# Question 1.1.1

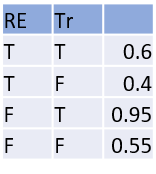
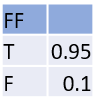
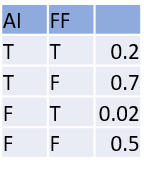
The prior probabilities for all the questions is as follows.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | never | rarely | sometimes | often | always | SUM |
| Tip 1 | STUDY | 0.021739 | 0.086957 | 0.130435 | 0.26087 | 0.5 | 1 |
| Tip 2 | REST | 0.352941 | 0.117647 | 0.352941 | 0.117647 | 0.058824 | 1 |
| Tip 3 | ALARM | 0.615385 | 0.051282 | 0.128205 | 0.102564 | 0.102564 | 1 |

It is not possible to compute the joint distribution as we do not have any data of how they occur together.

# Question 1.1.2

With the supplied variables, the following Bayesian Network was created, conditional probability tables are represented in the diagram:



## Assumptions made:

1. Artificial intelligence can help boost renewable energy by offering smart ways to load balance between durations of high/low production and high/low demand
2. AI and automation does have an impact on jobs lost
3. Low Fossil Fuel prices results in lower usage of renewable energy and vice versa
4. Low Fossil Fuel prices also means that people will travel more and not use public transport, resulting in more traffic
5. High traffic causes more global warming
6. Not using renewable energy also has an impact on global warming
7. Global warming results in climate change which results in unemployment in various sectors (especially farm related)

Querying the Bayesian Network:

Inference is the process of getting the probability distribution of some variables given a set of assignments to to other variables. There are two methods – enumeration or elimination. Both the methods give same results but the latter is faster than the former. Below is the example of enumeration and elimination in practice with AIMA library:

def Q1\_1\_2():

T, F = True, False

bayes\_net = BayesNet([

('AI', '', 0.8),

('FossilFuel', '', 0.4),

('RenewableEnergy', 'AI FossilFuel',

{(T, T): 0.2, (T,F):0.7, (F,T):0.02, (F,F):0.5}),

('Traffic', 'FossilFuel',

{T:0.95, F:0.1}),

('GlobalWarming', 'RenewableEnergy Traffic',

{(T,T):0.6, (T,F):0.4, (F,T):0.95, (F,F):0.55}),

('Employed', 'AI GlobalWarming',

{(T,T):0.01, (T,F):0.03, (F,T):0.03, (F,F):0.95})

])

*#print(bayes\_net.variable\_node('GlobalWarming').cpt)*

p\_employed = enumeration\_ask(X='Employed',

e={'AI': True,

'FossilFuel':True},

bn=bayes\_net)

print('-' \* 80)

print(f"given AI=true and FossilFuel=True:",

f"\n\t\tP(Employed)\t\t=\t\t{p\_employed.show\_approx()}")

print('-' \* 80)

p\_global\_warming = elimination\_ask(X='GlobalWarming',

e={'Employed':False,

'Traffic':False}, bn=bayes\_net)

print('-' \* 80)

print(f"Given Employed=False and Traffic=False, \n\t\tP(GlobalWarming)\t=",

f"\t\t{p\_global\_warming.show\_approx()}")

print('-' \* 80)

p\_ai = elimination\_ask(X='AI', e={'RenewableEnergy': True,

'GlobalWarming': True,

'Employed': True,

'Traffic':True,

'FossilFuel':True}, bn=bayes\_net)

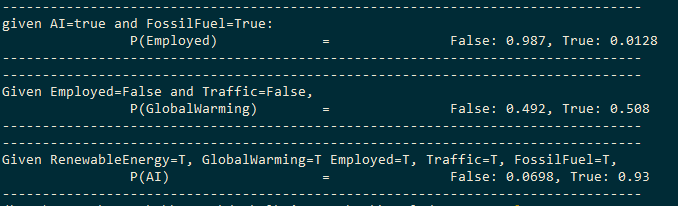
print('-' \* 80)

print(f"Given RenewableEnergy=T, GlobalWarming=T",

f"Employed=T, Traffic=T, FossilFuel=T, \n\t\tP(AI)\t\t\t=\t\t{p\_ai.show\_approx()}")

print('-' \* 80)

The results are as follows:

