```
In [ ]: !pip install pgmpy
```

```
In [3]: import pandas as pd
        from pgmpy.models import BayesianNetwork
        from pgmpy.estimators import MaximumLikelihoodEstimator, BayesianEstimator
        from pgmpy.inference import VariableElimination
```

```
In [4]: url = "https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/p
        columns = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach'.
```

```
In [11]: def process_chunk(chunk):
             processed_chunk = chunk.dropna()  # Example: remove NaN values
             return processed_chunk

         all_chunks = []

         for chunk in pd.read_csv(url, header=None, names=columns, na_values='?', chunksi
             processed = process_chunk(chunk)
             all_chunks.append(processed)

         final_data = pd.concat(all_chunks, ignore_index=True)
```

```
In [12]: data['target'] = data['target'].apply(lambda x: 0 if x == 0 else 1)
```

```
In [15]: model = BayesianNetwork([('age', 'target'), ('sex', 'target'), ('cp', 'target'),
                                  ('trestbps', 'target'), ('chol', 'target'),
                                  ])
```

```
In [16]: model.fit(data, estimator=MaximumLikelihoodEstimator)
```

```
In [17]: infer = VariableElimination(model)
```

```
In [18]: query_result = infer.map_query(variables=['target'], evidence={'age': 50, 'sex':
```
```
          0%|            | 0/2 [00:00<?, ?it/s]
          0%|            | 0/2 [00:00<?, ?it/s]
```

```
In [19]: if query_result['target'] == 1:
             print("Diagnosis: The patient is likely to have heart disease.")
         else:
             print("Diagnosis: The patient is unlikely to have heart disease.")
```

Diagnosis: The patient is unlikely to have heart disease.

```
In [21]: query_result = infer.map_query(variables=['target'], evidence={'age': 59, 'sex':
```
```
          0%|            | 0/2 [00:00<?, ?it/s]
          0%|            | 0/2 [00:00<?, ?it/s]
```

```
In [23]: if query_result['target'] == 0:
             print("Diagnosis: The patient is likely to have heart disease.")
         else:
             print("Diagnosis: The patient is unlikely to have heart disease.")
```

Diagnosis: The patient is likely to have heart disease.

In [ ]: