Exercise No. 7			
Topic:	**Module 3.0: Dynamic Systems and Markov Models**	**Week No.**	13-14
Course Code:	**CSST101**	**Term:**	1st Semester
Course Title:	**Advance Representation and Reasoning**	**Academic Year:**	2024-2025
Student Name		**Section**	
Due date		**Points**	

Topic 3.2: Advanced Probabilistic Models

Markov decision processes

Exercise 1: Setting Up the Environment

Objective: Prepare your Python environment and install the required libraries.

Instructions:

1. Open your preferred Python IDE or Jupyter Notebook.
2. Install the required libraries:

```
pip install pgmpy numpy pandas matplotlib
```

3. Create a Python script or notebook for this lab.

Exercise 2: Building a Simple DBN

Objective: Create a Dynamic Bayesian Network (DBN) to model sequential data.

Instructions:

1. Define the variables and dependencies for the DBN.
    - Example: Rain and Sprinkler variables.
    - Create intra-time dependencies and inter-time dependencies.
2. Define the Conditional Probability Tables (CPTs) for each variable.
3. Add the CPTs to the DBN model using the pgmpy library.

Example Code Snippet:

```
from pgmpy.models import DynamicBayesianNetwork as DBN
from pgmpy.factors.discrete import TabularCPD

model = DBN()
model.add_edges_from([(('Rain', 0), ('Sprinkler', 0)), (('Rain', 0), ('Rain', 1))])

model.add_cpds(
    TabularCPD(('Rain', 0), 2, [[0.8], [0.2]]),
    TabularCPD(('Sprinkler', 0), 2, [[0.6, 0.2], [0.4, 0.8]], evidence=[('Rain', 0)], evidence_card=[2]),
    TabularCPD(('Rain', 1), 2, [[0.7, 0.3], [0.3, 0.7]], evidence=[('Rain', 0)], evidence_card=[2])
)
```

Exercise 3: Performing Inference

Objective: Query the DBN model to predict the probabilities of future states.

Instructions:

1. Use the DBNInference module to create an inference object.
2. Query the DBN for probabilities of future states given initial evidence.
3. Print the inference results.

Example Code Snippet:

```
from pgmpy.inference import DBNInference

dbn_infer = DBNInference(model)
result = dbn_infer.query(variables=[('Rain', 1)], evidence={('Rain', 0): 1})
print(result)
```

Exercise 4: Visualizing the DBN

Objective: Create a visual representation of the DBN structure.

Instructions:

1. Use matplotlib or networkx to visualize the nodes and edges.
2. Add labels to represent states and transitions.

Example Code Snippet:

```
import networkx as nx
import matplotlib.pyplot as plt

graph = model.to_networkx()
nx.draw(graph, with_labels=True, node_size=3000, node_color='lightblue', font_size=10)
plt.show()
```

Exercise 5: Extending the DBN

Objective: Modify the DBN to include additional variables, such as WetGrass.

Instructions:

1. Add new nodes and dependencies to the DBN.
2. Define the corresponding CPTs and add them to the model.
3. Perform inference on the extended DBN and analyze the results.

Submission Format

1. Submit your Python script or Jupyter Notebook in your GitHub repository.
2. Organize the repository with labeled folders:
    - scripts/
    - notebooks/
    - README.md

Filename Format:
[SECTION]-[SURNAME]-EXER6 (e.g., 3A-BERNARDINO-EXER7).

Rubric for Exercise on DBNs Using Python

Criteria	Excellent (4 pts)	Good (3 pts)	Fair (2 pts)	Poor (1 pt)
Environment Setup	Fully functional, no errors.	Minor issues.	Partial functionality.	Not functional.
DBN Construction	Complete, accurate model.	Minor inaccuracies.	Major errors in setup.	Not implemented.
Inference	Accurate results, well-documented.	Mostly correct.	Significant errors.	Not functional.
Visualization	Clear, well-labeled.	Mostly clear.	Poorly labeled.	No visualization.
Documentation and Clarity	Comprehensive and easy to follow.	Partially documented.	Minimal documentation.	No documentation.