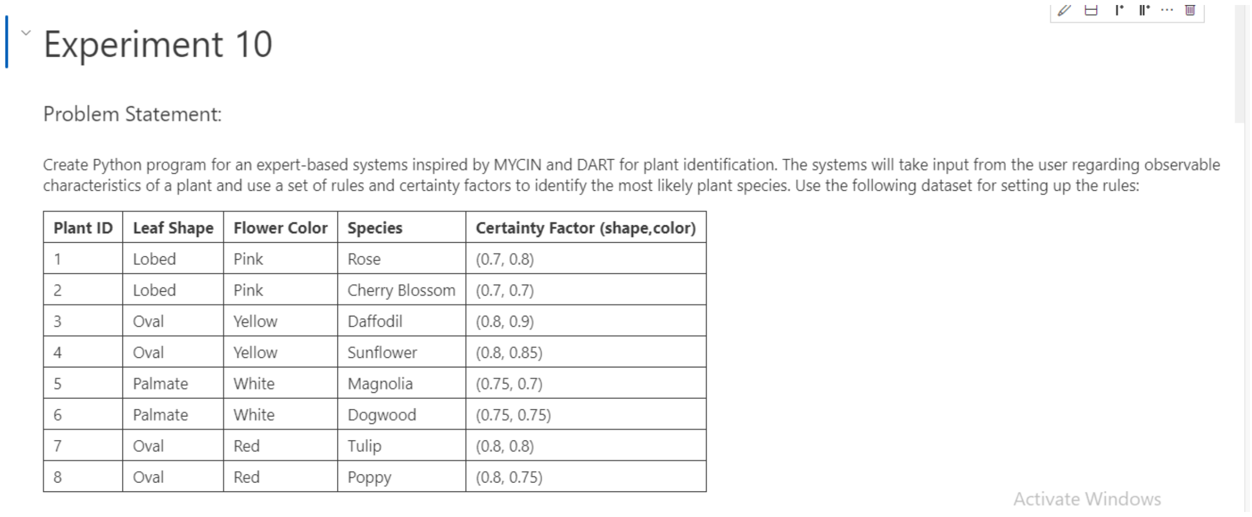
Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

Flowchart



Code:

