

AIM:

Program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard heart disease dataset.

Algorithm:

1. Identify the variables which is set of attributes in the dataset
2. Determine the domain of each variable that is set of values a variable may take
3. Create a directed graph network of nodes where each node represents the attribute and edges represent parent child relationship. Edge represents that the child variable is conditionally dependent on the parent.
4. Determine the prior and conditional probability for each attribute
5. Perform the inference on the model & determine the marginal probabilities.

Dataset: heart\_disease\_data.csv

Age	Gender	Family History	Diet	Lifestyle	Cholesterol	Heart Disease
Super Senior Citizen	Male	Yes	Medium	Sedentary	High	Yes
Super Senior Citizen	Female	Yes	Medium	Sedentary	High	Yes
Senior Citizen	Male	No	High	Moderate	Borderline	Yes
Teen	Male	Yes	Medium	Sedentary	Normal	No
Youth	Female	Yes	High	Athlete	Normal	No
Middle Aged	Male	Yes	Medium	Active	High	Yes
Teen	Male	Yes	High	Moderate	High	Yes
Super Senior Citizen	Male	Yes	Medium	Sedentary	High	Yes
Youth	Female	Yes	High	Athlete	Normal	No
Senior Citizen	Female	No	High	Athlete	Normal	Yes
Teen	Female	No	Medium	Moderate	High	Yes
Teen	Male	Yes	Medium	Sedentary	Normal	No
Teen	Female	No	High	Athlete	High	No
Middle Aged	Female	No	High	Athlete	High	No
Middle Aged	Male	Yes	Medium	Active	High	Yes

Age	Gender	FamilyHistory	Diet	Lifestyle	Cholesterol	Heart Disease
Youth	Female	Yes	High	Athlete	Borderline	No
SuperSenior Citizen	Male	Yes	High	Athlete	Normal	Yes
Senior Citizen	Female	No	Medium	Moderate	Borderline	Yes
Youth	Female	Yes	Medium	Athlete	Borderline	No
Teen	Male	Yes	Medium	Sedentary	Normal	No

### Program:

```
import numpy as np
```

```
import numpy as np
```

```
import csv
```

```
from sklearn import metrics
```

```
from sklearn import train_test_split
```

```
ageEnum = {'SuperSeniorCitizen': 0, 'Senior Citizen': 1, 'Middle Aged': 2, 'Youth': 3, 'Teen': 4}
```

```
genderEnum = {'Male': 0, 'Female': 1}
```

```
familyHistoryEnum = {'Yes': 0, 'No': 1}
```

```
dietEnum = {'High': 0, 'Medium': 1, 'Low': 2}
```

```
lifestyleEnum = {'Athlete': 0, 'Active': 1, 'Moderate': 2, 'Sedentary': 3}
```

```
cholesterolEnum = {'High': 0, 'Borderline': 1, 'Normal': 2}
```

```
heartDiseaseEnum = {'Yes': 0, 'No': 1}
```

```
with open('heart_disease_data.csv') as csvfile: reader = csv.reader(csvfile)
```

```
data = next(reader) data = []
```

```
for x in data:
```

```
    data.append([ageEnum[x[0]], genderEnum[x[1]], familyHistoryEnum[x[2]], dietEnum[x[3]], lifestyleEnum[x[4]], cholesterolEnum[x[5]], heartDiseaseEnum[x[6]])
```

```
N = len(data)
```

```
p_age = p.nodes.Dirichlet(1.0 * np.ones(5))
```

```
age = bp.nodes.Categorical(p_age, plates=(N,)) age.observe(data[:, 0])
```

```
p_gender = bp.nodes.Dirichlet(1.0 * np.ones(2)) gender = bp.nodes.Categorical(p_gender,
    plates=(N,)) gender.observe(data[:, 1])
```

```
p_familyhistory = bp.nodes.Dirichlet(1.0 * np.ones(2)) familyhistory = bp.nodes.
    Categorical(p_familyhistory, plates=(N,)) familyhistory.observe(data[:, 2])
```

```
p_diet = bp.nodes.Dirichlet(1.0 * np.ones(3)) diet = bp.nodes.Categorical(p_diet,
    plates=(N,)) diet.observe(data[:, 3])
```

```
p_lifestyle = bp.nodes.Dirichlet(1.0 * np.ones(4)) lifestyle = bp.nodes.Categorical
    (p_cholesterol, plates=(N,)) cholesterol.observe(data[:, 5])
```

```
p_heartdisease = bp.nodes.Dirichlet(np.ones(2), plates=(5, 2, 2, 3, 4, 3))
```

```
heartdisease = bp.nodes.Multimixture([age, gender, familyhistory, diet, lifestyle,
    cholesterol], bp.nodes.Categorical, p_heartdisease)
```

```
heartdisease.observe(data[:, 6])
```

```
p_heartdisease.update()
```

```
familyhistory = Yes, DietIntake = Medium, lifestyle = Sedentary, cholesterol = High")
```

```
familyhistoryEnum['Yes'], dietEnum['Medium'], lifestyleEnum['Sedentary'],
    cholesterolEnum['High'], bp.nodes.Categorical, p_heartdisease).get_moments()
[0][heartDiseaseEnum['Yes']])
```

```
while m == 0:
```

```
    print("\n")
```

```
    res = bp.nodes.Multimixture([int(input('Enter Age: ' + str(ageEnum))),
    int(input('Enter Gender: ' + str(genderEnum))), int(input('Enter
    FamilyHistory: ' + str(familyhistoryEnum))), int(input('Enter
    dietEnum: ' + str(dietEnum))), int(input('Enter lifestyle: ' +
```



```

str (lifestyleEnum))), int (input ('Enter cholesterol' + str (cholesterolEnum)))
bp.nodes. categorical, p_heartdisease).get_moments (1 [0] [heartDiseaseEnum[Yes]])
print ("probability (HeartDisease) = " + str (res))
m= int (input ("Enter for continue:0, Exit:1"))

```

output:

```

Enter Age: y{'SuperSeniorCitizen': 0, 'SeniorCitizen': 1, 'MiddleAged': 2, 'Youth': 3, 'Teen': 4}1
Enter Gender: {'Male': 0, 'Female': 1}1
Enter FamilyHistory: {'Yes': 0, 'No': 1}1
Enter dietEnum: {'High': 0, 'Medium': 1, 'Low': 2}2
Enter LifeStyle: {'Athlete': 0, 'Active': 1, 'Moderate': 2, 'Sedetary': 3}2
Enter Cholesterol: {'High': 0, 'BorderLine': 1, 'Normal': 2}1
Probability(HeartDisease) = 0.5
Enter for Continue:0, Exit :1 0

```

```

Enter Age: y{'SuperSeniorCitizen': 0, 'SeniorCitizen': 1, 'MiddleAged': 2, 'Youth': 3, 'Teen': 4}0
Enter Gender: {'Male': 0, 'Female': 1}0
Enter FamilyHistory: {'Yes': 0, 'No': 1}0
Enter dietEnum: {'High': 0, 'Medium': 1, 'Low': 2}0
Enter LifeStyle: {'Athlete': 0, 'Active': 1, 'Moderate': 2, 'Sedetary': 3}3
Enter Cholesterol: {'High': 0, 'BorderLine': 1, 'Normal': 2}0
Probability(HeartDisease) = 0.5
Enter for Continue:0, Exit :1

```