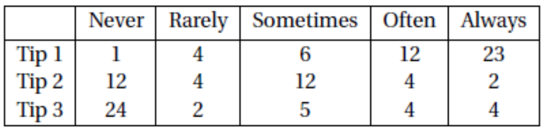
1.1 PROBABILITY

1.1.1 PROBABILITY DISTRIBUTION TABLE



	Never	Rarely	Sometimes	Often	Always
Tip 1	1	4	6	12	23
Tip 2	12	4	12	4	2
Tip 3	24	2	5	4	4

Tip 1 : “Study hard and you will do well, fail to do so and you will not”

	Tip 1	P(Tip1)
Never	1	0.02173
Rarely	4	0.0869
Sometimes	6	0.1304
Often	12	0.26
Always	23	0.5

- Probability of student never follow Tip 1 and yet succeed = 0.02173
- Probability of student rarely follow Tip 1 and yet succeed = 0.0869
- Probability of student sometimes follow Tip 1 and yet succeed = 0.1304
- Probability of student often follow Tip 1 and yet succeed = 0.26
- Probability of student always follow Tip 1 and succeed = 0.5

Tip 2 “Get plenty of rest and you will do well, fail to do so and you will not”

	Tip 2	P(Tip2)
Never	12	0.352941
Rarely	4	0.117647
Sometimes	12	0.352941
Often	4	0.117647
Always	2	0.058824

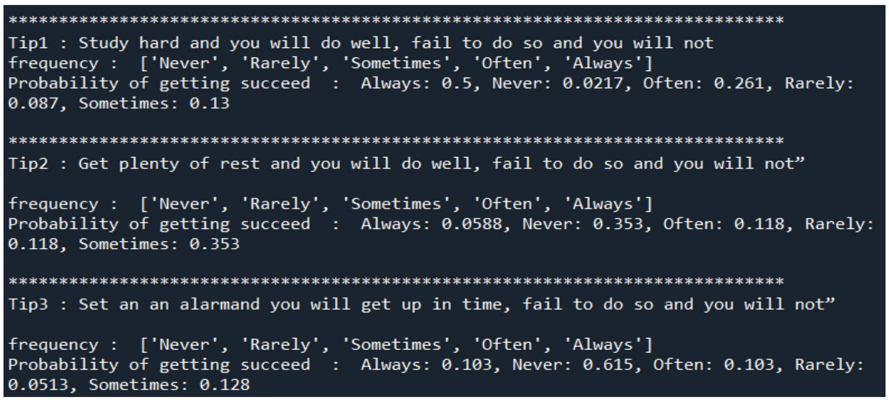
- Probability of student never follow Tip 2 and yet succeed = 0.352941
- Probability of student rarely follow Tip 2 and yet succeed = 0.117647
- Probability of student sometimes follow Tip 2 and yet succeed = 0.352941
- Probability of student often follow Tip 2 and yet succeed =0.117647
- Probability of student always follow Tip 2 and succeed = 0.058824

Tip 3 “Set an an alarm and you will get up in time, fail to do so and you will not”.

	Tip 2	Probability
Never	24	0.648649
Rarely	2	0.054054
Sometimes	5	0.135135
Often	4	0.108108
Always	4	0.054054

