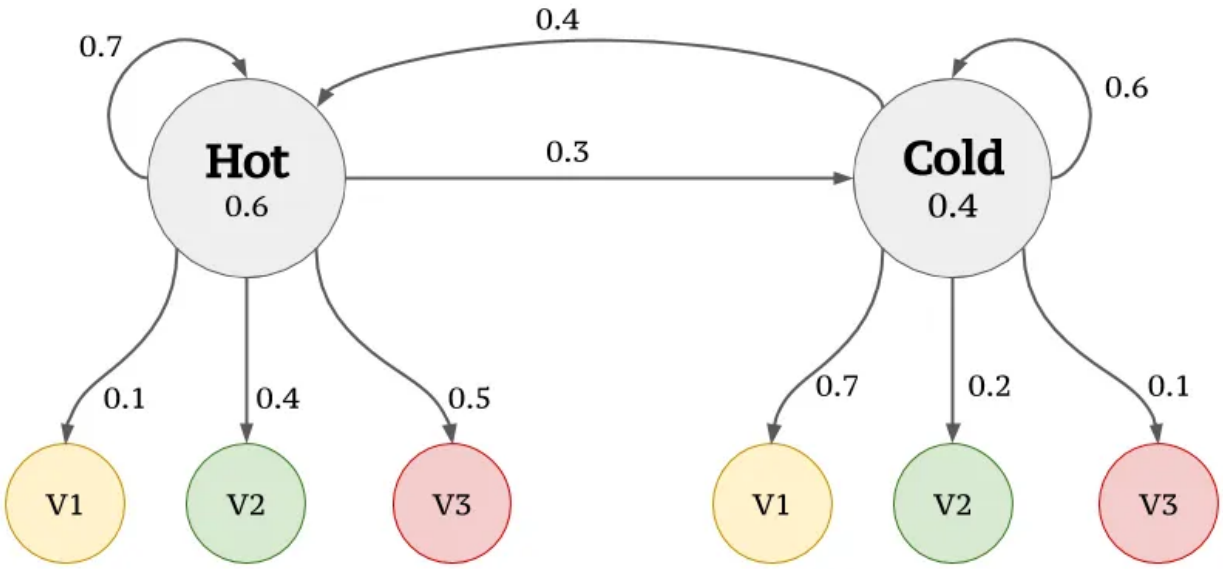
# EXPERIMENT NO. 6

## EXAMPLE




## CODE

```
from hmmlearn import hmm
import numpy as np
import math
states = ('Hot', 'Cold')
n_states = len(states)
observations = ('v1','v2','v3')
start_probability = {'Hot': 0.6, 'Cold': 0.4}
transition_probability = {
  'Hot' : {'Hot': 0.7, 'Cold' : 0.3},
  'Cold' : {'Hot' : 0.4, 'Cold' : 0.6}
  }
emission_probability = {
  'Hot' : {'v1' : 0.1, 'v2' : 0.4,'v3':0.5},
  'Cold' : {'v1' : 0.7, 'v2' : 0.2,'v3':0.1},
  }
```

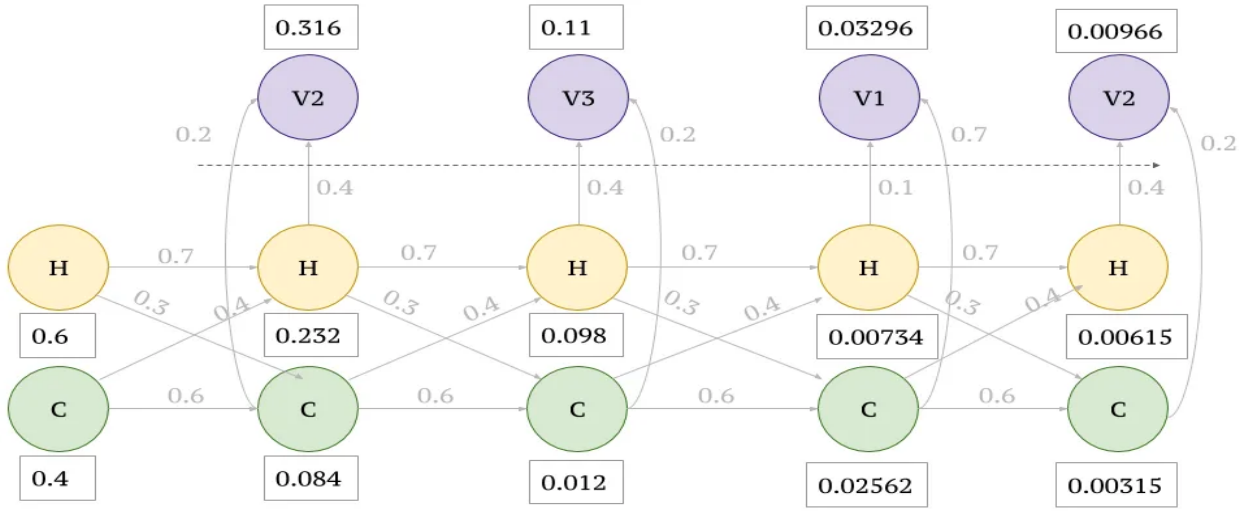
```
model = hmm.CategoricalHMM(n_components=n_states)
model.startprob_ = np.array([0.6, 0.4])
model.transmat_ = np.array([[0.7,0.3],
                            [0.4,0.6],])
model.emissionprob_ = np.array([[0.1, 0.4,0.5],
                                [0.7, 0.2,0.1]])

print("Probability of sequence {v2, v3, v1, v2} :
",math.exp(model.score(np.array([[1,2,0,1]]))), end='\n')
observations_sequence = np.array([1,2,0,1]).reshape(-1, 1)
hidden_states= model.predict(observations_sequence)
print(hidden_states)
```

## OUTPUT

```
Probability of sequence {v2, v3, v1, v2} :  0.009834399999999998
```

```
[0 0 1 0]
```

Verifying output with solved example-




Here, final probability is 0.0097
Best path is H,H,C,H (hidden_states)