## Notebook Information

- **Name:** Geron Simon A. Javier
- **Y&S:** BSCS 3B IS
- **Course:** CSST 101 | Advanced Representation and Reasoning
- **Topic:** Module 2.0: Probabilistic Reasoning in AI
- **Due date:** N/A

### Exercise 1: Setting Up the Environment

[1]:

```
# Install required library
!pip install pgmpy
```

```
Collecting pgmpy
  Downloading pgmpy-0.1.26-py3-none-any.whl.metadata (9.1 kB)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-
packages (from pgmpy) (3.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/
 ↪dist-packages
(from pgmpy) (1.26.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/
 ↪dist-packages
(from pgmpy) (1.13.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-
packages (from pgmpy) (1.5.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/
 ↪dist-packages
(from pgmpy) (2.2.2)
Requirement already satisfied: pyparsing in /usr/local/lib/python3.10/dist-
packages (from pgmpy) (3.1.4)
Requirement already satisfied: torch in /usr/local/lib/python3.10/
 ↪dist-packages
(from pgmpy) (2.4.1+cu121)
Requirement already satisfied: statsmodels in /usr/local/lib/python3.10/dist-
packages (from pgmpy) (0.14.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(from pgmpy) (4.66.5)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/
 ↪dist-packages
(from pgmpy) (1.4.2)
Requirement already satisfied: opt-einsum in /usr/local/lib/python3.10/dist-
packages (from pgmpy) (3.4.0)
Requirement already satisfied: xgboost in /usr/local/lib/python3.10/dist-
packages (from pgmpy) (2.1.1)
Requirement already satisfied: google-generativeai in
/usr/local/lib/python3.10/dist-packages (from pgmpy) (0.7.2)
Requirement already satisfied: google-ai-generativelanguage==0.6.6 in
/usr/local/lib/python3.10/dist-packages (from google-generativeai->pgmpy)
(0.6.6)
Requirement already satisfied: google-api-core in
/usr/local/lib/python3.10/dist-packages (from google-generativeai->pgmpy)
```

```
(2.19.2)
Requirement already satisfied: google-api-python-client in
/usr/local/lib/python3.10/dist-packages (from google-generativeai->pgmpy)
(2.137.0)
Requirement already satisfied: google-auth>=2.15.0 in
/usr/local/lib/python3.10/dist-packages (from google-generativeai->pgmpy)
(2.27.0)
Requirement already satisfied: protobuf in /usr/local/lib/python3.10/dist-
packages (from google-generativeai->pgmpy) (3.20.3)
Requirement already satisfied: pydantic in /usr/local/lib/python3.10/dist-
packages (from google-generativeai->pgmpy) (2.9.2)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.10/dist-packages (from google-generativeai->pgmpy)
(4.12.2)
Requirement already satisfied: proto-plus<2.0.0dev,>=1.22.3 in
/usr/local/lib/python3.10/dist-packages (from google-ai-
generativelanguage==0.6.6->google-generativeai->pgmpy) (1.24.0)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas->pgmpy) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas->pgmpy) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/
 ↪dist-
packages (from pandas->pgmpy) (2024.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn->pgmpy) (3.5.0)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.10/dist-
packages (from statsmodels->pgmpy) (0.5.6)
Requirement already satisfied: packaging>=21.3 in
/usr/local/lib/python3.10/dist-packages (from statsmodels->pgmpy) (24.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from torch->pgmpy) (3.16.1)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/
 ↪dist-packages
(from torch->pgmpy) (1.13.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/
 ↪dist-packages
(from torch->pgmpy) (3.1.4)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/
 ↪dist-packages
(from torch->pgmpy) (2024.6.1)
Requirement already satisfied: nvidia-nccl-cu12 in
/usr/local/lib/python3.10/dist-packages (from xgboost->pgmpy) (2.23.4)
Requirement already satisfied: googleapis-common-protos<2.0.dev0,>=1.56.2 in
/usr/local/lib/python3.10/dist-packages (from google-api-core->google-
generativeai->pgmpy) (1.65.0)
Requirement already satisfied: requests<3.0.0.dev0,>=2.18.0 in
/usr/local/lib/python3.10/dist-packages (from google-api-core->google-
generativeai->pgmpy) (2.32.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from google-auth>=2.15.0->google-
generativeai->pgmpy) (5.5.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from google-auth>=2.15.0->google-
generativeai->pgmpy) (0.4.1)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/
 ↪dist-
packages (from google-auth>=2.15.0->google-generativeai->pgmpy) (4.9)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from patsy>=0.5.6->statsmodels->pgmpy) (1.16.0)
Requirement already satisfied: httplib2<1.dev0,>=0.19.0 in
/usr/local/lib/python3.10/dist-packages (from␣
 ↪google-api-python-client->google-
generativeai->pgmpy) (0.22.0)
Requirement already satisfied: google-auth-httplib2<1.0.0,>=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from␣
 ↪google-api-python-client->google-
generativeai->pgmpy) (0.2.0)
Requirement already satisfied: uritemplate<5,>=3.0.1 in
/usr/local/lib/python3.10/dist-packages (from␣
 ↪google-api-python-client->google-
generativeai->pgmpy) (4.1.1)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2->torch->pgmpy) (2.1.5)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.10/dist-packages (from pydantic->google-
generativeai->pgmpy) (0.7.0)
Requirement already satisfied: pydantic-core==2.23.4 in
/usr/local/lib/python3.10/dist-packages (from pydantic->google-
generativeai->pgmpy) (2.23.4)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from sympy->torch->pgmpy) (1.3.0)
Requirement already satisfied: grpcio<2.0dev,>=1.33.2 in
/usr/local/lib/python3.10/dist-packages (from google-api-core[grpc]!=2.0.*,!
 ↪=2.1
.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.
 ↪*,<3.0
.0dev,>=1.34.1->google-ai-generativelanguage==0.6.
 ↪6->google-generativeai->pgmpy)
(1.64.1)
Requirement already satisfied: grpcio-status<2.0.dev0,>=1.33.2 in
/usr/local/lib/python3.10/dist-packages (from google-api-core[grpc]!=2.0.*,!
 ↪=2.1
.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.
 ↪*,<3.0
.0dev,>=1.34.1->google-ai-generativelanguage==0.6.
 ↪6->google-generativeai->pgmpy)
(1.48.2)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in
/usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-
auth>=2.15.0->google-generativeai->pgmpy) (0.6.1)
```

```
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from
requests<3.0.0.dev0,>=2.18.0->google-api-core->google-generativeai->pgmpy)
(3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests<3.0.0.dev0,>=2.18.0->google-api-core->google-
generativeai->pgmpy) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from
requests<3.0.0.dev0,>=2.18.0->google-api-core->google-generativeai->pgmpy)
(2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from
requests<3.0.0.dev0,>=2.18.0->google-api-core->google-generativeai->pgmpy)
(2024.8.30)
Downloading pgmpy-0.1.26-py3-none-any.whl (2.0 MB)
   ------------------------------ 2.0/2.0 MB
14.2 MB/s eta 0:00:00
Installing collected packages: pgmpy
Successfully installed pgmpy-0.1.26
```