- Probability of student never follow Tip 3 and yet succeed = 0.648649

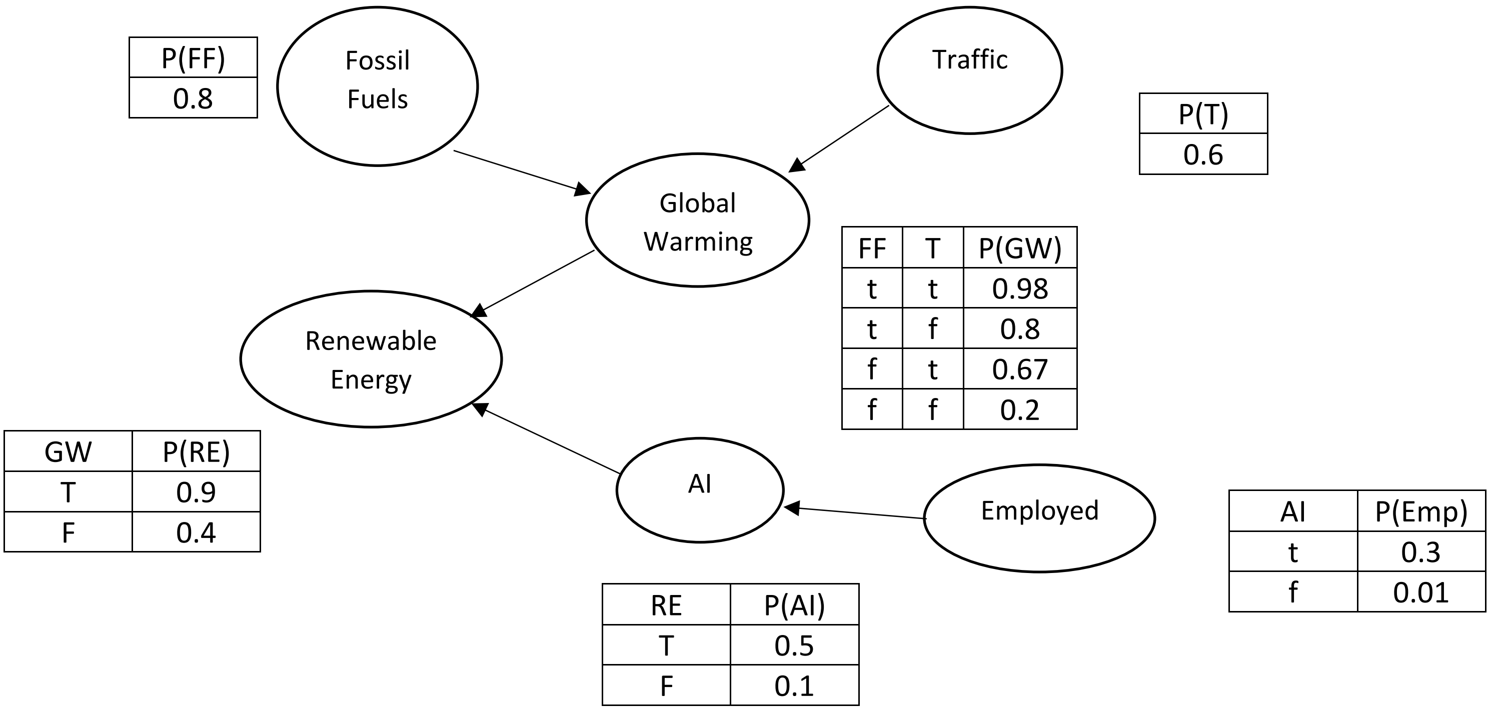
- Probability of student rarely follow Tip 3 and yet succeed = 0.054054
- Probability of student sometimes follow Tip 3 and yet succeed = 0.135135
- Probability of student often follow Tip 3 and yet succeed = 0.108108
- Probability of student always follow Tip 3 and succeed = 0.054054



```
******************************************************************************
Tip1 : Study hard and you will do well, fail to do so and you will not
frequency :  ['Never', 'Rarely', 'Sometimes', 'Often', 'Always']
Probability of getting succeed  :  Always: 0.5, Never: 0.0217, Often: 0.261, Rarely:
0.087, Sometimes: 0.13

******************************************************************************
Tip2 : Get plenty of rest and you will do well, fail to do so and you will not”

frequency :  ['Never', 'Rarely', 'Sometimes', 'Often', 'Always']
Probability of getting succeed  :  Always: 0.0588, Never: 0.353, Often: 0.118, Rarely:
0.118, Sometimes: 0.353

******************************************************************************
Tip3 : Set an an alarmand you will get up in time, fail to do so and you will not”

frequency :  ['Never', 'Rarely', 'Sometimes', 'Often', 'Always']
Probability of getting succeed  :  Always: 0.103, Never: 0.615, Often: 0.103, Rarely:
0.0513, Sometimes: 0.128
```

1.1.2 BAYESIAN NETWORKS

You have been given the following Random Variables:
- Fossil Fuels
- AI
- Global Warming
- Renewable Energy
- Traffic
- Employed



