Machine Problem No. 4			
Topic:	**Topic 2.2: Bayesian Networks**	**Week No.**	7-9
Course Code:	**CSST101**	**Term:**	1st Semester
Course Title:	**Advance Representation and Reasoning**	**Academic Year:**	2024-2025
Student Name		**Section**	
Due date		**Points**	

Machine Problem: Bayesian Network for Healthcare Diagnosis

Overview:

In this task, you will develop a Bayesian Network model to diagnose a common health condition using Python and the pgmpy library. The goal is to understand how probabilistic reasoning can aid in medical diagnosis by creating a network that models relationships between symptoms, conditions, and diagnostic results.

Healthcare Scenario:

Consider a diagnostic model for a condition like **heart disease**. The Bayesian Network will include various risk factors and symptoms. Your network will use these factors to infer the likelihood of heart disease in a patient.

Instructions:

Exercise 1: Setting Up the Environment

1. **Install Libraries:**
   - Use Google Colab or a local environment and install pgmpy using !pip install pgmpy.

2. **Import Libraries:**
   - Import the necessary libraries:
     - pgmpy for the Bayesian Network.
     - pandas for data handling.
     - networkx for visualization.
     - matplotlib for plotting graphs.

Exercise 2: Define the Bayesian Network Structure

1. **Define the Variables:**
   - Define the following variables in the Bayesian Network:
     - **Age:** (Young, Middle-aged, Old)
     - **Smoking:** (Yes, No)
     - **Exercise:** (Regular, None)
     - **Cholesterol:** (High, Normal)
     - **Blood Pressure:** (High, Normal)
     - **Heart Disease:** (Yes, No)
     - **Chest Pain:** (Yes, No)
     - **ECG Result:** (Abnormal, Normal)

2. **Define the Relationships:**
   - Model the following dependencies:
     - Heart Disease depends on Age, Smoking, Exercise, Cholesterol, and Blood Pressure.
     - Chest Pain depends on Heart Disease.
     - ECG Result depends on Heart Disease.

Exercise 3: Define Conditional Probability Tables (CPTs)

1. **Create CPTs Using pgmpy:**
   - Use TabularCPD from pgmpy to define the CPTs for each variable.
   - Use hypothetical probability values to represent the relationships. For example:
     - Probability of Heart Disease given different combinations of Age, Smoking, Exercise, Cholesterol, and Blood Pressure.
     - Probability of Chest Pain given the presence or absence of Heart Disease.
     - Probability of an ECG Result being abnormal given the presence or absence of Heart Disease.

Exercise 4: Inference in the Bayesian Network

1. **Perform Probabilistic Inference:**
   - Use the variable elimination method to perform the following queries:
     - Determine the probability of having Heart Disease given the patient is Middle-aged, a Smoker, has High Cholesterol, and High Blood Pressure.
     - Calculate the probability of an Abnormal ECG Result given that the patient is experiencing Chest Pain.

2. **Posterior Probability Analysis:**
   - Find the probability distribution of Heart Disease in patients who do not exercise.

Exercise 5: Parameter Learning from Simulated Data

1. **Simulate a Healthcare Dataset:**
   - Create a synthetic dataset with 1,000 patient observations, including information about Age, Smoking, Exercise, Cholesterol, Blood Pressure, Heart Disease, Chest Pain, and ECG Result.
   - Ensure the dataset reflects the dependencies in the network.

2. **Estimate Parameters:**
   - Use the synthetic dataset to estimate the Conditional Probability Tables (CPTs) for Heart Disease, Chest Pain, and ECG Result nodes using Maximum Likelihood Estimation.
   - Compare the learned parameters with the initial CPTs.

Exercise 6: Network Visualization and Analysis

1. **Visualize the Network:**
   - Use networkx to draw the Bayesian Network. Label nodes and edges to represent the relationships clearly.

2. **Sensitivity Analysis:**
   - Analyze how changes in the probability of Smoking affect the likelihood of Heart Disease.

- Plot the results using matplotlib to show the sensitivity of the model to changes in risk factors.

This machine problem encourages the development of an interactive Python program that helps users understand probability theory and decision-making under uncertainty in AI contexts. Through coding, simulations, and real-time feedback, you will demonstrate both technical and conceptual mastery.

