### Experiment – No-10

### *Problem Statement :* Given an example of weather conditions and playing sports. You need to calculate the probability of playing sports. Now, you need to classify whether players will play or not, based on the weather condition.

### 

***Program:***

weather =['sunny','sunny','overcast','rainy','rainy','rainy','overcast','sunny','rainy','sunny','overcast','overcast','rainy']

temp = ['hot','hot','hot','mild','cool','cool','cool','mild','cool','mild','mild','hot','hot']

play = ['no','no','yes','yes','no','yes','no','yes','yes','no','yes','no','no']

from sklearn import preprocessing

le = preprocessing.LabelEncoder()

weather\_encoded = le.fit\_transform(weather)

print(weather\_encoded)

type(weather\_encoded)

temp\_encoded = le.fit\_transform(temp)

play\_encoded = le.fit\_transform(play)

print(temp\_encoded)

print(play\_encoded)

features = list(zip(weather\_encoded,temp\_encoded))

print(features)

from sklearn.naive\_bayes import GaussianNB

model = GaussianNB()

model.fit(features,play\_encoded)

predicted = model.predict([[0,2]])

print("Predicted Value = ",predicted)

### *Output :*

[2 2 0 1 1 1 0 2 2 1 2 0 0 1]

Temp: [1 1 1 2 0 0 0 2 0 2 2 2 1 2]

Play: [0 0 1 1 1 0 1 0 1 1 1 1 1 0]

[(2, 1), (2, 1), (0, 1), (1, 2), (1, 0), (1, 0), (0, 0), (2, 2), (2, 0), (1, 2), (2, 2), (0, 2), (0, 1), (1, 2)]

Predicted Value: [1]