P(FF)
0.8

P(T)
0.6

FF	T	P(GW)
t	t	0.98
t	f	0.8
f	t	0.67
f	f	0.2

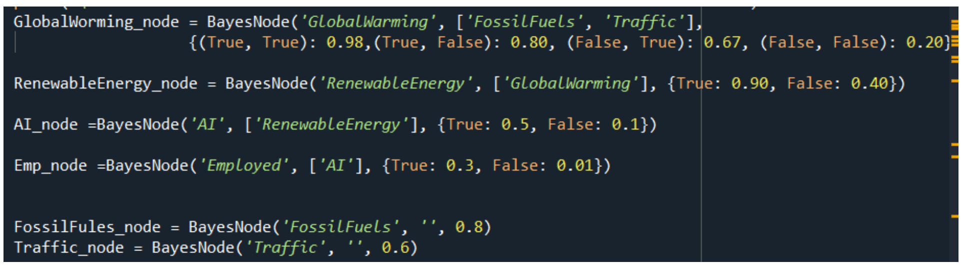
GW	P(RE)
T	0.9
F	0.4

RE	P(AI)
T	0.5
F	0.1

AI	P(Emp)
t	0.3
f	0.01

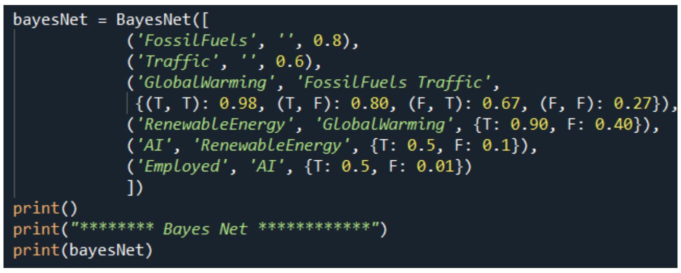
- Fossil Fuels and Traffic are two causes of Global Warming
- We can control global warming using Renewable Energy
- AI technology improve the renewable energy sector
- Employed develop the AI technology in the renewable energy sector

Setting the node



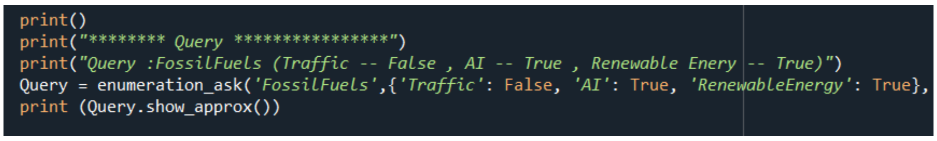
```
GlobalWorming_node = BayesNode('GlobalWarming', ['FossilFuels', 'Traffic'],
                    {(True, True): 0.98,(True, False): 0.80, (False, True): 0.67, (False, False): 0.20}

RenewableEnergy_node = BayesNode('RenewableEnergy', ['GlobalWarming'], {True: 0.90, False: 0.40})

AI_node =BayesNode('AI', ['RenewableEnergy'], {True: 0.5, False: 0.1})

Emp_node =BayesNode('Employed', ['AI'], {True: 0.3, False: 0.01})


FossilFules_node = BayesNode('FossilFuels', '', 0.8)
Traffic_node = BayesNode('Traffic', '', 0.6)
```

Apply the Bayes Net function



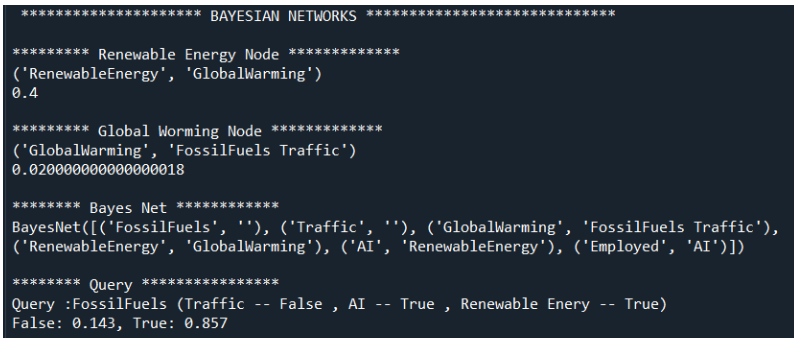
```
bayesNet = BayesNet([
            ('FossilFuels', '', 0.8),
            ('Traffic', '', 0.6),
            ('GlobalWarming', 'FossilFuels Traffic',
             {(T, T): 0.98, (T, F): 0.80, (F, T): 0.67, (F, F): 0.27}),
            ('RenewableEnergy', 'GlobalWarming', {T: 0.90, F: 0.40}),
            ('AI', 'RenewableEnergy', {T: 0.5, F: 0.1}),
            ('Employed', 'AI', {T: 0.5, F: 0.01})
            ])
print()
print("******** Bayes Net ************")
print(bayesNet)
```

