Reg No: 23MCA1030 Vinayak Kumar Singh

```
import pandas as pd

data = pd.read_csv("/content/Dataset.csv")
print(data)
```

```
        age  income student credit_rating Class: buys_computer
0      <=30    high      no          fair                   no
1      <=30    high      no     excellent                   no
2   31...40    high      no          fair                  yes
3       >40  medium      no          fair                  yes
4       >40     low     yes          fair                  yes
5       >40     low     yes     excellent                   no
6   31...40     low     yes     excellent                  yes
7      <=30  medium      no          fair                   no
8      <=30     low     yes          fair                  yes
9       >40  medium     yes          fair                  yes
10     <=30  medium     yes     excellent                  yes
11  31...40  medium      no     excellent                  yes
12  31...40    high     yes          fair                  yes
13      >40  medium      no     excellent                   no
```

```
y = data['Class: buys_computer']
x = data.drop('Class: buys_computer', axis=1)

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
x.age=le.fit_transform(data['age'])
x.income=le.fit_transform(data['income'])
x.student=le.fit_transform(data['student'])
x.credit_rating=le.fit_transform(data['credit_rating'])
print(x)
```

```
    age  income  student  credit_rating
0     1       0        0              1
1     1       0        0              0
2     0       0        0              1
3     2       2        0              1
4     2       1        1              1
5     2       1        1              0
6     0       1        1              0
7     1       2        0              1
8     1       1        1              1
9     2       2        1              1
10    1       2        1              0
11    0       2        0              0
12    0       0        1              1
13    2       2        0              0
```

```
from sklearn.naive_bayes import GaussianNB
model=GaussianNB()
model.fit(x,y)
predicted=model.predict([[1,2,1,1]])
print("predicted Value",predicted)
```

```
predicted Value ['yes']
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439:
UserWarning: X does not have valid feature names, but GaussianNB was
fitted with feature names
  warnings.warn(
```

```
x=data.drop('Class: buys_computer',axis=1)
from sklearn.preprocessing import OneHotEncoder
le=OneHotEncoder()
y = pd.get_dummies(data, columns =
['age','income','student','credit_rating'])
print(y)
```

```
   Class: buys_computer  age_31...40  age_<=30  age_>40
income_high  \
0                    no            0         1        0            1

1                    no            0         1        0            1

2                   yes            1         0        0            1

3                   yes            0         0        1            0

4                   yes            0         0        1            0

5                    no            0         0        1            0

6                   yes            1         0        0            0

7                    no            0         1        0            0

8                   yes            0         1        0            0

9                   yes            0         0        1            0

10                  yes            0         1        0            0

11                  yes            1         0        0            0

12                  yes            1         0        0            1

13                   no            0         0        1            0


    income_low  income_medium  student_no  student_yes  \
0            0              0           1            0
1            0              0           1            0
2            0              0           1            0
3            0              1           1            0
4            1              0           0            1
5            1              0           0            1
6            1              0           0            1
7            0              1           1            0
8            1              0           0            1
9            0              1           0            1
10           0              1           0            1
11           0              1           1            0
12           0              0           0            1
13           0              1           1            0

    credit_rating_excellent  credit_rating_fair
0                         0                   1
1                         1                   0
2                         0                   1
3                         0                   1
4                         0                   1
5                         1                   0
6                         1                   0
7                         0                   1
8                         0                   1
9                         0                   1
10                        1                   0
11                        1                   0
12                        0                   1
13                        1                   0
```

```
x=data.drop('Class: buys_computer',axis=1)
from sklearn.preprocessing import OneHotEncoder
le=OneHotEncoder()
enc_data =
pd.DataFrame(le.fit_transform(data[['age','income','student','credit_r
ating']]))

New_df = data.join(enc_data)
print(enc_data)
print(New_df)
```

```
                                                    0
0     (0, 1)\t1.0\n  (0, 3)\t1.0\n  (0, 6)\t1.0\n ...
1     (0, 1)\t1.0\n  (0, 3)\t1.0\n  (0, 6)\t1.0\n ...
2     (0, 0)\t1.0\n  (0, 3)\t1.0\n  (0, 6)\t1.0\n ...
3     (0, 2)\t1.0\n  (0, 5)\t1.0\n  (0, 6)\t1.0\n ...