Repository Organization:

Folders: Organize your repository with folders such as scripts/, colab_notebooks/, README.md.
Labels: Label files appropriately and maintain clear documentation.

Submission Requirements:

- Organize your code into a well-structured Python script or Jupyter Notebook.
- Document the logic and steps using comments in the code and save it in PDF File.
- Upload the files to a GitHub repository with clear labels and a README.md file explaining your implementation.
- Use the filename format: [SECTION]-[SURNAME]-HEALTH-MP4. For example: 3A-BERNARDINO-HEALTH-MP4.

Rubric for Bayesian Network Healthcare Diagnosis

Criteria	Excellent (4 Points)	Good (3 Points)	Fair (2 Points)	Poor (1 Point)
Exercise 1: Environment Setup	- Correctly installs pgmpy and imports all necessary libraries without errors. - Code is clean and well-commented.	- Minor issues in library installation or imports, but the main functionality is intact. - Code is mostly clean with some comments.	- Libraries are partially installed or imported, causing some functionality issues. - Limited documentation in code.	- Libraries are not installed or imported correctly, leading to multiple errors. - No documentation in code.
Exercise 2: Define Network Structure	- The Bayesian Network structure is defined accurately, reflecting all specified relationships. - Variables and dependencies are correctly modeled. - Clear documentation of logic.	- The network structure is mostly correct; minor errors in relationships or missing a variable. - Code has some comments.	- Significant errors in network structure, with major relationships incorrectly defined. - Code lacks clarity and proper documentation.	- Network structure is not defined, or critical errors are present. - No documentation or explanation provided.
Exercise 3: Define CPTs	- All CPTs are correctly implemented using realistic probability values. - Code is clean and easy to understand.	- CPTs are mostly correct; minor inaccuracies in probability values. - Code is somewhat clear but may lack some comments.	- Significant errors in defining CPTs, with several incorrect probability values. - Code is difficult to understand due to lack of documentation.	- CPTs are not implemented, or critical errors make them unusable. - Code is unclear and not documented.
Exercise 4: Inference	- Performs inference accurately using variable elimination. - Results are correctly calculated and printed. - Code is well-organized and thoroughly commented.	- Inference is mostly correct but contains minor errors in query formulation or interpretation. - Code is partially documented.	- Attempts inference but with significant errors leading to incorrect results. - Code lacks clarity and has minimal comments.	- Inference is not performed, or results are completely incorrect. - No documentation provided.
Exercise 5: Parameter Learning	- Generates a synthetic dataset accurately reflecting the network structure. - Estimates CPTs	- Dataset is mostly generated correctly; minor issues affect parameter estimation. - Some comparison	- Significant issues in dataset generation; affects parameter estimation accuracy. - Minimal or unclear	- Dataset is not generated, and parameter estimation is absent or incorrect.

Criteria	Excellent (4 Points)	Good (3 Points)	Fair (2 Points)	Poor (1 Point)
	correctly using Maximum Likelihood Estimation. - Comparison with initial CPTs is documented.	with initial CPTs is provided.	comparison with initial CPTs.	- No comparison with initial CPTs.
Exercise 6: Visualization & Analysis	- Network structure is visualized clearly using networkx. - Sensitivity analysis is thorough, and plots are well-presented with labels.	- Visualization is mostly clear; minor issues in labeling or presentation. - Basic sensitivity analysis provided.	- Visualization is unclear or poorly presented, making the network hard to understand. - Sensitivity analysis is attempted but incomplete.	- Visualization is missing or entirely incorrect. - No sensitivity analysis provided.
Documentation	- Code is well-documented, with comments explaining each step and logic throughout the exercises. - Demonstrates clear understanding of Bayesian Networks.	- Code has some documentation, but some parts lack clarity or explanation.	- Minimal documentation, making it difficult to follow the logic of the code.	- No documentation provided; code is unclear and hard to understand.
Creativity and Insight	- Demonstrates creativity by adding additional insights or exploring alternative scenarios. - Goes beyond the given requirements to enhance the network model.	- Shows some creativity in implementation but mostly follows the given instructions.	- Limited creativity; implements basic requirements without exploring further.	- No evidence of creativity; strictly follows the instructions without any additional insights.