Evaluate the query



```
print()
print("******** Query ****************")
print("Query :FossilFuels (Traffic -- False , AI -- True , Renewable Enery -- True)")
Query = enumeration_ask('FossilFuels',{'Traffic': False, 'AI': True, 'RenewableEnergy': True},
print (Query.show_approx())
```

Query : evaluate the probability of fossil fuels in the respect of traffic is False ,AI is True and Renewable Energy is True



```
 ********************* BAYESIAN NETWORKS *****************************

********* Renewable Energy Node *************
('RenewableEnergy', 'GlobalWarming')
0.4

********* Global Worming Node *************
('GlobalWarming', 'FossilFuels Traffic')
0.020000000000000018

******** Bayes Net ************
BayesNet([('FossilFuels', ''), ('Traffic', ''), ('GlobalWarming', 'FossilFuels Traffic'),
('RenewableEnergy', 'GlobalWarming'), ('AI', 'RenewableEnergy'), ('Employed', 'AI')])

******** Query ****************
Query :FossilFuels (Traffic -- False , AI -- True , Renewable Enery -- True)
False: 0.143, True: 0.857
```

Bayesian networks :

- Probabilistic graphical model
- apply Bayes Theorem for conditionally dependent and conditionally independent relationship between random nodes

- It gives the compact ,graphical representation of joint probability distribution of using conditional independence

Pros :

1. Structural Learning
2. Fast response
3. Decision analysis

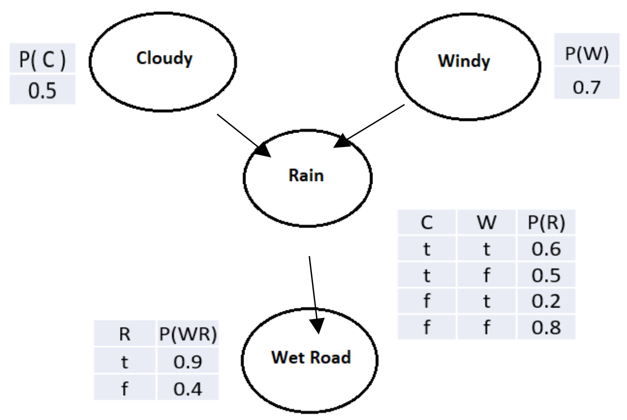
Cons :

1. Suitable for small dataset
2. Not supported for loop networks

Example :

The probability of cloudy 0.5 and probability of windy is 0.7

If both are high then the probability of road getting wet is high



P(C)
0.5

P(W)
0.7

C	W	P(R)
t	t	0.6
t	f	0.5
f	t	0.2
f	f	0.8

R	P(WR)
t	0.9
f	0.4

1.2.2 NAIVE BAYES LEARNER

DataSet : Lenses (https://archive.ics.uci.edu/ml/datasets/Lenses)

Dataset Information :

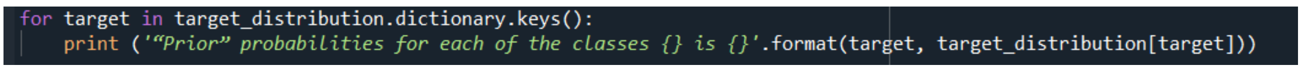
- The examples are complete and noise free.
- The examples highly simplified the problem. The attributes do not
- fully describe all the factors affecting the decision as to which type,
- if any, to fit.

Attribute Information:

2 Classes
1 : the patient should be fitted with hard contact lenses,
2 : the patient should be fitted with soft contact lenses,
3 : the patient should not be fitted with contact lenses.