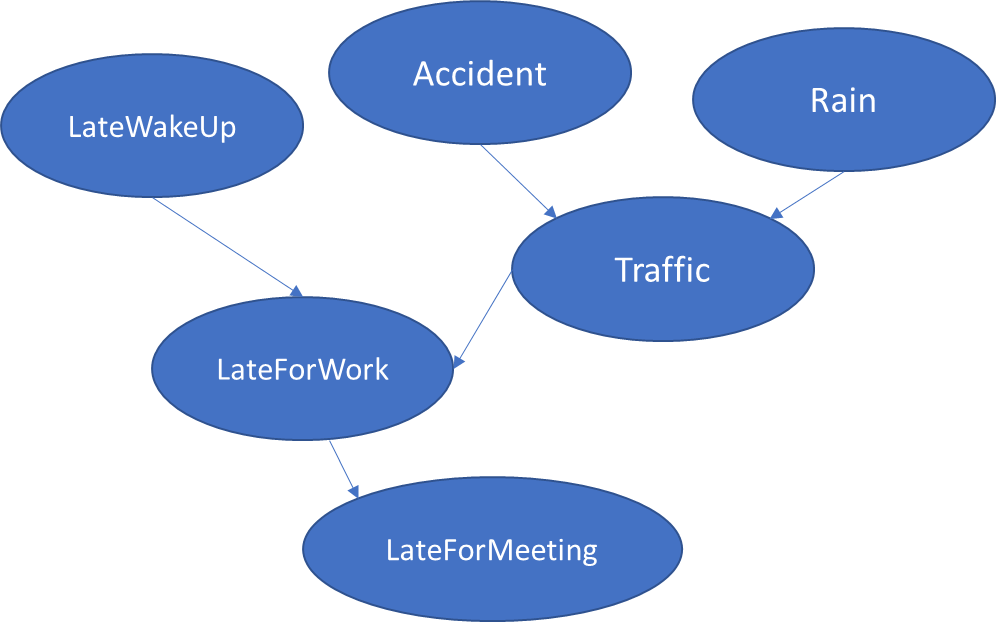


## Explanation of Bayesian Networks

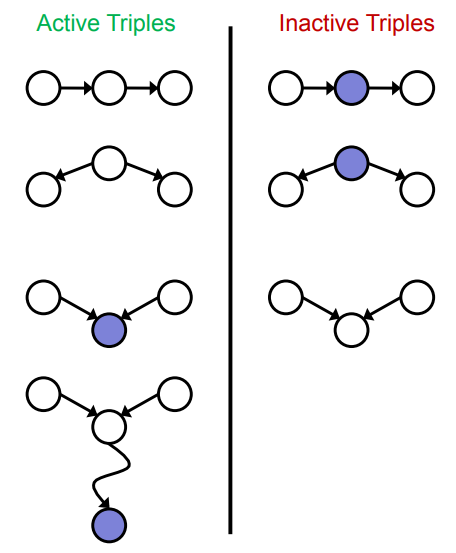
The best prediction with Bayesian Learning is given by Bayes Optimal Classifier, but Bayes Optimal classifier quickly becomes intractable as it requires computing the full joint probability distribution table. Such a table has O(dN) entries if there are N variables with d values each. Naïve Bayes (NB) classifier gets around this problem by making a naïve assumption that all the N variables are independent. While NB works well for some classes of problems, in other classes of problems, NB’s assumption of independence is inadequate. Bayes Networks (BN) takes the middle path between Bayes Optimal and NB classifiers. A BN is a ***graphical model that efficiently represents the joint probability distribution table*.** A BN is a Directed Acyclic Graph comprises of:

1. Nodes that represent each Random Variable
2. Arcs that represent probabilistic dependence between nodes. Absence of an arc denotes independence or conditional independence
3. Every node has a Conditional Probability Table (CPT) where the conditional probability given its parents is stored. This conditional probability table only contains the conditional probability given its immediate parents and no other nodes.

Conditional dependence and independence can be leveraged for efficient inference.

***Each node is conditionally independent of its non-descendants given the value of its parents. Consider the given network***. Normally, LateForMeeting is related to LateWakeUp, but if one is given that LateForWork=T, then LateForMeeting becomes conditionally independent of LateWakeUp. These notions of conditional dependence and independence can be leveraged for efficient storage and retrieval.

In general, inference in a BN is also intractable, but for some special BNs it becomes tractable, and even for BNs where it is not tractable, there are efficient ways of approximating inference using sampling.

D-Separation is the criteria to determine if X and Y are conditionally independent given the evidence; all undirected paths between X and Y are considered, and if all of them are inactive, then they are conditionally independent.

A path is inactive if all triplets in that path are active, so even if one of the triplets are inactive, then it is inactive. This often leads to cut-offs and thereby greater conditional independence.

Active and inactive triplets can be recognized by the forms on the left, where the purple circles represent the evidence and white circles the hidden nodes.

Figure from CS188 Slides

Another advantage of BN is that it allows humans to create the graphical representation of the network using domain knowledge. It also allows an intuitive representation of the relationships between RVs, and it is easy to judge what influences another at a glance.

One model is where humans give the graph structure, and the CPTs can be learned from the data. This is often used in practice. Another variant used is where the graph structure itself can be learned from the data, but that often produces graphs that are not intuitive to humans.