Machine Learning

**Ex:09 Naive Bayes**

**Name:** Athithraja. R

**Reg.no:** 2022503702

1. Apply a naive bayes classification algorithm and predict whether the customer buy the car or not using the described input instances in the sheetCode:

**Code:**

import pandas as pd

from sklearn.naive\_bayes import CategoricalNB

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix

import numpy as np

data = pd.read\_excel('car\_buy\_dataset.xlsx')

print("Actual Data: \n")

print(data)

label\_encoder = LabelEncoder()

data['Age'] = label\_encoder.fit\_transform(data['Age'])

data['Income'] = label\_encoder.fit\_transform(data['Income'])

data['Martial Status'] = label\_encoder.fit\_transform(data['Martial Status'])

data['Credit Rating'] = label\_encoder.fit\_transform(data['Credit Rating'])

data['Buy Car'] = label\_encoder.fit\_transform(data['Buy Car'])

print("\n\nAfter Label Encoding: \n"+str(data))

X = data.drop(columns=['Buy Car'])

y = data['Buy Car'].values

model = CategoricalNB()

model.fit(X, y)

y\_pred = model.predict(X)

print("\nActual Output- Car Buy or not: ",y)

print("\nPredicted Output- Car Buy or not: ",y\_pred)

cm=confusion\_matrix(y,y\_pred)

print('Confusion Matrix: \n'+str(cm))

print('Classification Report: '+classification\_report(y,y\_pred))

test=np.array([1,2,0,1])

test\_df = pd.DataFrame([test], columns=X.columns)

testout = model.predict(test\_df)

print('Input: ',test)

print('Class label: ',testout)

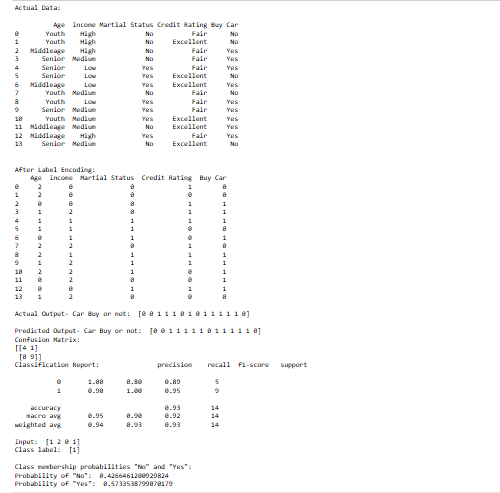
probabilities = model.predict\_proba([test])

print('\nClass membership probabilities "No" and "Yes": ')

print('Probability of "No": ',probabilities[0][0])

print('Probability of "Yes": ',probabilities[0][1])

**Output:**



1. Forest Fire using Naive:

**Code:**

import pandas as pd

from sklearn.naive\_bayes import GaussianNB

from sklearn.preprocessing import StandardScaler

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import classification\_report

from sklearn.metrics import confusion\_matrix

import numpy as np

data = pd.read\_csv('Forest\_Fire\_Dataset.csv')

print("Actual Data: \n")

print(data)

data['Fire'], unique\_values = pd.factorize(data['Classes'])

inplist=data.columns[:-3]

scale=StandardScaler()

data[inplist]=scale.fit\_transform(data[inplist])

#print(data)

X=data.iloc[:,:-3].values

y=data['Fire'].values

model = GaussianNB()

model.fit(X, y)

y\_pred = model.predict(X)

print("\nActual Output- Fire or Not Fire: ",y)

print("\nPredicted Output- Fire or Not Fire: ",y\_pred)

cm=confusion\_matrix(y,y\_pred)

print('\nConfusion Matrix: \n'+str(cm))

print('\nClassification Report: '+classification\_report(y,y\_pred))

**Output:**

