Experiment No : 2 (AAI)

**Code :**

from pgmpy.models import BayesianModel

from pgmpy.inference import VariableElimination

from pgmpy.factors.discrete import TabularCPD

model = BayesianModel([

('Difficulty', 'Grade'),

('Intelligence', 'Grade'),

('Grade', 'Study')

])

cpd\_difficulty = TabularCPD(variable='Difficulty', variable\_card=2, values=[[0.6], [0.4]])

cpd\_intelligence = TabularCPD(variable='Intelligence', variable\_card=2, values=[[0.7], [0.3]])

cpd\_grade = TabularCPD(

variable='Grade', variable\_card=3,

values=[

[0.3, 0.05, 0.9, 0.5],

[0.4, 0.25, 0.08, 0.3],

[0.3, 0.7, 0.02, 0.2]

],

evidence=['Difficulty', 'Intelligence'],

evidence\_card=[2, 2]

)

cpd\_study = TabularCPD(

variable='Study', variable\_card=2,

values=[

[0.2, 0.8, 0.9],

[0.8, 0.2, 0.1]

],

evidence=['Grade'],

evidence\_card=[3]

)

model.add\_cpds(cpd\_difficulty, cpd\_intelligence, cpd\_grade, cpd\_study)

assert model.check\_model()

inference = VariableElimination(model)

query\_result\_study = inference.query(variables=['Study'], evidence={'Difficulty': 0, 'Intelligence': 1})

query\_result\_grade = inference.query(variables=['Grade'], evidence={'Difficulty': 0, 'Intelligence': 1})

query\_result\_intelligence = inference.query(variables=['Intelligence'], evidence={'Difficulty': 0})

def format\_table(query\_result, variable\_name):

table = []

table.append(f"+--------------+--------------+")

table.append(f"| {variable\_name:<12} | phi({variable\_name}) |")

table.append(f"+==============+==============+")

for i, val in enumerate(query\_result.values):

table.append(f"| {variable\_name}({i}) | {val:.4f} |")

table.append(f"+--------------+--------------+")

return table

table\_study = format\_table(query\_result\_study, "Study")

table\_grade = format\_table(query\_result\_grade, "Grade")

table\_intelligence = format\_table(query\_result\_intelligence, "Intelligence")

max\_rows = max(len(table\_study), len(table\_grade), len(table\_intelligence))

while len(table\_study) < max\_rows:

table\_study.append(" " \* 30)

while len(table\_grade) < max\_rows:

table\_grade.append(" " \* 30)

while len(table\_intelligence) < max\_rows:

table\_intelligence.append(" " \* 30)

for row\_study, row\_grade, row\_intelligence in zip(table\_study, table\_grade, table\_intelligence):

print(f"{row\_study} {row\_grade} {row\_intelligence}")

**output:**

+--------------+--------------+ +--------------+--------------+ +---------------------+--------------------+

| Study | phi(Study) | | Grade | phi(Grade) | | Intelligence | phi(Intelligence)|

+=========+=========+ +=========+=========+ +=============+=============+

| Study(0) | 0.5200 | | Grade(0) | 0.5000 | | Intelligence(0) | 0.7000 |

+--------------+--------------+ +--------------+--------------+ +---------------------+--------------------+

| Study(1) | 0.4800 | | Grade(1) | 0.3000 | | Intelligence(1) | 0.3000 |

+--------------+--------------+ +--------------+--------------+ +---------------------+--------------------+

| Grade(2) | 0.2000 |

+---------------+--------------+

+--------------+--------------+ +--------------+--------------+ +---------------------+--------------------+

| Study | phi(Study) | | Grade | phi(Grade) | | Intelligence | phi(Intelligence)|

+=========+=========+ +=========+=========+ +=============+=============+

| Study(0) | 0.8400 | | Grade(0) | 0.0500 | | Intelligence(0) | 0.7000 |

+--------------+--------------+ +--------------+--------------+ +---------------------+--------------------+

| Study(1) | 0.1600 | | Grade(1) | 0.2500 | | Intelligence(1) | 0.3000 |

+--------------+--------------+ +--------------+--------------+ +---------------------+--------------------+

| Grade(2) | 0.7000 |

+---------------+--------------+