

In [2]:

```
#importing required libraries
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder

#reading the dataset
df = pd.read_csv("output.csv")
df.head()
```

Out[2]:

	age	job	marital	education	smoker	monthlyincome	houseowner	loan	contact
0	30	unemployed	married	primary	no	6787	no	no	cellphone
1	33	services	married	secondary	no	9789	yes	yes	cellphone
2	35	services	single	Graduate	no	6350	yes	no	cellphone
3	59	blue-collar	married	secondary	no	5000	yes	no	unknown
4	36	self-employed	married	Graduate	no	5307	yes	no	cellphone

In [3]:

```
#Label encoding for categorical variables
labelencoder = LabelEncoder()
df['job'] = labelencoder.fit_transform(df['job'].astype(str))
df['marital'] = labelencoder.fit_transform(df['marital'].astype(str))
df['education'] = labelencoder.fit_transform(df['education'].astype(str))
df['smoker'] = labelencoder.fit_transform(df['houseowner'].astype(str))
df['loan'] = labelencoder.fit_transform(df['loan'].astype(str))
df['houseowner'] = labelencoder.fit_transform(df['houseowner'].astype(str))
df['contact'] = labelencoder.fit_transform(df['contact'].astype(str))
df['target_buy'] = labelencoder.fit_transform(df['target_buy'].astype(str))
df.head()
```

Out[3]:

	age	job	marital	education	smoker	monthlyincome	houseowner	loan	contact	MOD	mo
0	30	11	1	1	0	6787	0	0	0	79	
1	33	8	1	2	1	9789	1	1	0	220	
2	35	8	2	0	1	6350	1	0	0	185	
3	59	1	1	2	1	5000	1	0	2	226	
4	36	7	1	0	1	5307	1	0	0	341	

In [4]:

```
#data separation
features = df.iloc[:, :-1].values
labels = df.iloc[:, -1].values
print("Features\n", features[:5, :], '\n')
print("Labels\n", labels[:5])
```

Features

```
[[ 30  11  1  1  0 6787  0  0  0  79 6915]
 [ 33  8  1  2  1 9789  1  1  0 220 10049]
 [ 35  8  2  0  1 6350  1  0  0 185 6587]
 [ 59  1  1  2  1 5000  1  0  2 226 5193]
 [ 36  7  1  0  1 5307  1  0  0 341 5326]]
```

Labels

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

In [5]:

```
#onehotencoding
onehotencoder = OneHotEncoder(categorical_features = [1])
features = onehotencoder.fit_transform(features).toarray()
features = features[:, 1:] #avoiding the dummy variable trap
onehotencoder = OneHotEncoder(categorical_features = [13])
features = onehotencoder.fit_transform(features).toarray()
features = features[:, 1:] #avoiding the dummy variable trap
onehotencoder = OneHotEncoder(categorical_features = [15])
features = onehotencoder.fit_transform(features).toarray()
features = features[:, 1:] #avoiding the dummy variable trap
onehotencoder = OneHotEncoder(categorical_features = [22])
features = onehotencoder.fit_transform(features).toarray()
features = features[:, 1:] #avoiding the dummy variable trap

print(features[0, :])
```

```
[ 0.00000000e+00  0.00000000e+00  1.00000000e+00  0.00000000e+00
 0.00000000e+00  1.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
 0.00000000e+00  1.00000000e+00  0.00000000e+00  3.00000000e+01
 0.00000000e+00  6.78700000e+03  0.00000000e+00  0.00000000e+00
 7.90000000e+01  6.91500000e+03]
```

In [6]:

```
#splitting the dataset into train and test set
x_train, x_test, y_train, y_test = train_test_split(features, labels, test_size = 0.3, random_state = 0)
print('x_train shape: ', x_train.shape)
print('x_test shape: ', x_test.shape)
print('y_train shape: ', y_train.shape)
print('y_test shape: ', y_test.shape)
```

```
x_train shape: (3150, 26)
x_test shape: (1350, 26)
y_train shape: (3150,)
y_test shape: (1350,)
```

In []:

In []: