In [1]:
```
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
from sklearn.linear_model import LogisticRegression
```

In [3]:
```
# To Import Dataset
sd=pd.read_csv(r"c:\Users\user\Downloads\C8_loan-train.csv")
sd
```

Out[3]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
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 [4]:
```
sd.dropna()
sd
```

Out[4]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
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]:
```
sd.fillna(20)
```

Out[5]:

ried	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Lo
No	0	Graduate	No	5849	0.0	20.0	
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	
...	...	...	...	...	...	...	
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 [9]:
```
feature_matrix = sd[['ApplicantIncome','Loan_Amount_Term']]
target_vector=sd['Credit_History']
```

In [10]:
```
feature_matrix.shape
```

Out[10]:
```
(614, 2)
```

In [11]:
```
target_vector.shape
```

Out[11]:
```
(614,)
```

In [12]:
```
from sklearn.preprocessing import StandardScaler
```

In [13]:
```
fs=StandardScaler().fit_transform(feature_matrix)
```

In [15]:
```
observation =[[1.2,2.3,3.3]]
```

In [17]:
```
logr.predict_proba(observation)[0][0]
```

```
---------------------------------------------------------------------------
NotFittedError                            Traceback (most recent call last)
<ipython-input-17-069118590587> in <module>
----> 1 logr.predict_proba(observation)[0][0]

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py
in predict_proba(self, X)
   1461             where classes are ordered as they are in ``self.classes_`
`.
   1462         """
-> 1463         check_is_fitted(self)
   1464
   1465         ovr = (self.multi_class in ["ovr", "warn"] or

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in inn
er_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64
     65             # extra_args > 0

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in che
ck_is_fitted(estimator, attributes, msg, all_or_any)
   1039
   1040     if not attrs:
-> 1041         raise NotFittedError(msg % {'name': type(estimator).__name_
_})
   1042
   1043

NotFittedError: This LogisticRegression instance is not fitted yet. Call 'fi
t' with appropriate arguments before using this estimator.
```