LAB 3

Task 1 : Try the algo on Dataset3 - LabelEncoding of features:and Train test Division 95%-5%

In [1]:
```
import pandas as pd
from sklearn import preprocessing
from sklearn.metrics import confusion_matrix,precision_score,recall_score,ac
curacy_score
from sklearn.naive_bayes import GaussianNB,MultinomialNB
from sklearn.model_selection import train_test_split

datasets = pd.read_csv('/home/nihar/Desktop/SEM 7/ML/Lab/Lab3/Dataset3.csv')
```

In [2]:
```
#creating labelEncoder
le = preprocessing.LabelEncoder()

# Converting string labels into numbers.
outlook_encoded=le.fit_transform(datasets['Outlook'])
outlook_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))
print("\nOutllok mapping:",outlook_name_mapping)
print("Outlook :" ,outlook_encoded)

temp_encoded=le.fit_transform(datasets['Temp'])
temp_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))
print("\nTemprature mapping:",temp_name_mapping)
print("Temprature :" ,outlook_encoded)

wind_encoded=le.fit_transform(datasets['Wind'])
wind_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))
print("\nWind mapping:",wind_name_mapping)
print("Wind :" ,wind_encoded)
```

```
Outllok mapping: {'O': 0, 'R': 1, 'S': 2}
Outlook : [1 1 0 1 2 0 0 1 0 2 1 0 0 2]

Temprature mapping: {'C': 0, 'H': 1, 'M': 2}
Temprature : [1 1 0 1 2 0 0 1 0 2 1 0 0 2]

Wind mapping: {'F': 0, 'T': 1}
Wind : [0 1 0 0 0 1 1 0 0 0 1 1 0 1]
```

In [3]:
```
#Combinig outlook, temprature, wind and humidity into single listof tuples

features=tuple(zip(outlook_encoded,temp_encoded,wind_encoded,datasets['Humid
ity']))
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 [4]:
```
#split data set into train and test sets

x_train, x_test, y_train, y_test = train_test_split(features,
                         datasets['Class'], test_size = 0.05, random_state =
129)
print(x_test)
```

```
[(0, 2, 1, 0)]
```

In [5]:
```
#Create a Classifier
mnb=MultinomialNB()

# Train the model using the training sets
mnb.fit(x_train,y_train)

#Predict the response for test dataset
y_pred = mnb.predict(x_test)
print(y_pred)
```

```
[0]
```

In [6]:
```
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",accuracy_score(y_test, y_pred))

print("\nConfusion Matrix :")
confusion_matrix(y_test, y_pred)
```

```
Accuracy: 0.0

Confusion Matrix :
```

Out[6]:
```
array([[0, 0],
       [1, 0]])
```

In [7]:
```
precision = precision_score(y_test, y_pred)
print('\nprecision: {}'.format(precision))

recall = recall_score(y_test, y_pred)
print('\nrecall: {}'.format(recall))
```

```
precision: 0.0

recall: 0.0

/home/nihar/.local/lib/python3.6/site-packages/sklearn/metrics/_classificati
on.py:1221: UndefinedMetricWarning: Precision is ill-defined and being set t
o 0.0 due to no predicted samples. Use `zero_division` parameter to control
this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [8]:
```
# (1) What will be the value of Play, if Outlook is 'Rainy', Temperature is
  'Mild', Humidity ='Normal', and Wind = 'False'?
print(mnb.predict([[1,2,0,1]]))

# (2) What will be the value of Play, if Outlook is 'Sunny', Temeprature is
  'Cool', Humidity ='High', and Wind = 'True'?
print(mnb.predict([[2,0,1,2]]))
```

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