```python
#!pip install pgmpy
from pgmpy.models import DiscreteBayesianNetwork
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination

# Estrutura da rede
modelo = DiscreteBayesianNetwork([('Chuva', 'Umidade'), ('Irrigacao', 'Umidade')])

# Definindo CPDs
cpd_chuva = TabularCPD(variable='Chuva', variable_card=2, values=[[0.7], [0.3]])
cpd_irrigacao = TabularCPD(variable='Irrigacao', variable_card=2, values=[[0.6], [0.4]])
cpd_umidade = TabularCPD(
  variable='Umidade', variable_card=2,
  values=[
    [0.8, 0.4, 0.2, 0.1],  # P(Umidade=0)
    [0.2, 0.6, 0.8, 0.9]   # P(Umidade=1)
  ],
  evidence=['Chuva', 'Irrigacao'],
  evidence_card=[2, 2]
)

# Adicionando CPDs ao modelo
modelo.add_cpds(cpd_chuva, cpd_irrigacao, cpd_umidade)

# Verificando o modelo
modelo.check_model()

# Inferência
inferencia = VariableElimination(modelo)
resultado = inferencia.query(variables=['Umidade'], evidence={'Chuva': 1})
print(resultado)
```

Explicação do Código

1. Importação das Bibliotecas

```
from pgmpy.models import DiscreteBayesianNetwork
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination
```

DiscreteBayesianNetwork → Cria a estrutura da Rede Bayesiana.
TabularCPD → Define as Probabilidades Condicionais (CPDs).
VariableElimination → Permite fazer inferências na rede.

2. Definição da Estrutura da Rede

```
modelo = DiscreteBayesianNetwork([('Chuva', 'Umidade'), ('Irrigacao', 'Umidade')])
```

- Criamos uma Rede Bayesiana Discreta com duas relações:
    - Chuva → Umidade 🌧️ ➡️ 🌱
    - Irrigação → Umidade 🚿 ➡️ 🌱
- Ou seja, a umidade do solo depende tanto da chuva quanto da irrigação.

Representação gráfica da rede:

```markdown
Chuva        Irrigação
  |              |
  └──► Umidade
```

3. Definição das Probabilidades Condicionais (CPDs)

CPD 1: Probabilidade de Chuva

```
cpd_chuva = TabularCPD(variable='Chuva', variable_card=2, values=[[0.7], [0.3]])
```

A variável "Chuva" pode ter dois estados:

- 0 = Não chove
- 1 = Chove

Probabilidades:

- 70% de chance de NÃO chover ⛅ (P(Chuva=0) = 0.7)
- 30% de chance de chover 🌧️ (P(Chuva=1) = 0.3)

CPD 2: Probabilidade de Irrigação

```
cpd_irrigacao = TabularCPD(variable='Irrigacao', variable_card=2, values=[[0.6], [0.4]])
```

A variável "Irrigacao" também pode ter dois estados:

- 0 = Não há irrigação
- 1 = Há irrigação

Probabilidades:

- 60% de chance de NÃO haver irrigação 🚫 🚿 (P(Irrigacao=0) = 0.6)
- 40% de chance de haver irrigação 🚿 (P(Irrigacao=1) = 0.4)

CPD 3: Probabilidade de Umidade Condicionada à Chuva e Irrigação

```
cpd_umidade = TabularCPD(
    variable='Umidade', variable_card=2,
    values=[
        [0.8, 0.4, 0.2, 0.1],  # P(Umidade=0) (Solo seco)
        [0.2, 0.6, 0.8, 0.9]   # P(Umidade=1) (Solo úmido)
    ],
    evidence=['Chuva', 'Irrigacao'],
    evidence_card=[2, 2]
)
```

- A Umidade depende de Chuva e Irrigação (variáveis evidence=['Chuva', 'Irrigacao']).
- A matriz values define as probabilidades para cada combinação de Chuva e Irrigação:

Chuva	Irrigação	P(Solo Seco)	P(Solo Úmido)
0 (Não)	0 (Não)	80%	20%
0 (Não)	1 (Sim)	40%	60%
1 (Sim)	0 (Não)	20%	80%
1 (Sim)	1 (Sim)	10%	90%

Como ler essa tabela?

- Se não choveu e não houve irrigação, o solo tem 80% de chance de estar seco e 20% de estar úmido.
- Se choveu e houve irrigação, o solo tem 90% de chance de estar úmido e 10% de estar seco.

4. **Adicionando as CPDs ao Modelo**

```
modelo.add_cpds(cpd_chuva, cpd_irrigacao, cpd_umidade)
```

Agora, a rede já sabe **como calcular probabilidades** a partir das CPDs.

5. **Verificação do Modelo**

```
modelo.check_model()
```

Essa linha **confirma** que as probabilidades estão corretas e a rede é **matematicamente válida**.

6. **Inferência na Rede Bayesiana**

```
inferencia = VariableElimination(modelo)
```

Criamos um **objeto de inferência**, que nos permite **fazer perguntas** ao modelo.

7. **Consulta: Qual a Probabilidade de Umidade dado que Chove?**

```
resultado = inferencia.query(variables=['Umidade'], evidence={'Chuva': 1})
print(resultado)
```

variables=['Umidade'] → Queremos saber a probabilidade de umidade.
evidence={'Chuva': 1} → Sabemos que choveu e queremos ver como isso afeta a umidade.

8. **Saída Esperada**

```
+------------+----------+
| Umidade(0) | 0.12     |
| Umidade(1) | 0.88     |
+------------+----------+
```

Interpretação:

Se sabemos que **choveu** (Chuva = 1), então:

- **12% de chance do solo estar seco**.
- **88% de chance do solo estar úmido**.

Como isso foi calculado?
O modelo usou a **Regra de Bayes** para combinar todas as probabilidades da rede e chegar nessa resposta.

Usando a Regra da Probabilidade Total e Regra de Bayes

Vamos calcular:

P(Chove | Solo úmido) → inversão com a **Regra de Bayes:**

$$P(Chuva|Umido) = \frac{P(Umido|Chuva) \cdot P(Chuva)}{P(Umido)}$$

Mas primeiro precisamos calcular:

$$P(Umido) = P(Umido|Chuva) \cdot P(Chuva) + P(Umido|NãoChove) \cdot P(NãoChove)$$

```
# Probabilidades a priori
P_chuva = 0.3
P_nao_chuva = 0.7

# Condicionais
P_umido_dado_chuva = 0.8
P_umido_dado_nao_chuva = 0.1

# Passo 1: Regra da probabilidade total - calcular P(Umido)
P_umido = P_umido_dado_chuva * P_chuva + P_umido_dado_nao_chuva * P_nao_chuva

# Passo 2: Regra de Bayes - calcular P(Chuva | Umido)
P_chuva_dado_umido = (P_umido_dado_chuva * P_chuva) / P_umido

# Mostrando os resultados
print(f"P(Umido) = {P_umido:.4f}")
print(f"P(Chuva | Umido) = {P_chuva_dado_umido:.4f}")
```

Saída esperada:

```
P(Umido) = 0.31
P(Chuva | Umido) = 0.7742
```

Interpretação:

- A chance de o solo estar úmido, sem sabermos se choveu: **31%**
- Se **sabemos que o solo está úmido**, então a chance de **ter chovido** é de **77,42%**