4     (0, 2)\t1.0\n  (0, 4)\t1.0\n  (0, 7)\t1.0\n ...
5     (0, 2)\t1.0\n  (0, 4)\t1.0\n  (0, 7)\t1.0\n ...
6     (0, 0)\t1.0\n  (0, 4)\t1.0\n  (0, 7)\t1.0\n ...
```

```
7     (0, 1)\t1.0\n  (0, 5)\t1.0\n  (0, 6)\t1.0\n ...
8     (0, 1)\t1.0\n  (0, 4)\t1.0\n  (0, 7)\t1.0\n ...
9     (0, 2)\t1.0\n  (0, 5)\t1.0\n  (0, 7)\t1.0\n ...
10    (0, 1)\t1.0\n  (0, 5)\t1.0\n  (0, 7)\t1.0\n ...
11    (0, 0)\t1.0\n  (0, 5)\t1.0\n  (0, 6)\t1.0\n ...
12    (0, 0)\t1.0\n  (0, 3)\t1.0\n  (0, 7)\t1.0\n ...
13    (0, 2)\t1.0\n  (0, 5)\t1.0\n  (0, 6)\t1.0\n ...
        age  income student credit_rating Class: buys_computer  \
0      <=30    high      no          fair                   no
1      <=30    high      no     excellent                   no
2   31...40    high      no          fair                  yes
3       >40  medium      no          fair                  yes
4       >40     low     yes          fair                  yes
5       >40     low     yes     excellent                   no
6   31...40     low     yes     excellent                  yes
7      <=30  medium      no          fair                   no
8      <=30     low     yes          fair                  yes
9       >40  medium     yes          fair                  yes
10     <=30  medium     yes     excellent                  yes
11  31...40  medium      no     excellent                  yes
12  31...40    high     yes          fair                  yes
13      >40  medium      no     excellent                   no

                                                    0
0     (0, 1)\t1.0\n  (0, 3)\t1.0\n  (0, 6)\t1.0\n ...
1     (0, 1)\t1.0\n  (0, 3)\t1.0\n  (0, 6)\t1.0\n ...
2     (0, 0)\t1.0\n  (0, 3)\t1.0\n  (0, 6)\t1.0\n ...
3     (0, 2)\t1.0\n  (0, 5)\t1.0\n  (0, 6)\t1.0\n ...
4     (0, 2)\t1.0\n  (0, 4)\t1.0\n  (0, 7)\t1.0\n ...
5     (0, 2)\t1.0\n  (0, 4)\t1.0\n  (0, 7)\t1.0\n ...
6     (0, 0)\t1.0\n  (0, 4)\t1.0\n  (0, 7)\t1.0\n ...
7     (0, 1)\t1.0\n  (0, 5)\t1.0\n  (0, 6)\t1.0\n ...
8     (0, 1)\t1.0\n  (0, 4)\t1.0\n  (0, 7)\t1.0\n ...
9     (0, 2)\t1.0\n  (0, 5)\t1.0\n  (0, 7)\t1.0\n ...
10    (0, 1)\t1.0\n  (0, 5)\t1.0\n  (0, 7)\t1.0\n ...
11    (0, 0)\t1.0\n  (0, 5)\t1.0\n  (0, 6)\t1.0\n ...
12    (0, 0)\t1.0\n  (0, 3)\t1.0\n  (0, 7)\t1.0\n ...
13    (0, 2)\t1.0\n  (0, 5)\t1.0\n  (0, 6)\t1.0\n ...
```

```python
X=data.drop(['Class: buys_computer'],axis=1)
y=data['Class: buys_computer']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.25,random_state = 0)
from sklearn.preprocessing import OneHotEncoder
cols = ['age','income','student','credit_rating']
encoder = OneHotEncoder(sparse_output=False)
X_train = encoder.fit_transform(X_train[cols])
X_test = encoder.transform(X_test[cols])
```

```
from sklearn.naive_bayes import GaussianNB
model_gnb = GaussianNB()
model_gnb.fit(X_train, y_train)
y_pred = model_gnb.predict(X_test)
from sklearn.metrics import accuracy_score
print(100*accuracy_score(y_test, y_pred))
print()
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print('Confusion Matrix')
print(cm)
```

```
50.0

Confusion Matrix
[[0 0]
 [2 2]]
```