In [1]:

```
import pandas as pd
from sklearn import tree
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB

data = pd.read_csv('playtennis.csv')
print("The first 5 Values of data is : \n", data.head())

X = data.iloc[:, :-1]
print("\nThe First 5 values of the train data is\n", X.head())

y= data.iloc[:, -1]
print("\nThe First 5 values of train output is\n", y.head())

le_outlook = LabelEncoder()
X.Outlook = le_outlook.fit_transform(X.Outlook)

le_Temperature = LabelEncoder ()
X. Temperature = le_Temperature.fit_transform(X.Temperature)

le_Humidity = LabelEncoder()
X. Humidity = le_Humidity.fit_transform(X.Humidity)

le_Windy=LabelEncoder()
X.Windy = le_Windy.fit_transform(X.Windy)

print("\nNow the Train output is\n", X.head())

le_PlayTennis=LabelEncoder()
y= le_PlayTennis.fit_transform(y)
print("\nNow the Train output is\n",y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

classifier = GaussianNB ()

classifier.fit(X_train, y_train)

from sklearn.metrics import accuracy_score
print("Accuracy is:", accuracy_score (classifier.predict (X_test), y_test))
```

```
The first 5 Values of data is :
     Outlook Temperature Humidity   Windy Playtennis
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

The First 5 values of the train data is
     Outlook Temperature Humidity   Windy
0     sunny         hot     high    weak
1     sunny         hot     high  strong
2  overcast         hot     high    weak
3      rain        mild     high    weak
4      rain        cool   normal    weak

The First 5 values of train output is
 0     no
1     no
2    yes
3    yes
4    yes
Name: Playtennis, dtype: object

Now the Train output is
    Outlook  Temperature  Humidity  Windy
0        2            1         0      1
1        2            1         0      0
2        0            1         0      1
3        1            2         0      1
4        1            0         1      1

Now the Train output is
 [0 0 1 1 1 0 1 0 1 1 1 1 1 0]
Accuracy is: 0.6666666666666666
```

```
C:\Users\Venu HK\AppData\Local\Temp\ipykernel_10848\12185859.py:16: SettingWithCopyWarni
ng:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  X.Outlook = le_outlook.fit_transform(X.Outlook)
C:\Users\Venu HK\AppData\Local\Temp\ipykernel_10848\12185859.py:19: SettingWithCopyWarni
ng:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  X. Temperature = le_Temperature.fit_transform(X.Temperature)
C:\Users\Venu HK\AppData\Local\Temp\ipykernel_10848\12185859.py:22: SettingWithCopyWarni
ng:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  X. Humidity = le_Humidity.fit_transform(X.Humidity)
C:\Users\Venu HK\AppData\Local\Temp\ipykernel_10848\12185859.py:25: SettingWithCopyWarni
ng:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  X.Windy = le_Windy.fit_transform(X.Windy)
```

In [ ]: