```
pip install hmmlearn
```

```
Collecting hmmlearn
  Downloading hmmlearn-0.3.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (161 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 161.1/161.1 kB 2.8 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.10 in /usr/local/lib/python3.10/dist-packages (from hmmlearn) (1.25.2)
Requirement already satisfied: scikit-learn!=0.22.0,>=0.16 in /usr/local/lib/python3.10/dist-packages (from hmmlearn) (1.2.2)
Requirement already satisfied: scipy>=0.19 in /usr/local/lib/python3.10/dist-packages (from hmmlearn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn!=0.22.0,>=0.16->hmmlearn
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn!=0.22.0,>=0.16->hm
Installing collected packages: hmmlearn
Successfully installed hmmlearn-0.3.2
```

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from hmmlearn import hmm
```

```
states = ["Sunny", "Rainy"]
n_states = len(states)
print('Number of hidden states :',n_states)
observations = ["Dry", "Wet"]
n_observations = len(observations)
print('Number of observations  :',n_observations)
```

```
Number of hidden states : 2
Number of observations  : 2
```

```
state_probability = np.array([0.6, 0.4])
print("State probability: ", state_probability)
transition_probability = np.array([[0.7, 0.3],
                                   [0.3, 0.7]])
print("\nTransition probability:\n", transition_probability)
emission_probability= np.array([[0.9, 0.1],
                                [0.2, 0.8]])
print("\nEmission probability:\n", emission_probability)
```

```
State probability:  [0.6 0.4]

Transition probability:
 [[0.7 0.3]
 [0.3 0.7]]

Emission probability:
 [[0.9 0.1]
 [0.2 0.8]]
```

```
model = hmm.CategoricalHMM(n_components=n_states)
model.startprob_ = state_probability
model.transmat_ = transition_probability
model.emissionprob_ = emission_probability
```

```
observations_sequence = np.array([0, 1, 0, 1, 0, 0]).reshape(-1, 1)
observations_sequence
```

```
array([[0],
       [1],
       [0],
       [1],
       [0],
       [0]])
```

```
hidden_states = model.predict(observations_sequence)
print("Most likely hidden states:", hidden_states)
```

```
Most likely hidden states: [0 1 1 1 0 0]
```

```
log_probability, hidden_states = model.decode(observations_sequence,
                                              lengths = len(observations_sequence),
                                              algorithm ='viterbi' )

print('Log Probability :',log_probability)
print("Most likely hidden states:", hidden_states)
```

```
Log Probability : -6.360602626270058
Most likely hidden states: [0 1 1 1 0 0]
```

```
sns.set_style("whitegrid")
plt.plot(hidden_states, '-o', label="Hidden State")
plt.xlabel('Time step')
plt.ylabel('Most Likely Hidden State')
plt.title("Sunny or Rainy")
plt.legend()
plt.show()
```