[2]:

```python
# Import necessary libraries
import pandas as pd
import numpy as np
from pgmpy.models import BayesianNetwork
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination
from pgmpy.estimators import MaximumLikelihoodEstimator
import networkx as nx
import matplotlib.pyplot as plt
```

### Exercise 2: Building a Simple Bayesian Network

[3]:

```python
# Define the structure
model = BayesianNetwork([('Weather', 'Traffic'), ('Traffic', 'Late')])

# Define Conditional Probability Tables (CPTs)
cpd_weather = TabularCPD(variable='Weather', variable_card=2, values=[[0.7],␣
  ↪[0.3]])
cpd_traffic = TabularCPD(variable='Traffic', variable_card=2,
                         values=[[0.8, 0.3],
                                 [0.2, 0.7]],
                         evidence=['Weather'],
                         evidence_card=[2])
cpd_late = TabularCPD(variable='Late', variable_card=2,
                      values=[[0.9, 0.6],
                              [0.1, 0.4]],
                      evidence=['Traffic'],
                      evidence_card=[2])

# Add CPDs to the model
```

```
model.add_cpds(cpd_weather, cpd_traffic, cpd_late)
```

**Exercise 3: Querying the Bayesian Network**

[4]:

```
# Perform exact inference
infer = VariableElimination(model)
result = infer.query(['Late'], evidence={'Weather': 1})  # 1 represents Rainy
print("Probability of being Late given that it's Rainy:")
print(result)
```

```
Probability of being Late given that it's Rainy:
+---------+-------------+
| Late    |   phi(Late) |
+=========+=============+
| Late(0) |      0.6900 |
+---------+-------------+
| Late(1) |      0.3100 |
+---------+-------------+
```

