In [81]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [82]:

In [83]:
```
dftrain=pd.read_csv(r"C:\USERS\user\Downloads\C8_loan-train - C8_loan-train.cs
```

Out[83]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coap
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	
...	...	...	...	...	...	...	...	
609	LP002978	Female	No	0	Graduate	No	2900	
610	LP002979	Male	Yes	3+	Graduate	No	4106	
611	LP002983	Male	Yes	1	Graduate	No	8072	
612	LP002984	Male	Yes	2	Graduate	No	7583	
613	LP002990	Female	No	0	Graduate	Yes	4583	

614 rows × 13 columns

In [84]:

Out[84]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [85]:
```
a=dftrain[['Dependents','ApplicantIncome','CoapplicantIncome','Loan_Amount_Ter
```

Out[85]:

	Dependents	ApplicantIncome	CoapplicantIncome	Loan_Amount_Term
0	0	5849	0.0	360.0
1	1	4583	1508.0	360.0
2	0	3000	0.0	360.0
3	0	2583	2358.0	360.0
4	0	6000	0.0	360.0
...	...	...	...	...
609	0	2900	0.0	360.0
610	3+	4106	0.0	180.0
611	1	8072	240.0	360.0
612	2	7583	0.0	360.0
613	0	4583	0.0	360.0

614 rows × 4 columns

In [86]:
```
b=dftrain.head(10)
```

Out[86]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coappli
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	
5	LP001011	Male	Yes	2	Graduate	Yes	5417	
6	LP001013	Male	Yes	0	Not Graduate	No	2333	
7	LP001014	Male	Yes	3+	Graduate	No	3036	
8	LP001018	Male	Yes	2	Graduate	No	4006	
9	LP001020	Male	Yes	1	Graduate	No	12841	

In [95]:
```
a=b[['ApplicantIncome','CoapplicantIncome','Loan_Amount_Term']]
```

Out[95]:

	ApplicantIncome	CoapplicantIncome	Loan_Amount_Term
0	5849	0.0	360.0
1	4583	1508.0	360.0
2	3000	0.0	360.0
3	2583	2358.0	360.0
4	6000	0.0	360.0
5	5417	4196.0	360.0
6	2333	1516.0	360.0
7	3036	2504.0	360.0
8	4006	1526.0	360.0
9	12841	10968.0	360.0

In [97]:
```
c=a.iloc[:,0:3]
```

In [98]:

Out[98]:
```
(10, 3)
```

In [99]:

Out[99]:
```
(10,)
```

In [100]:

In [101]:

In [102]:
```
logr=LogisticRegression()
```

Out[102]:
```
LogisticRegression()
```

In [108]:

In [109]:
```
prediction=logr.predict(observation)
```

Out[109]:
```
array(['N'], dtype=object)
```

In [110]:

Out[110]:
```
array(['N', 'Y'], dtype=object)
```

In [111]:

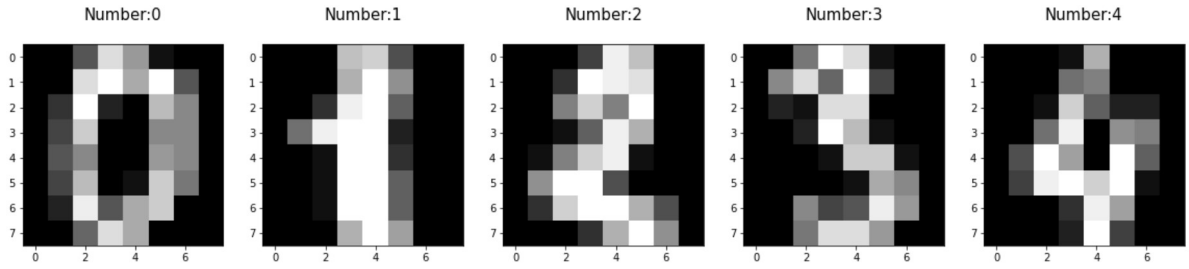
Out[111]:
```
0.7230489707749604
```

In [112]:
```
import re
from sklearn.datasets import load_digits
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
```

