

# LAB 03 Task1

## Naive Bayes classifier

In [30]:

```
import numpy as np
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.naive_bayes import GaussianNB, MultinomialNB
```

In [31]:

```
import pandas as pd
data = pd.read_csv('D:\SEM7\ML\ML_Lab\Lab_3\Dataset3.csv')
print("\nData :\n", data)
```

Data :

	Outlook	Temp	Wind	Humidity	Class
0	R	H	F	1	0
1	R	H	T	2	0
2	O	H	F	1	1
3	R	M	F	1	1
4	S	C	F	1	1
5	O	C	T	0	0
6	O	C	T	1	1
7	R	M	F	1	0
8	O	C	F	0	1
9	S	M	F	2	1
10	R	C	T	2	0
11	O	M	T	0	1
12	O	H	F	1	1
13	S	M	T	1	0

In [32]:

```
#Data Partitioning
a = data.iloc[:, :-1].values
b = data.iloc[:, -1].values

print("Input : ")
print(a)
print("Output: ")
print(b)
```

Input :

```
[['R' 'H' 'F' 1]
 ['R' 'H' 'T' 2]
 ['O' 'H' 'F' 1]
 ['R' 'M' 'F' 1]
 ['S' 'C' 'F' 1]
 ['O' 'C' 'T' 0]
 ['O' 'C' 'T' 1]
 ['R' 'M' 'F' 1]
 ['O' 'C' 'F' 0]
 ['S' 'M' 'F' 2]
 ['R' 'C' 'T' 2]
 ['O' 'M' 'T' 0]
 ['O' 'H' 'F' 1]
 ['S' 'M' 'T' 1]]
```

Output:

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

## Creating LabelEncoder for Outlook,Temp ,Wind

In [35]:

```
#Label Encoding on columns
label = LabelEncoder()
Outlook=label.fit_transform(a[ : ,0])
a[ : ,0]=Outlook
print(Outlook)
temp=label.fit_transform(a[ : ,1])
a[ : ,1]=temp
print(temp)
Wind=label.fit_transform(a[ : ,2])
a[ : ,2]=Wind
print(Wind)
```

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

In [37]:

```
#Combining Features
features=tuple(zip(Outlook,temp,Wind,a[ : ,3]))
print("Features:",features)
```

```
Features: ((1, 1, 0, 1), (1, 1, 1, 2), (0, 1, 0, 1), (1, 2, 0, 1), (2, 0, 0,
1), (0, 0, 1, 0), (0, 0, 1, 1), (1, 2, 0, 1), (0, 0, 0, 0), (2, 2, 0, 2),
(1, 0, 1, 2), (0, 2, 1, 0), (0, 1, 0, 1), (2, 2, 1, 1))
```

In [38]:

```
model=MultinomialNB()  
model.fit(features,b)
```

Out[38]:

MultinomialNB()

In [39]:

```
#What will be the value of Play, if Outlook is 'Rainy', Temperature is 'Mild', Humidity ='N  
predicted_value= model.predict([[1,2,0,1]])  
print("Predicted Value:", predicted_value)
```

Predicted Value: [1]

In [40]:

```
#What will be the value of Play, if Outlook is 'Sunny', Temperature is 'Cool', Humidity ='H  
predicted_value= model.predict([[2,0,1,2]])  
print("Predicted Value:", predicted_value)
```

Predicted Value: [0]

In [45]:

```
from sklearn.model_selection import train_test_split  
data_training, data_testing, target_training, target_testing = train_test_split(a, b, test_
```

In [46]:

```
gnb = GaussianNB()  
gnb.fit(data_training, target_training)  
target_predict = gnb.predict(data_testing)
```

In [47]:

```
from sklearn import metrics  
print("Accuracy:",metrics.accuracy_score(target_testing, target_predict))
```

Accuracy: 0.7142857142857143

In [48]:

```
from sklearn.metrics import precision_score  
from sklearn.metrics import recall_score  
precision = precision_score(target_testing, target_predict)  
recall = recall_score(target_testing, target_predict)  
print('precision: {}'.format(precision))  
print('recall: {}'.format(recall))
```

precision: 1.0  
recall: 0.6

In [ ]: