Author:

- **Sidharth**
- *2K18/ MC/ 114*

Objective:

Demonstrate Markov Chain Process. WAP to implement Markov Chain special cases

1. To find steady state probabilities in case of ergodic Markov Chain.
2. To find that the specific state in a Markov chain is a recurrent or transient.

Theory:

A Markov chain is a mathematical system that experiences transitions from one state to another according to certain probabilistic rules. The defining characteristic of a Markov chain is that no matter how the process arrived at its present state, the possible future states are fixed. In other words, the probability of transitioning to any particular state is dependent solely on the current state and time elapsed. The state space, or set of all possible states, can be anything: letters, numbers, weather conditions, baseball scores, or stock performances.

Code & Output:

1. To find steady state probabilities in case of ergodic Markov Chain.

```
import numpy as np
Arr = np.array([[0,0.5,0.5,0],[0.25,0,0.5,0.25],
                [0.6,0.4,0,0],[0.3,0.4,0.3,0]])
n = 4
print("Transition Probability Matrix: ")
print(Arr)
Arr = Arr.T
for i in range(0,n):
    Arr[i][i] -= 1
    Arr[0][i] = 1
B = [0]*n
B[0] = 1
z = np.linalg.solve(Arr,B)
print("\nSteady State Probability Distribution given:")
print(z)
```

```
Transition Probability Matrix:
[[0.   0.5  0.5  0.  ]
 [0.25 0.   0.5  0.25]
 [0.6  0.4  0.   0.  ]
 [0.3  0.4  0.3  0.  ]]

Steady State Probability Distribution given:
[0.29353779 0.30668127 0.32311062 0.07667032]
```

2. To find that the specific state in a Markov chain is a recurrent or transient.

```
import numpy as np
m = np.array([[1, 0 , 0 , 0, 0], [0 , 1, 0 ,0, 0] , [0 ,0 ,0.5, 0.
              [0 , 0.5, 0, 0.5, 0] , [0.25, 0.25, 0 , 0, 0.5]])
print("Transition Probability Matrix: ")
print(m)
transient=[]
recurrent=[]
row, col = m.shape
for i in range(row):
    flag=True
    for j in range(col):
        if m[i][j]>0:
            if m[j][i]==0:
                flag=False
                transient.append(i)
                break
    if flag :
        recurrent.append(i)
print("\nTransient State:", transient)
print("Recurrent State:", recurrent)
```

```
Transition Probability Matrix:
[[1.   0.   0.   0.   0.  ]
 [0.   1.   0.   0.   0.  ]
 [0.   0.   0.5  0.5  0.  ]
 [0.   0.5  0.   0.5  0.  ]
 [0.25 0.25 0.   0.   0.5 ]]

Transient State: [2, 3, 4]
Recurrent State: [0, 1]
```

Result:

Implemented Markov Chain special cases to find steady state probabilities in case of ergodic Markov Chain and to find that the specific state in a Markov chain is a recurrent or transient.

Discussion:

The probability of transitioning to any particular state is dependent solely on the current state and time elapsed. The state space, or set of all possible states, can be anything: letters, numbers, weather conditions, baseball scores, or stock performances.