PROGRAM 7: Write a program to construct a **Bayesian network** considering

medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML

|  |
| --- |
| import numpy as np  import pandas as pd  import csv  from pgmpy.estimators import MaximumLikelihoodEstimator  from pgmpy.models import BayesianModel  from pgmpy.inference import VariableElimination  #Read the attributes  lines = list(csv.reader(open('C:\\Users\\ISE\\Desktop\\7\\data7\_names.csv', 'r')));  attributes = lines[0]  #attributes = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang','oldpeak', 'slope', 'ca', 'thal', 'heartdisease']  #Read Cleveland Heart dicease data  heartDisease = pd.read\_csv('C:\\Users\\ISE\\Desktop\\7\\data7\_heart.csv', names = attributes)  heartDisease = heartDisease.replace('?', np.nan)  # Display the data  print('Few examples from the dataset are given below')  print(heartDisease.head())  print('\nAttributes and datatypes')  print(heartDisease.dtypes)  # Model Baysian Network  model = BayesianModel([('age', 'trestbps'), ('age', 'fbs'), ('sex', 'trestbps'), ('sex', 'trestbps'),  ('exang', 'trestbps'),('trestbps','heartdisease'),('fbs','heartdisease'),  ('heartdisease','restecg'),('heartdisease','thalach'),('heartdisease','chol')])  # Learning CPDs using Maximum Likelihood Estimators  print('\nLearning CPDs using Maximum Likelihood Estimators...');  model.fit(heartDisease, estimator=MaximumLikelihoodEstimator)  # Inferencing with Bayesian Network  print('\nInferencing with Bayesian Network:')  HeartDisease\_infer = VariableElimination(model)  # Computing the probability of bronc given smoke.  print('\n1.Probability of HeartDisease given Age=20')  q = HeartDisease\_infer.query(variables=['heartdisease'], evidence={'age': 28 })  print(q['heartdisease'])  print('\n2. Probability of HeartDisease given chol (Cholestoral) =100')  q = HeartDisease\_infer.query(variables=['heartdisease'], evidence={'chol': 100})  print(q['heartdisease']) |

Few examples from the dataset are given below

age sex cp trestbps chol fbs restecg thalach exang oldpeak \

0 63.0 1.0 1.0 145.0 233.0 1.0 2.0 150.0 0.0 2.3

1 67.0 1.0 4.0 160.0 286.0 0.0 2.0 108.0 1.0 1.5

2 67.0 1.0 4.0 120.0 229.0 0.0 2.0 129.0 1.0 2.6

3 37.0 1.0 3.0 130.0 250.0 0.0 0.0 187.0 0.0 3.5

4 41.0 0.0 2.0 130.0 204.0 0.0 2.0 172.0 0.0 1.4

slope ca thal heartdisease

0 3.0 0.0 6.0 0

1 2.0 3.0 3.0 2

2 2.0 2.0 7.0 1

3 3.0 0.0 3.0 0

4 1.0 0.0 3.0 0

Attributes and datatypes

age float64

sex float64

cp float64

trestbps float64

chol float64

fbs float64

restecg float64

thalach float64

exang float64

oldpeak float64

slope float64

ca object

thal object

heartdisease int64

dtype: object

Learning CPDs using Maximum Likelihood Estimators...

Inferencing with Bayesian Network:

1.Probability of HeartDisease given Age=20

╒════════════════╤═════════════════════╕

│ heartdisease │ phi(heartdisease) │

╞════════════════╪═════════════════════╡

│ heartdisease\_0 │ 0.6791 │

├────────────────┼─────────────────────┤

│ heartdisease\_1 │ 0.1212 │

├────────────────┼─────────────────────┤

│ heartdisease\_2 │ 0.0810 │

├────────────────┼─────────────────────┤

│ heartdisease\_3 │ 0.0939 │

├────────────────┼─────────────────────┤

│ heartdisease\_4 │ 0.0247 │

╘════════════════╧═════════════════════╛

2. Probability of HeartDisease given chol (Cholestoral) =100

╒════════════════╤═════════════════════╕

│ heartdisease │ phi(heartdisease) │

╞════════════════╪═════════════════════╡

│ heartdisease\_0 │ 0.5400 │

├────────────────┼─────────────────────┤

│ heartdisease\_1 │ 0.1533 │

├────────────────┼─────────────────────┤

│ heartdisease\_2 │ 0.1303 │

├────────────────┼─────────────────────┤

│ heartdisease\_3 │ 0.1259 │

├────────────────┼─────────────────────┤

│ heartdisease\_4 │ 0.0506 │

╘════════════════╧═════════════════════╛

PROGRAM 8: **Program 8 :** Apply **EM algorithm** to cluster a set of data stored in a .CSV file. Use the same data set for clustering using ***k*-Means algorithm**. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

|  |
| --- |
| **import** **matplotlib.pyplot** **as** **plt**  **from** **sklearn** **import** datasets  **from** **sklearn.cluster** **import** KMeans  **import** **pandas** **as** **pd**  **import** **numpy** **as** **np**  iris = datasets.load\_iris()  X = pd.DataFrame(iris.data)  X.columns = ['Sepal\_Length','Sepal\_Width','Petal\_Length','Petal\_Width']  y = pd.DataFrame(iris.target)  y.columns = ['Targets']  model = KMeans(n\_clusters=3)  model.fit(X)  plt.figure(figsize=(14, 14))  colormap = np.array(['red', 'lime', 'black'])  plt.subplot(2, 2, 1)  plt.scatter(X.Petal\_Length, X.Petal\_Width, c=colormap[y.Targets], s=40)  plt.title('Real Clusters')  plt.xlabel('Petal Length')  plt.ylabel('Petal Width')  plt.subplot(2, 2, 2)  plt.scatter(X.Petal\_Length, X.Petal\_Width, c=colormap[model.labels\_], s=40)  plt.title('K-Means Clustering')  plt.xlabel('Petal Length')  plt.ylabel('Petal Width')  **from** **sklearn** **import** preprocessing  scaler = preprocessing.StandardScaler()  scaler.fit(X)  xsa = scaler.transform(X)  xs = pd.DataFrame(xsa, columns=X.columns)  **from** **sklearn.mixture** **import** GaussianMixture  gmm = GaussianMixture(n\_components=3)  gmm.fit(xs)  gmm\_y = gmm.predict(xs)  plt.subplot(2, 2, 3)  plt.scatter(X.Petal\_Length, X.Petal\_Width, c=colormap[gmm\_y], s=40)  plt.title('GMM Clustering')  plt.xlabel('Petal Length')  plt.ylabel('Petal Width')  plt.show()  print('Observation: The GMM using EM algorithm based clustering matched the true labels more closely than the Kmeans.') |

Out put :

