

```
In [18]: # import Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [19]: from sklearn.linear_model import LogisticRegression
```

```
In [20]: # To Import Dataset
sd=pd.read_csv(r"c:\Users\user\Downloads\C8_loan-test.csv")
sd
```

Out[20]:

| | 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 [21]: sd.dropna()  
sd
```

Out[21]:

| | 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 [22]: sd.fillna(20)
```

Out[22]:

| | 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 [23]: feature_matrix = sd[['ApplicantIncome', 'Loan_Amount_Term']]
target_vector=sd['Credit_History']
```

```
In [24]: feature_matrix.shape
```

```
Out[24]: (367, 2)
```

```
In [25]: target_vector.shape
```

```
Out[25]: (367,)
```

```
In [26]: from sklearn.preprocessing import StandardScaler
```

```
In [27]: fs=StandardScaler().fit_transform(feature_matrix)
```

```
In [28]: observation = [[1.2,2.3,3.3]]
```

```
In [30]: logr.predict_proba(observation)
```

```
-----
NotFittedError                                Traceback (most recent call last)
<ipython-input-30-7c5bc94db2a6> in <module>
----> 1 logr.predict_proba(observation)

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 inner
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 'fit' with appropriate arguments before using this estimator.

In []: