

## ## 6.

Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.

In [2]:

```
import numpy as np
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

In [6]:

```
play_tennis = pd.read_csv("Pgm 6 PlayTennis.csv")
play_tennis.head(2)
```

Out[6]:

	Outlook	Temperature	Humidity	Wind	Play Tennis
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No

In [9]:

```
number = LabelEncoder()
play_tennis['Outlook'] = number.fit_transform(play_tennis['Outlook'])
play_tennis['Temperature'] = number.fit_transform(play_tennis['Temperature'])
play_tennis['Humidity'] = number.fit_transform(play_tennis['Humidity'])
play_tennis['Wind'] = number.fit_transform(play_tennis['Wind'])
play_tennis['Play Tennis'] = number.fit_transform(play_tennis['Play Tennis'])

play_tennis.head()
```

Out[9]:

	Outlook	Temperature	Humidity	Wind	Play Tennis
0	2	1	0	1	0
1	2	1	0	0	0
2	0	1	0	1	1
3	1	2	0	1	1
4	1	0	1	1	1

In [11]:

```
features = ["Outlook", "Temperature", "Humidity", "Wind"]
target = "Play Tennis"

XTrain, XTest, YTrain, YTest = train_test_split(play_tennis[features], play_tennis[target],
```

In [14]:

```
model = GaussianNB()
model.fit(XTrain, YTrain)

pred = model.predict(XTest)
accuracy = accuracy_score(YTest, pred)
print("Accuracy : ", accuracy)
```

Accuracy : 0.8

In [20]:

```
print(model.predict([[2,1,0,0]]))
```

[0]