**Exercise 4: Parameter Learning**

[5]:

```
# Simulate a dataset
np.random.seed(42)
n_samples = 1000
weather = np.random.choice([0, 1], size=n_samples, p=[0.7, 0.3])
traffic = np.where(weather == 0,
                   np.random.choice([0, 1], size=n_samples, p=[0.8, 0.2]),
                   np.random.choice([0, 1], size=n_samples, p=[0.3, 0.7]))
late = np.where(traffic == 0,
                np.random.choice([0, 1], size=n_samples, p=[0.9, 0.1]),
                np.random.choice([0, 1], size=n_samples, p=[0.6, 0.4]))

data = pd.DataFrame({'Weather': weather, 'Traffic': traffic, 'Late': late})

# Estimate parameters
mle = MaximumLikelihoodEstimator(model, data)
cpd_traffic_mle = mle.estimate_cpd('Traffic')
cpd_late_mle = mle.estimate_cpd('Late')

print("Estimated CPD for Traffic:")
print(cpd_traffic_mle)
print("\nEstimated CPD for Late:")
print(cpd_late_mle)
```

```
Estimated CPD for Traffic:
+------------+---------------------+--------------------+
| Weather    | Weather(0)          | Weather(1)         |
+------------+---------------------+--------------------+
| Traffic(0) | 0.7865168539325843  | 0.2951388888888889 |
+------------+---------------------+--------------------+
| Traffic(1) | 0.21348314606741572 | 0.7048611111111112 |
+------------+---------------------+--------------------+
Estimated CPD for Late:
+---------+---------------------+---------------------+
| Traffic | Traffic(0)          | Traffic(1)          |
+---------+---------------------+---------------------+
| Late(0) | 0.9209302325581395  | 0.6084507042253521  |
+---------+---------------------+---------------------+
| Late(1) | 0.07906976744186046 | 0.39154929577464787 |
+---------+---------------------+---------------------+
```

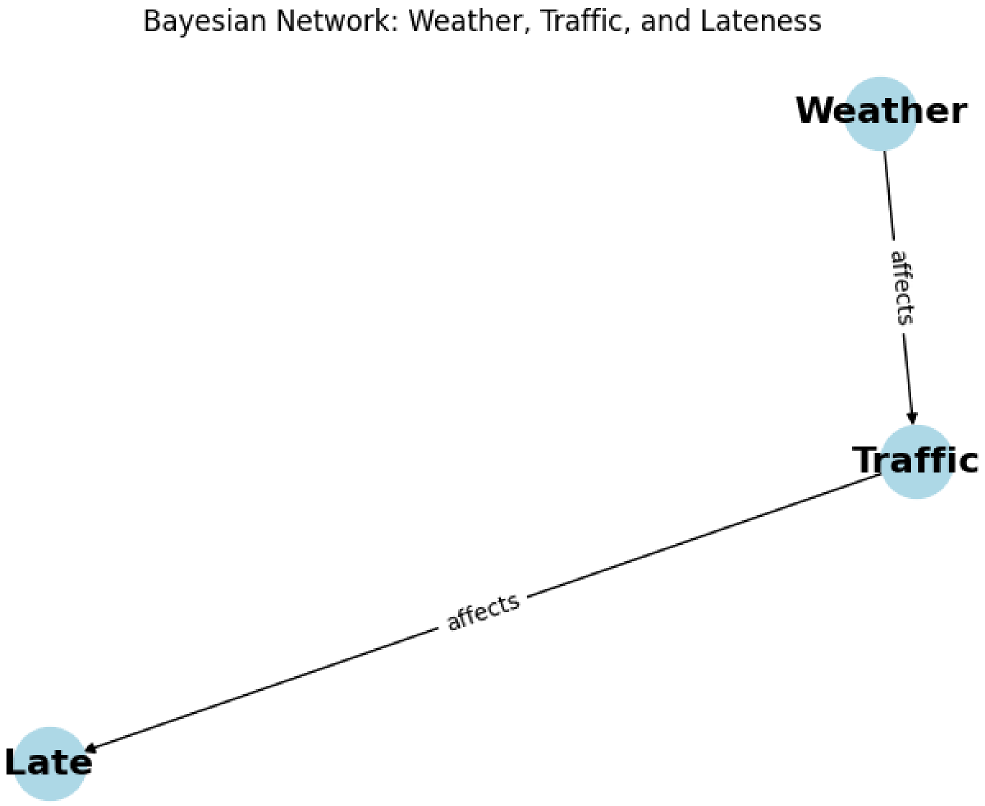
### Exercise 5: Visualizing the Bayesian Network

[6]:

```
# Create a directed graph
G = nx.DiGraph()
G.add_edges_from(model.edges())

# Draw the graph
pos = nx.spring_layout(G)
nx.draw(G, pos, with_labels=True, node_color='lightblue',
        node_size=1000, font_size=16, font_weight='bold')
edge_labels = {('Weather', 'Traffic'): 'affects', ('Traffic', 'Late'):␣
 ↪'affects'}
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels)

plt.title("Bayesian Network: Weather, Traffic, and Lateness")
plt.axis('off')
plt.show()
```


Bayesian Network: Weather, Traffic, and Lateness
Weather
affects
Traffic
affects
Late