In [113]:
```
digits=load_digits()
```

Out[113]:
```
{'data': array([[ 0.,  0.,  5., ...,  0.,  0.,  0.],
        [ 0.,  0.,  0., ..., 10.,  0.,  0.],
        [ 0.,  0.,  0., ..., 16.,  9.,  0.],
        ...,
        [ 0.,  0.,  1., ...,  6.,  0.,  0.],
        [ 0.,  0.,  2., ..., 12.,  0.,  0.],
        [ 0.,  0., 10., ..., 12.,  1.,  0.]]),
 'target': array([0, 1, 2, ..., 8, 9, 8]),
 'frame': None,
 'feature_names': ['pixel_0_0',
  'pixel_0_1',
  'pixel_0_2',
  'pixel_0_3',
  'pixel_0_4',
  'pixel_0_5',
  'pixel_0_6',
  'pixel_0_7',
  'pixel_1_0',
  'pixel_1_1',
```

In [114]:
```
plt.figure(figsize=(20,4))
for index,(image,label) in enumerate(zip(digits.data[0:5],digits.target[0:5])):
    plt.subplot(1,5,index+1)
    plt.imshow(np.reshape(image,(8,8)),cmap=plt.cm.gray)
```



In [115]:
```
```

In [116]:
```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
```

```
(1257, 64)
(540, 64)
(1257,)
(540,)
```

In [117]:
```
logre=LogisticRegression(max_iter=10000)
```

Out[117]:
```
LogisticRegression(max_iter=10000)
```

In [118]:

```
[8 4 5 0 4 3 9 1 6 2 2 9 7 4 3 3 8 7 7 3 2 3 9 0 2 3 0 2 6 5 5 3 8 1 4 8 9
 0 7 7 1 3 6 9 0 3 6 0 4 5 6 8 0 5 4 5 6 2 9 1 3 5 6 1 5 9 2 5 3 3 3 3 3 9
 4 5 5 5 5 2 1 5 2 1 2 6 5 2 9 2 9 0 0 4 2 4 0 0 7 2 2 0 8 4 1 6 1 6 0 3 7
 9 7 5 0 0 5 5 8 9 3 4 6 1 6 2 9 9 1 2 5 5 6 3 9 6 2 6 4 6 6 7 3 3 5 1 7 1
 7 3 4 6 1 2 3 2 1 7 5 3 5 7 3 2 9 5 6 5 0 1 3 2 8 4 3 9 0 2 7 6 2 9 8 0 2
 2 8 1 0 8 1 4 5 5 9 7 3 6 3 1 7 1 4 0 5 3 7 3 0 9 3 8 9 6 8 4 0 5 7 5 4 9
 4 8 2 9 1 6 3 5 8 5 8 5 4 7 4 7 1 9 7 3 2 4 5 1 7 9 2 6 2 1 3 6 5 8 5 0 9
 3 4 6 1 2 2 8 3 3 4 0 4 2 9 6 9 6 8 9 6 8 3 2 2 1 7 2 3 8 7 5 7 9 3 4 6 7
 6 8 1 4 4 4 4 2 1 7 1 7 7 1 9 0 6 1 8 2 1 0 5 9 2 3 5 8 0 9 5 7 3 7 0 5 9
 1 5 5 6 0 2 3 3 4 2 1 0 0 8 2 9 9 2 8 9 9 9 7 2 9 9 8 3 1 0 3 5 2 5 9 1 8
 4 7 1 1 2 8 7 5 7 1 5 6 3 5 5 9 5 1 2 3 1 9 7 4 3 7 2 5 8 7 2 8 2 3 4 3 2
 9 6 1 1 0 7 7 4 8 7 9 1 4 2 6 6 8 1 2 8 1 3 6 5 4 6 0 9 4 1 8 3 4 1 4 8 0
 0 0 2 2 8 2 1 3 1 0 9 4 7 0 8 8 6 2 1 9 7 8 1 0 4 7 7 2 1 8 1 9 9 1 1 5 6
 5 7 0 3 5 5 3 1 0 2 3 9 0 3 6 6 4 6 8 3 6 7 0 5 4 9 7 4 4 8 1 1 3 6 4 0 9
 5 4 0 8 5 8 4 2 6 4 6 2 0 6 9 1 4 6 6 5 4 0]
```

In [119]:

```
0.9555555555555556
```

In [ ]:

In [ ]: