```
weather=['sunny','sunny','overcast','rainy','rainy','rainy','overcast','sunny','sunny','rainy','sunny','overcast','overcast','rainy']
temp=['hot','hot','mild','cool','cool','cool','mild','cool','mild','mild','mild','mild','mild']
play=['no','no','yes','yes','no','yes','yes','yes','yes','yes','yes','no']
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
weather_encoded=le.fit_transform(weather)
print(weather_encoded)
     [2 2 0 1 1 1 0 2 2 1 2 0 0 1]
temp_encoded=le.fit_transform(temp)
label=le.fit_transform(play)
print("Temp:",temp_encoded)
print("Play:",label)
     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]
features=[tup for tup in zip(weather_encoded,temp_encoded)]
print(features)
     [(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)]
from sklearn.naive_bayes import GaussianNB
model=GaussianNB()
model.fit(features,label)
     ▼ GaussianNB
     GaussianNB()
predicted=model.predict([[0,2]])
print("Predicted Value:",predicted)
     Predicted Value: [1]
from sklearn import datasets
wine = datasets.load_wine()
print("Features: ",wine.feature_names)
print("\nlabels: ", wine.target_names)
     labels: ['class_0' 'class_1' 'class_2']
wine.data.shape
     (178, 13)
print(wine.data[0:5])
     [[1.423e+01 1.710e+00 2.430e+00 1.560e+01 1.270e+02 2.800e+00 3.060e+00
      2.800e-01 2.290e+00 5.640e+00 1.040e+00 3.920e+00 1.065e+03]
      [1.320e+01 1.780e+00 2.140e+00 1.120e+01 1.000e+02 2.650e+00 2.760e+00
      2.600e-01 1.280e+00 4.380e+00 1.050e+00 3.400e+00 1.050e+03]
      [1.316e+01 2.360e+00 2.670e+00 1.860e+01 1.010e+02 2.800e+00 3.240e+00
      3.000e-01 2.810e+00 5.680e+00 1.030e+00 3.170e+00 1.185e+03]
      [1.437e+01 1.950e+00 2.500e+00 1.680e+01 1.130e+02 3.850e+00 3.490e+00
      2.400e-01 2.180e+00 7.800e+00 8.600e-01 3.450e+00 1.480e+03]
      [1.324e+01 2.590e+00 2.870e+00 2.100e+01 1.180e+02 2.800e+00 2.690e+00
      3.900e-01 1.820e+00 4.320e+00 1.040e+00 2.930e+00 7.350e+02]]
print(wine.target)
```

```
from sklearn.model_selection import train_test_split
 \textbf{X\_train, X\_test, y\_train, y\_test = train\_test\_split(wine.data,wine.target,test\_size=0.3,random\_state=109) } 
from \ sklearn.naive\_bayes \ import \ GaussianNB
gnb=GaussianNB()
gnb.fit(X_train,y_train)
      ▼ GaussianNB
     GaussianNB()
y_pred=gnb.predict(X_test)
print("Y predicted values: ", y_pred)
     Y predicted values: [0 0 1 2 0 1 0 0 1 0 2 2 2 2 0 1 1 0 0 1 2 1 0 2 0 0 1 2 1 1 0 1 1 0
      2 2 0 2 1 0 0 0 2 2 0 1 1 2 0 0 2]
from sklearn import metrics
print("Accuracy: ", metrics.accuracy_score(y_test, y_pred))
     Accuracy: 0.9074074074074
n = input("Enter a postive number")
a=0
b=1
if ()
```