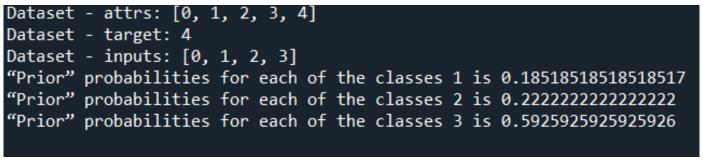
1. age of the patient: (1) young, (2) pre-presbyopic, (3) presbyopic
2. spectacle prescription: (1) myope, (2) hypermetrope
3. astigmatic: (1) no, (2) yes
4. tear production rate: (1) reduced, (2) normal

1. Load the data Lenses and calculate the prior probabilities for each class



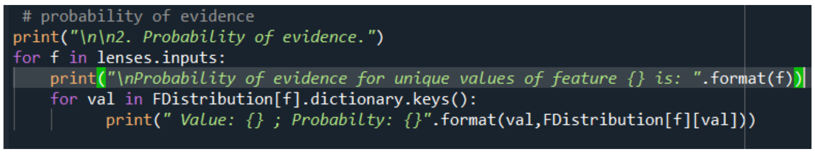
```
for target in target_distribution.dictionary.keys():
    print ('“Prior” probabilities for each of the classes {} is {}'.format(target, target_distribution[target]))
```

Output :



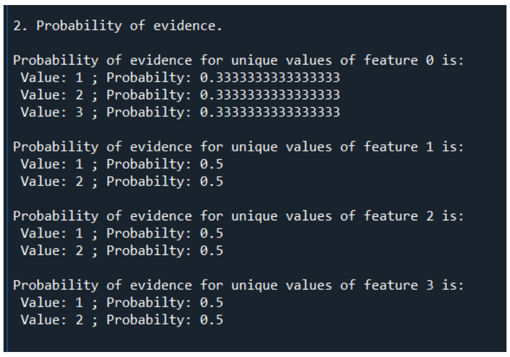
```
Dataset - attrs: [0, 1, 2, 3, 4]
Dataset - target: 4
Dataset - inputs: [0, 1, 2, 3]
"Prior" probabilities for each of the classes 1 is 0.18518518518518517
"Prior" probabilities for each of the classes 2 is 0.2222222222222222
"Prior" probabilities for each of the classes 3 is 0.5925925925925926
```

2.Calculate the probabilty of evidence



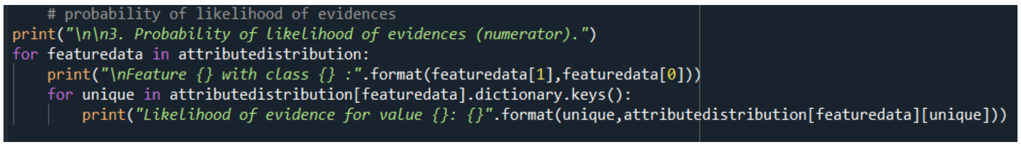
```
 # probability of evidence
print("\n\n2. Probability of evidence.")
for f in lenses.inputs:
    print("\nProbability of evidence for unique values of feature {} is: ".format(f))
    for val in FDistribution[f].dictionary.keys():
        print(" Value: {} ; Probabilty: {}".format(val,FDistribution[f][val]))
```

Output:



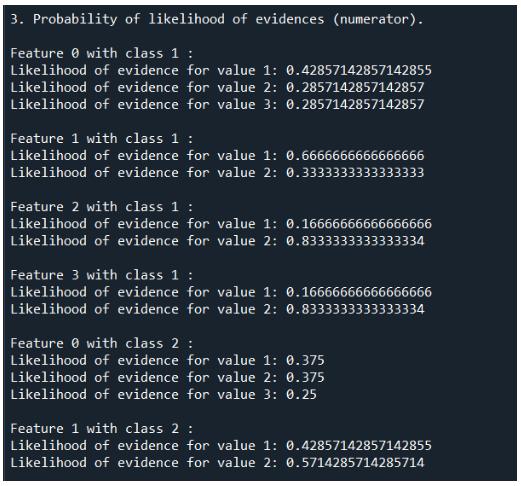
```
2. Probability of evidence.

Probability of evidence for unique values of feature 0 is:
 Value: 1 ; Probabilty: 0.3333333333333333
 Value: 2 ; Probabilty: 0.3333333333333333
 Value: 3 ; Probabilty: 0.3333333333333333

Probability of evidence for unique values of feature 1 is:
 Value: 1 ; Probabilty: 0.5
 Value: 2 ; Probabilty: 0.5

Probability of evidence for unique values of feature 2 is:
 Value: 1 ; Probabilty: 0.5
 Value: 2 ; Probabilty: 0.5

Probability of evidence for unique values of feature 3 is:
 Value: 1 ; Probabilty: 0.5
 Value: 2 ; Probabilty: 0.5
```

