In [55]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [56]:

In [57]:
```
dftrain=pd.read_csv(r"C:\USERS\user\Downloads\C8_loan-test - C8_loan-test.csv"
```

Out[57]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coap
0	LP001015	Male	Yes	0	Graduate	No	5720	
1	LP001022	Male	Yes	1	Graduate	No	3076	
2	LP001031	Male	Yes	2	Graduate	No	5000	
3	LP001035	Male	Yes	2	Graduate	No	2340	
4	LP001051	Male	No	0	Not Graduate	No	3276	
...	...	...	...	...	...	...	...	
362	LP002971	Male	Yes	3+	Not Graduate	Yes	4009	
363	LP002975	Male	Yes	0	Graduate	No	4158	
364	LP002980	Male	No	0	Graduate	No	3250	
365	LP002986	Male	Yes	0	Graduate	No	5000	
366	LP002989	Male	No	0	Graduate	Yes	9200	

367 rows × 12 columns

In [54]:

Out[54]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area'],
      dtype='object')
```

In [58]:
```
a=dftrain[['Dependents','ApplicantIncome','CoapplicantIncome','LoanAmount','Lo
```

Out[58]:

	Dependents	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term
0	0	5720	0	110.0	360.0
1	1	3076	1500	126.0	360.0
2	2	5000	1800	208.0	360.0
3	2	2340	2546	100.0	360.0
4	0	3276	0	78.0	360.0
...	...	...	...	...	...
362	3+	4009	1777	113.0	360.0
363	0	4158	709	115.0	360.0
364	0	3250	1993	126.0	360.0
365	0	5000	2393	158.0	360.0
366	0	9200	0	98.0	180.0

367 rows × 5 columns

In [59]:
```
b=dftrain.head(10)
```

Out[59]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coappli
0	LP001015	Male	Yes	0	Graduate	No	5720	
1	LP001022	Male	Yes	1	Graduate	No	3076	
2	LP001031	Male	Yes	2	Graduate	No	5000	
3	LP001035	Male	Yes	2	Graduate	No	2340	
4	LP001051	Male	No	0	Not Graduate	No	3276	
5	LP001054	Male	Yes	0	Not Graduate	Yes	2165	
6	LP001055	Female	No	1	Not Graduate	No	2226	
7	LP001056	Male	Yes	2	Not Graduate	No	3881	
8	LP001059	Male	Yes	2	Graduate	NaN	13633	
9	LP001067	Male	No	0	Not Graduate	No	2400	

In [60]:
```
a=b[['Dependents','ApplicantIncome','CoapplicantIncome','LoanAmount','Loan_Amo
```

Out[60]:

	Dependents	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term
0	0	5720	0	110.0	360.0
1	1	3076	1500	126.0	360.0
2	2	5000	1800	208.0	360.0
3	2	2340	2546	100.0	360.0
4	0	3276	0	78.0	360.0
5	0	2165	3422	152.0	360.0
6	1	2226	0	59.0	360.0
7	2	3881	0	147.0	360.0
8	2	13633	0	280.0	240.0
9	0	2400	2400	123.0	360.0

In [61]:
```
c=a.iloc[:,0:5]
```

In [62]:

Out[62]:
```
(10, 5)
```

In [63]:

Out[63]:
```
(10,)
```

In [64]:

In [65]:

In [66]:
```
logr=LogisticRegression()
```

Out[66]:
```
LogisticRegression()
```

In [69]:

In [70]:
```
prediction=logr.predict(observation)
```

Out[70]:
```
array(['Rural'], dtype=object)
```

In [71]:

Out[71]:
```
array(['Rural', 'Semiurban', 'Urban'], dtype=object)
```

In [72]:

Out[72]:
```
0.9966637992611229
```

In [73]:

```
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 [74]:

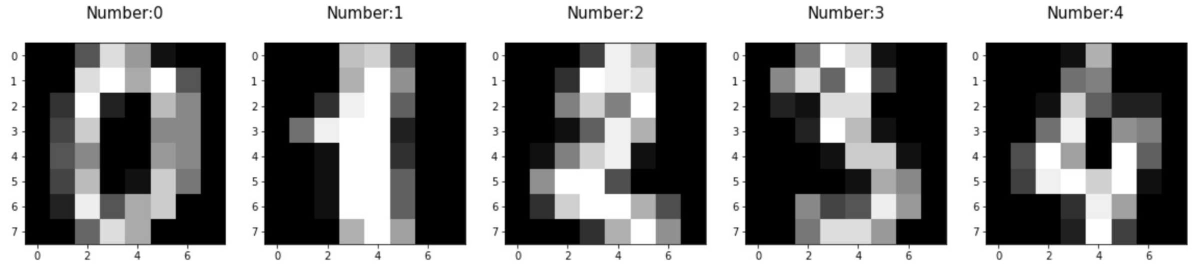
```
digits=load_digits()
```

Out[74]:

```
{'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 [75]:

```
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 [76]:

In [77]:

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
```

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

```
In [78]: logre=LogisticRegression(max_iter=10000)

Out[78]: LogisticRegression(max_iter=10000)

In [79]:

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

In [80]:

0.9537037037037037

In [ ]:
```