EXPERIMENT NO. 8

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: 30.09.2023
Faculty Signature:
Grade:

Objective(s):

- Understand and study Bayesian Network.
- Implement Bayesian Network for solving a real-world graph-based problem.

Outcome:

Students will be familiarized with concepts of Bayesian Network.

Problem Statement:

Using Bayesian network, implement a Python program to calculate probability in disease diagnosis.

A medical clinic is trying to determine the probability that a patient has a certain disease. They have a test that can be used to diagnose the disease, but the test is not always accurate. The clinic knows that the probability of a false positive is 1%, and the probability of a false negative is 5%. The clinic has a patient who has tested positive for the disease.
They want to use Bayesian Theorem to calculate the probability that the patient actually has the disease. The Bayesian network will consist of two nodes: 'D' (for disease) and 'T' (for test result). conditional probability distributions (CPDs) for the nodes will be based on the provided information

Background Study:

Bayesian networks, also known as belief networks or probabilistic graphical models, are powerful tools for representing and reasoning about uncertainty in probabilistic systems. They are widely used in various fields, including medical diagnosis, natural language processing, finance, and more. A Bayesian network is a directed acyclic graph (DAG) in which nodes represent random variables, and edges represent probabilistic dependencies between the variables.

Question Bank:

1. What is a Bayesian Theorem?
   The Bayesian Theorem, also known as Bayes' Theorem, is a fundamental concept in probability theory and statistics. It describes the probability of an event occurring based on prior knowledge or information. The theorem mathematically relates the conditional probability of an event A given event B, with the conditional probability of event B given event A, along with the probabilities of events A and B independently.

2. Discuss the applications of Bayesian Theorem?

- Medical Diagnosis: Bayesian methods help in disease diagnosis by incorporating prior probabilities and test results to estimate the probability of a patient having a particular condition.
- Spam Filtering: Bayesian algorithms are used in spam email filters, considering word probabilities to determine the likelihood of an email being spam.
- Machine Learning: Bayesian networks assist in modeling dependencies between variables, aiding in probabilistic reasoning and prediction tasks.
- Natural Language Processing: Bayesian methods are employed in language models, part-of-speech tagging, and sentiment analysis.
- Recommendation Systems: Bayesian techniques help in collaborative filtering and content-based recommendation to suggest relevant products or content.
- Financial Analysis: Bayesian inference is used in risk assessment, portfolio optimization, and credit scoring.
- Weather Forecasting: Bayesian models can integrate historical data and current observations to improve the accuracy of weather predictions.
- Fault Diagnosis: In engineering, Bayesian networks aid in diagnosing faults in complex systems by analyzing sensor data and known relationships.
- Image Processing: Bayesian techniques enhance image denoising, object recognition, and image segmentation.

Bayesian methods are versatile and applicable in various fields where uncertainty and probabilistic reasoning are involved.

Student Work Area

Algorithm/Flowchart/Code/Sample Outputs

Experiment 8

+ Code + Markdown

Problem Statement:

Using Bayesian network, implement a Python program to calculate probability in disease diagnosis.

A medical clinic is trying to determine the probability that a patient has a certain disease. The test which is used to diagnose the disease is not always accurate. The clinic knows that the probability of a false positive is 1%, and the probability of a false negative is 5%. The clinic has a patient who has tested positive for the disease.

Knowing that disease occurs in a patient 25% times. They want to use Bayesian Theorem to calculate the probability that the patient actually has the disease. The Bayesian network will consist of two nodes: 'D' (for disease) and 'T' (for test result). conditional probability distributions (CPDs) for the nodes will be based on the provided information