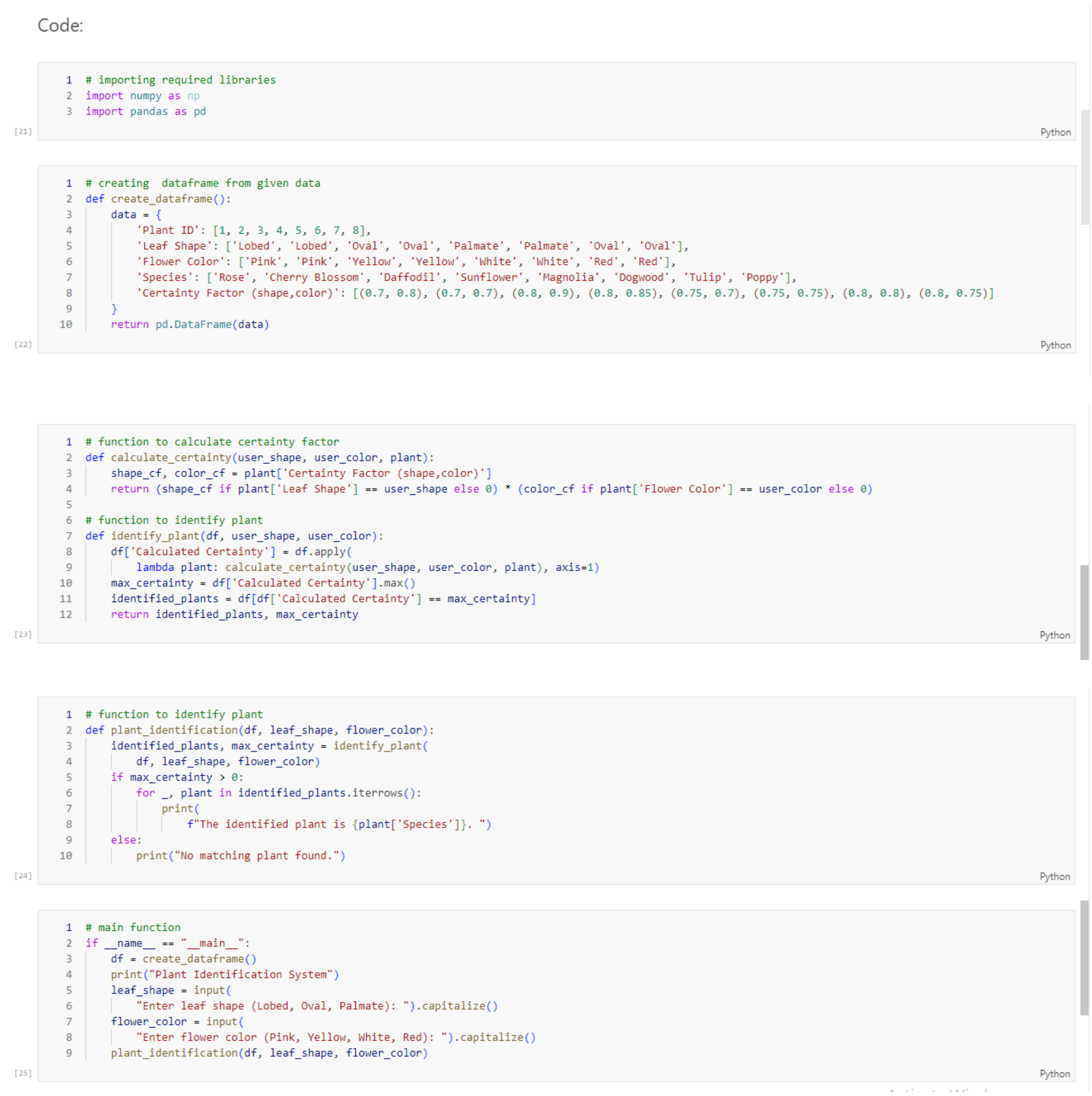


Experiment 10

Problem Statement:

Create Python program for an expert-based systems inspired by MYCIN and DART for plant identification. The systems will take input from the user regarding observable characteristics of a plant and use a set of rules and certainty factors to identify the most likely plant species. Use the following dataset for setting up the rules:

Plant ID	Leaf Shape	Flower Color	Species	Certainty Factor (shape,color)
1	Lobed	Pink	Rose	(0.7, 0.8)
2	Lobed	Pink	Cherry Blossom	(0.7, 0.7)
3	Oval	Yellow	Daffodil	(0.8, 0.9)
4	Oval	Yellow	Sunflower	(0.8, 0.85)
5	Palmate	White	Magnolia	(0.75, 0.7)
6	Palmate	White	Dogwood	(0.75, 0.75)
7	Oval	Red	Tulip	(0.8, 0.8)
8	Oval	Red	Poppy	(0.8, 0.75)



Code:

```python
# importing required libraries
import numpy as np
import pandas as pd
```

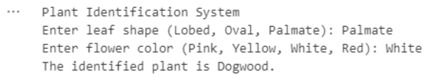
```python
# creating  dataframe from given data
def create_dataframe():
    data = {
        'Plant ID': [1, 2, 3, 4, 5, 6, 7, 8],
        'Leaf Shape': ['Lobed', 'Lobed', 'Oval', 'Oval', 'Palmate', 'Palmate', 'Oval', 'Oval'],
        'Flower Color': ['Pink', 'Pink', 'Yellow', 'Yellow', 'White', 'White', 'Red', 'Red'],
        'Species': ['Rose', 'Cherry Blossom', 'Daffodil', 'Sunflower', 'Magnolia', 'Dogwood', 'Tulip', 'Poppy'],
        'Certainty Factor (shape,color)': [(0.7, 0.8), (0.7, 0.7), (0.8, 0.9), (0.8, 0.85), (0.75, 0.7), (0.75, 0.75), (0.8, 0.8), (0.8, 0.75)]
    }
    return pd.DataFrame(data)
```

```python
# function to calculate certainty factor
def calculate_certainty(user_shape, user_color, plant):
    shape_cf, color_cf = plant['Certainty Factor (shape,color)']
    return (shape_cf if plant['Leaf Shape'] == user_shape else 0) * (color_cf if plant['Flower Color'] == user_color else 0)

# function to identify plant
def identify_plant(df, user_shape, user_color):
    df['Calculated Certainty'] = df.apply(
        lambda plant: calculate_certainty(user_shape, user_color, plant), axis=1)
    max_certainty = df['Calculated Certainty'].max()
    identified_plants = df[df['Calculated Certainty'] == max_certainty]
    return identified_plants, max_certainty
```

```python
# function to identify plant
def plant_identification(df, leaf_shape, flower_color):
    identified_plants, max_certainty = identify_plant(
        df, leaf_shape, flower_color)
    if max_certainty > 0:
        for _, plant in identified_plants.iterrows():
            print(
                f"The identified plant is {plant['Species']}. ")
    else:
        print("No matching plant found.")
```

```python
# main function
if __name__ == "__main__":
    df = create_dataframe()
    print("Plant Identification System")
    leaf_shape = input(
        "Enter leaf shape (Lobed, Oval, Palmate): ").capitalize()
    flower_color = input(
        "Enter flower color (Pink, Yellow, White, Red): ").capitalize()
    plant_identification(df, leaf_shape, flower_color)
```



```
Plant Identification System
Enter leaf shape (Lobed, Oval, Palmate): Palmate
Enter flower color (Pink, Yellow, White, Red): White
The identified plant is Dogwood.
```