In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [2]:

```
tennis=[
    ["Sunny","Hot","High","Weak","No"],
    ["Sunny","Hot","High","Strong","No"],
    ["Overcast","Hot","High","Weak","Yes"],
    ["Rain","Mild","High","Weak","Yes"],
    ["Rain","Cool","Normal","Weak","Yes"],
    ["Rain","Cool","Normal","Strong","No"],
    ["Overcast","Cool","Normal","Strong","Yes"],
    ["Sunny","Mild","High","Weak","No"],
    ["Sunny","Cool","Normal","Weak","Yes"],
    ["Rain","Mild","Normal","Weak","Yes"],
    ["Sunny","Mild","Normal","Strong","Yes"],
    ["Overcast","Mild","High","Strong","Yes"],
    ["Overcast","Hot","Normal","Weak","Yes"],
    ["Rain","Mild","High","Strong","No"]
]

df=pd.DataFrame(tennis, columns=['Outlook', 'Temp', 'Humidity', 'Wind', 'Decision'])
```

In [3]:

```
print(df)
```

```
     Outlook  Temp Humidity    Wind Decision
0      Sunny   Hot     High    Weak       No
1      Sunny   Hot     High  Strong       No
2   Overcast   Hot     High    Weak      Yes
3       Rain  Mild     High    Weak      Yes
4       Rain  Cool   Normal    Weak      Yes
5       Rain  Cool   Normal  Strong       No
6   Overcast  Cool   Normal  Strong      Yes
7      Sunny  Mild     High    Weak       No
8      Sunny  Cool   Normal    Weak      Yes
9       Rain  Mild   Normal    Weak      Yes
10     Sunny  Mild   Normal  Strong      Yes
11  Overcast  Mild     High  Strong      Yes
12  Overcast   Hot   Normal    Weak      Yes
13      Rain  Mild     High  Strong       No
```

In [4]:

```
type(df)
```

Out[4]:

```
pandas.core.frame.DataFrame
```

In [5]:

```
df.replace({"Outlook":{"Sunny":0, "Overcast":1, "Rain":2},
            "Temp": {"Hot":0, "Mild":1, "Cool":2},
            "Humidity": {"High":0, "Normal":1},
            "Wind": {"Weak":0, "Strong":1},
            "Decision": {"No":0, "Yes":1}
            }, inplace=True)
```

In [6]:

```
print(df)
```

```
    Outlook  Temp  Humidity  Wind  Decision
0         0     0         0     0         0
1         0     0         0     1         0
2         1     0         0     0         1
3         2     1         0     0         1
4         2     2         1     0         1
5         2     2         1     1         0
6         1     2         1     1         1
7         0     1         0     0         0
8         0     2         1     0         1
9         2     1         1     0         1
10        0     1         1     1         1
11        1     1         0     1         1
12        1     0         1     0         1
13        2     1         0     1         0
```

In [7]:

```
x=df.drop(columns=['Decision'])
print(x)
```

```
    Outlook  Temp  Humidity  Wind
0         0     0         0     0
1         0     0         0     1
2         1     0         0     0
3         2     1         0     0
4         2     2         1     0
5         2     2         1     1
6         1     2         1     1
7         0     1         0     0
8         0     2         1     0
9         2     1         1     0
10        0     1         1     1
11        1     1         0     1
12        1     0         1     0
13        2     1         0     1
```

In [8]:

```
y=df['Decision']
y
```

Out[8]:

```
0     0
1     0
2     1
3     1
4     1
5     0
6     1
7     0
8     1
9     1
10    1
11    1
12    1
13    0
Name: Decision, dtype: int64
```

In [9]:

```
from sklearn.model_selection import train_test_split
```

In [10]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state
```

In [11]:

```
print(x.shape,x_train.shape,x_test.shape,y.shape,y_train.shape,y_test.shape)
```

```
(14, 4) (10, 4) (4, 4) (14,) (10,) (4,)
```

In [12]:

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
```

In [13]:

```
nb.fit(x_train, y_train)
```

Out[13]:

```
GaussianNB
GaussianNB()
```

In [14]:

```
y_pred=nb.predict(x_test)
y_pred=list(y_pred)
print(y_pred)
```

```
[0, 0, 1, 0]
```

In [15]:

```
from sklearn.metrics import confusion_matrix, accuracy_score
```

In [16]:

```
cm=confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[0 0]
 [3 1]]
```

In [17]:

```
acc=accuracy_score(y_pred,y_test)
print(acc)
```

```
0.25
```

In [18]:

```
MSE=sum((y_test-y_pred)**2)/len(y_test)
print("Mean Square Error: ",MSE)

MAE=sum((abs(y_test-y_pred))**2)/len(y_test)
print("Mean Square Error: ",MAE)
```

```
Mean Square Error:  0.75
Mean Square Error:  0.75
```

In [ ]:

```

```