Installing Required Libraries

```
[1] ✓ 20.8s                                                                    Python
1 %pip install pgmpy
...
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from pandas->pgmpy) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from pandas->pgmpy) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from pandas->pgmpy) (2023.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from scikit-learn->pgmpy) (3.2.0)
Collecting patsy>=0.5.2 (from statsmodels->pgmpy)
  Downloading patsy-0.5.3-py2.py3-none-any.whl (233 kB)
     ---------------------------------------- 0.0/233.8 kB ? eta -:--:--
     ---------------------------------------- 233.8/233.8 kB 7.2 MB/s eta 0:00:00
Requirement already satisfied: packaging>=21.3 in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from statsmodels->pgmpy) (23.2)
Requirement already satisfied: filelock in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from torch->pgmpy) (3.12.4)
Requirement already satisfied: typing-extensions in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from torch->pgmpy) (4.8.0)
Requirement already satisfied: sympy in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from torch->pgmpy) (1.12)
Requirement already satisfied: jinja2 in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from torch->pgmpy) (3.1.2)
Requirement already satisfied: fsspec in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from torch->pgmpy) (2023.10.0)
Requirement already satisfied: colorama in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from tqdm->pgmpy) (0.4.6)
Requirement already satisfied: six in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from patsy>=0.5.2->statsmodels->pgmpy) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from jinja2->torch->pgmpy) (2.1.3)
Requirement already satisfied: mpmath>=0.19 in c:\users\mainp\appdata\local\programs\python\python310\lib\site-packages (from sympy->torch->pgmpy) (1.3.0)
Downloading pgmpy-0.1.24-py3-none-any.whl (2.0 MB)
   ---------------------------------------- 0.0/2.0 MB ? eta -:--:--
   ---------- ----------------------------- 0.7/2.0 MB 11.5 MB/s eta 0:00:01
   ------------------------- -------------- 1.3/2.0 MB 11.8 MB/s eta 0:00:01
   ---------------------------------- ------ 1.7/2.0 MB 10.6 MB/s eta 0:00:01
   ---------------------------------------- 2.0/2.0 MB 10.4 MB/s eta 0:00:00
Downloading tqdm-4.66.1-py3-none-any.whl (78 kB)
   ---------------------------------------- 0.0/78.3 kB ? eta -:--:--
```

Code:

```
[2] ✓ 1m 41.7s                                                                 Python
1 # importing required libraries
2 from pgmpy.models import BayesianNetwork
3 from pgmpy.factors.discrete import TabularCPD
4 from pgmpy.inference import VariableElimination
...
c:\Users\mainp\AppData\Local\Programs\Python\Python310\lib\site-packages\tqdm\auto.py:21: TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See https://ipywidget
  from .autonotebook import tqdm as notebook_tqdm
```

Bayesian Network Creation using BayesianNetwork from pgmpy AND Defining CPDs using Tabular CPDS

```
1  def create_bayesian_network():
2      # TODO: Create a Bayesian Network
3      model = BayesianNetwork([('D', 'T')])
4
5      # TODO: Define conditional probability distributions (CPDs)
6      # CPD for 'D' (Disease)
7      cpd_disease = TabularCPD(
8          variable='D', variable_card=2, values=[[0.75], [0.25]])
9      # CPD for 'T' (Test Result)
10
11     cpd_test = TabularCPD(variable='T', variable_card=2,
12                           values=[[0.99, 0.05],
13                                   [0.01, 0.95]],
14                           evidence=['D'], evidence_card=[2])
15
```

```python
    # TODO: Add CPDs to the model
    # model.add_cpds(cpd_disease, cpd_test)
    model.add_cpds(cpd_disease, cpd_test)

    # TODO: Check if the model is valid
    # assert model.check_model()
    assert model.check_model()

    # TODO: Return the created model
    # return model
    return model
```

[3] ✓ 0.0s Python

Calculating Disease Bayesian Probablity

```python
def calculate_probability_of_disease(model, test_result):
    # TODO: Create an inference object
    inference = VariableElimination(model)

    # TODO: Given the test result, calculate the probability of having the disease
    probabilities = inference.query(variables=['D'], evidence={'T': test_result})

    # TODO: Return the calculated probability
    return probabilities.values[1]  # 1 for positive disease

```

[4] ✓ 0.0s Python

Main function to solve the problem

```python
if __name__ == "__main__":
    bayesian_network = create_bayesian_network()
    test_result = 1  # 1 for positive test result, 0 for negative

    probability = calculate_probability_of_disease(bayesian_network, test_result)
    print(f"Probability of having the disease given a positive test result: {probability:.4f}")
```

[5] ✓ 0.0s Python

```
Probability of having the disease given a positive test result: 0.9694
```