In [6]:

```
import numpy as np
import pandas as pd
```

In [7]:

```
dataset=pd.read_csv(r"C:\Users\kotha\Downloads\bank.csv")
```

In [8]:

```
dataset.head()
```

Out[8]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month
0	59	admin.	married	secondary	no	2343	yes	no	unknown	5	may
1	56	admin.	married	secondary	no	45	no	no	unknown	5	may
2	41	technician	married	secondary	no	1270	yes	no	unknown	5	may
3	55	services	married	secondary	no	2476	yes	no	unknown	5	may
4	54	admin.	married	tertiary	no	184	no	no	unknown	5	may

In [9]:

```
dataset.isnull().any()
```

Out[9]:

```
age          False
job          False
marital      False
education    False
default      False
balance      False
housing      False
loan         False
contact      False
day          False
month        False
duration     False
campaign     False
pdays        False
previous     False
poutcome     False
deposit      False
dtype: bool
```

In [10]:

```
dataset.head(1)
```

Out[10]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	dur
0	59	admin.	married	secondary	no	2343	yes	no	unknown	5	may	

In [11]:

```
print(dataset["job"].unique())
print(dataset["marital"].unique())
print(dataset["education"].unique())
print(dataset["default"].unique())
print(dataset["housing"].unique())
print(dataset["loan"].unique())
print(dataset["contact"].unique())
print(dataset["month"].unique())
print(dataset["poutcome"].unique())
```

```
['admin.' 'technician' 'services' 'management' 'retired' 'blue-collar'
 'unemployed' 'entrepreneur' 'housemaid' 'unknown' 'self-employed'
 'student']
['married' 'single' 'divorced']
['secondary' 'tertiary' 'primary' 'unknown']
['no' 'yes']
['yes' 'no']
['no' 'yes']
['unknown' 'cellular' 'telephone']
['may' 'jun' 'jul' 'aug' 'oct' 'nov' 'dec' 'jan' 'feb' 'mar' 'apr' 'sep']
['unknown' 'other' 'failure' 'success']
```

In [12]:

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
dataset["job"]=le.fit_transform(dataset["job"])
dataset["marital"]=le.fit_transform(dataset["marital"])
dataset["education"]=le.fit_transform(dataset["education"])
dataset["default"]=le.fit_transform(dataset["default"])
dataset["housing"]=le.fit_transform(dataset["housing"])
dataset["loan"]=le.fit_transform(dataset["loan"])
dataset["contact"]=le.fit_transform(dataset["contact"])
dataset["month"]=le.fit_transform(dataset["month"])
dataset["poutcome"]=le.fit_transform(dataset["poutcome"])
dataset["deposit"]=le.fit_transform(dataset["deposit"])
```

In [13]:

```
dataset.head(5)
```

Out[13]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration
0	59	0	1	1	0	2343	1	0	2	5	8	1042
1	56	0	1	1	0	45	0	0	2	5	8	1467
2	41	9	1	1	0	1270	1	0	2	5	8	1389
3	55	7	1	1	0	2476	1	0	2	5	8	579
4	54	0	1	2	0	184	0	0	2	5	8	673

In [21]:

```
x=dataset.iloc[:,0:16].values
y=dataset.iloc[:,16:17].values
```

In [22]:

```
x.shape
```

Out[22]:

```
(11162, 16)
```

In [23]:

```
y.shape
```

Out[23]:

```
(11162, 1)
```

In [15]:

```
from sklearn.preprocessing import OneHotEncoder
one=OneHotEncoder()
z=one.fit_transform(x[:,1:2]).toarray()
t=one.fit_transform(x[:,2:3]).toarray()
u=one.fit_transform(x[:,3:4]).toarray()
v=one.fit_transform(x[:,8:9]).toarray()
w=one.fit_transform(x[:,10:11]).toarray()
b=one.fit_transform(x[:,15:16]).toarray()
x=np.delete(x,[1,2,3,8,10,12,15],axis=1)
x=np.concatenate((b,w,v,u,t,z,x),axis=1)
```

In [16]:

```
z
```

Out[16]:

```
array([[1., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 1., 0., 0.],
       ...,
       [0., 0., 0., ..., 1., 0., 0.],
       [0., 0., 0., ..., 1., 0., 0.],
       [0., 0., 0., ..., 1., 0., 0.]])
```

In [17]:

```
x.shape
```

Out[17]:

```
(11162, 47)
```

In [24]:

```
y.shape
```

Out[24]:

```
(11162, 1)
```

In [18]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [19]:

```
x_train.shape
```

Out[19]:

```
(8929, 47)
```

In [25]:

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_train
```

Out[25]:

```
array([[-0.35031555, -0.22657536, -0.32939951, ..., -0.2483547 ,
        -0.48428507, -0.35888264],
       [-0.35031555, -0.22657536, -0.32939951, ..., -0.31759131,
        -0.48428507, -0.35888264],
       [-0.35031555, -0.22657536, -0.32939951, ...,  1.76816153,
        -0.48428507, -0.35888264],
       ...,
       [-0.35031555, -0.22657536, -0.32939951, ..., -1.03015141,
        -0.48428507, -0.35888264],
       [-0.35031555, -0.22657536, -0.32939951, ..., -0.41856137,
        -0.48428507, -0.35888264],
       [-0.35031555, -0.22657536, -0.32939951, ...,  1.0296377 ,
        -0.48428507, -0.35888264]])
```

In [26]:

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_test=sc.fit_transform(x_test)
x_test
```

Out[26]:

```
array([[ 2.80376731, -0.21765422, -0.31106004, ..., -0.77542752,
         2.62409874,  0.62465525],
       [-0.35666298, -0.21765422,  3.21481343, ..., -0.42011029,
         1.30248101,  1.63872661],
       [-0.35666298, -0.21765422, -0.31106004, ..., -0.54045968,
        -0.46885387, -0.38941612],
       ...,
       [-0.35666298, -0.21765422, -0.31106004, ...,  3.37949177,
        -0.46885387, -0.38941612],
       [-0.35666298, -0.21765422, -0.31106004, ..., -0.88431506,
        -0.46885387, -0.38941612],
       [-0.35666298, -0.21765422, -0.31106004, ..., -0.13356413,
        -0.46885387, -0.38941612]])
```

In [27]:

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
y_train=sc.fit_transform(y_train)
y_train
```

Out[27]:

```
array([[-0.95541123],
       [ 1.04666971],
       [ 1.04666971],
       ...,
       [-0.95541123],
       [-0.95541123],
       [ 1.04666971]])
```

In [28]:

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
y_test=sc.fit_transform(y_test)
y_test
```

Out[28]:

```
array([[-0.92364064],
       [ 1.08267215],
       [-0.92364064],
       ...,
       [ 1.08267215],
       [ 1.08267215],
       [-0.92364064]])
```

In [ ]: