```
[Sunny, Mild, Normal, False], [Sunny, Cool, High, False]
import numpy as np
import pandas as pd
from sklearn import preprocessing
from sklearn.naive_bayes import GaussianNB

df = pd.read_csv('golf.csv')
#print(df)
outlook = np.array(df.iloc[:, 0])
temp = np.array(df.iloc[:,1])
humidity = np.array(df.iloc[:,2])
windy = np.array(df.iloc[:,3])
y = np.array(df.iloc[:, 4])
le = preprocessing.LabelEncoder()
outlook_encoded=np.array(le.fit_transform(outlook))
temp_encoded=np.array(le.fit_transform(temp))
humidity_encoded=np.array(le.fit_transform(humidity))
windy_encoded=np.array(le.fit_transform(windy))
x = [tup for tup in zip(outlook_encoded, temp_encoded, humidity_encoded, windy_encoded)]
y=np.array(le.fit_transform(y))
model = GaussianNB().fit(x,y)
pred=[1, 1, 0, 1],[2, 2, 1, 0], [2, 0, 0, 0]
pred=np.array(pred)
predict1=model.predict([pred[0]])
predict2=model.predict([pred[1]])
predict3=model.predict([pred[2]])
print(f'predict inputs after being enumerated are {pred}')
print(f'If 0 it means no, and 1 means yes')
print(f'{predict1}')
print(f'{predict2}')
print(f'{predict3}')
```

```
raulrodriguez@Rauls-Air WorkSpaceVSPython % /usr/local/bi
ML.py"
predict inputs after being enumerated are [[1 1 0 1]
 [2 2 1 0]
 [2 0 0 0]]
If 0 it means no, and 1 means yes
[0]
[1]
[0]
raulrodriguez@Rauls-Air WorkSpaceVSPython %
```