3.Calculate probability of likelihood of evidence.



```
    # probability of likelihood of evidences
print("\n\n3. Probability of likelihood of evidences (numerator).")
for featuredata in attributedistribution:
    print("\nFeature {} with class {} :".format(featuredata[1],featuredata[0]))
    for unique in attributedistribution[featuredata].dictionary.keys():
        print("Likelihood of evidence for value {}: {}".format(unique,attributedistribution[featuredata][unique]))
```

Output:



```
3. Probability of likelihood of evidences (numerator).

Feature 0 with class 1 :
Likelihood of evidence for value 1: 0.42857142857142855
Likelihood of evidence for value 2: 0.2857142857142857
Likelihood of evidence for value 3: 0.2857142857142857

Feature 1 with class 1 :
Likelihood of evidence for value 1: 0.6666666666666666
Likelihood of evidence for value 2: 0.3333333333333333

Feature 2 with class 1 :
Likelihood of evidence for value 1: 0.16666666666666666
Likelihood of evidence for value 2: 0.8333333333333334

Feature 3 with class 1 :
Likelihood of evidence for value 1: 0.16666666666666666
Likelihood of evidence for value 2: 0.8333333333333334

Feature 0 with class 2 :
Likelihood of evidence for value 1: 0.375
Likelihood of evidence for value 2: 0.375
Likelihood of evidence for value 3: 0.25

Feature 1 with class 2 :
Likelihood of evidence for value 1: 0.42857142857142855
Likelihood of evidence for value 2: 0.5714285714285714
```

1.2.1 DATA

Naive Bayes Classifier :

- This is probabilistic classifier, the main objective is to process, analyse and categorize the data
- Each feature makes equal contribution to the target class.
- Each feature is independent and equally contributes to the probability
- Processes the training dataset to calculate the conditional probabilities and class probabilities
- Easy to implement and fast computation and preforms well on larger data

There are 3 types of naïve Bayes classifier

- Multinomial Naïve Bayes
- Bernoulli Naïve Bayes
- Gaussian Naïve Bayes

Example :

Climate	Temperature	windy	Play Football
Sunny	Hot	TRUE	Yes
Rainy	Mild	TRUE	No
Rainy	Hot	FALSE	Yes
Sunny	Cool	TRUE	Yes

We can find out whether the day is preferable to play football by classifying the data .The columns are the features and the rows are the individual entriesWe can use the Naïve Bayes classifier to predict the day which is preferable to play football using Bayes theorem

Probability :

Calculate a specific event occurred ,given certain of attempts

Probability = events / outcomes

Conditional Independence

We have a dataset with set of class (A) and we want to classify an item with a set of features (F)

Conditional probability given to the (F) : P(A/F) = P(F/A) * P(A) / P(F)

Will have to do this process for every class :

The calculation of the conditional probability then depends on the calculation of the following:
The probability of **Class(A,B,C)** in the dataset.
The conditional probability of each feature occurring in an item classified in **Class**.
The probabilities of each individual feature.

Bayes theorem

Naïve Bayes is classification algorithm : to predict the class membership of unclassified data

P(A/B) = P(B/A) * P(A) / P(B)

P(A/B) : Conditional probability : probability of A given B

P(B/A) : Conditional probability : probability of B given A

P(A) and P(B) : Probability of A and B (A and B are events

The conditional Probability is depends on :

- The probability of class in a dataset
- The conditional probability of each feature from class
- The probability of each feature

References :

https://github.com/aimacode/aima-python

https://www.sciencedirect.com/topics/engineering/naive-bayes-classifier

https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c