import Libraries

In [1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

import Linear Regression

In [2]:

```python
from sklearn.linear_model import LogisticRegression
```

In [3]:

```python
lgr=LogisticRegression()
```

Select Required data from certain columns

In [4]:

```python
a=pd.read_csv("loan.csv")
a
```

Out[4]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	C
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 [5]:

```
c=a.dropna()
c
```

Out[5]:

ried	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount
Yes	1	Graduate	No	4583	1508.0	128.0
Yes	0	Graduate	Yes	3000	0.0	66.0
Yes	0	Not Graduate	No	2583	2358.0	120.0
No	0	Graduate	No	6000	0.0	141.0
Yes	2	Graduate	Yes	5417	4196.0	267.0
...	...	...	...	...	...	...
No	0	Graduate	No	2900	0.0	71.0
Yes	3+	Graduate	No	4106	0.0	40.0
Yes	1	Graduate	No	8072	240.0	253.0
Yes	2	Graduate	No	7583	0.0	187.0
No	0	Graduate	Yes	4583	0.0	133.0

In [6]:

```
c.columns
```

Out[6]:

```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmoun
t',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Statu
s'],
      dtype='object')
```

In [51]:

```
fm=c[['ApplicantIncome',
'Loan_Amount_Term',]]
tv=c[['Credit_History']]
```

Shape

In [52]:

```python
fm.shape
```

Out[52]:

```
(480, 2)
```

In [53]:

```python
tv.shape
```

Out[53]:

```
(480, 1)
```

To make the data in order (feature matrix)

In [54]:

```python
from sklearn.preprocessing import StandardScaler
```

In [55]:

```python
fs=StandardScaler().fit_transform(fm)
```

Imply Logistic Regression

In [56]:

```python
lgr.fit(fm,tv)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63:
DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using r
avel().
  return f(*args, **kwargs)
```

Out[56]:

```
LogisticRegression()
```

Prediction

In [60]:

```python
ab=[[3,90]]
```

In [61]:

```python
pre=lgr.predict(ab)
```

In [62]:

```python
print(pre)
```

```
[1.]
```

To check the output var we have got

In [63]:

```python
lgr.classes_
```

Out[63]:

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

Prediction in Probablity value

In [64]:

```python
lgr.predict_proba(ab)[0][1]
```

Out[64]:

```
0.6173644303036664
```

In [ ]: