```
In [1]: import pandas as pd
```

```
In [3]: data=pd.read_csv("LoanApprovalPrediction.csv")
```

```
In [5]: data.head()
```

Out[5]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
0	LP001002	Male	No	0.0	Graduate	No	5849	
1	LP001003	Male	Yes	1.0	Graduate	No	4583	
2	LP001005	Male	Yes	0.0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0.0	Not Graduate	No	2583	
4	LP001008	Male	No	0.0	Graduate	No	6000	

```
In [7]: data.isnull().sum()
```

```
Out[7]:
Loan_ID               0
Gender                0
Married               0
Dependents           12
Education             0
Self_Employed         0
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           21
Loan_Amount_Term     14
Credit_History       49
Property_Area         0
Loan_Status           0
dtype: int64
```

```
In [9]: data.fillna(method="ffill",inplace=True)
        data.fillna(method="bfill",inplace=True)
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_10092\1360262160.py:1: FutureWarning: DataFrame.f
illna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or ob
j.bfill() instead.
  data.fillna(method="ffill",inplace=True)
C:\Users\Dell\AppData\Local\Temp\ipykernel_10092\1360262160.py:2: FutureWarning: DataFrame.f
illna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or ob
j.bfill() instead.
  data.fillna(method="bfill",inplace=True)
```

```
In [15]: df=pd.get_dummies(data,columns=["Gender","Married","Dependents","Education","Self_Employed"
```

```
In [17]: df
```

Out[17]:

	Loan_ID	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_Histo
0	LP001002	5849	0.0	128.0	360.0	
1	LP001003	4583	1508.0	128.0	360.0	
2	LP001005	3000	0.0	66.0	360.0	
3	LP001006	2583	2358.0	120.0	360.0	
4	LP001008	6000	0.0	141.0	360.0	
...	...	...	...	...	...	
593	LP002978	2900	0.0	71.0	360.0	
594	LP002979	4106	0.0	40.0	180.0	
595	LP002983	8072	240.0	253.0	360.0	
596	LP002984	7583	0.0	187.0	360.0	
597	LP002990	4583	0.0	133.0	360.0	

598 rows × 23 columns

In [19]:
```
df.dtypes
```

Out[19]:
```
Loan_ID                     object
ApplicantIncome              int64
CoapplicantIncome          float64
LoanAmount                 float64
Loan_Amount_Term           float64
Credit_History             float64
Gender_Female                 bool
Gender_Male                   bool
Married_No                    bool
Married_Yes                   bool
Dependents_0.0                bool
Dependents_1.0                bool
Dependents_2.0                bool
Dependents_3.0                bool
Education_Graduate            bool
Education_Not Graduate        bool
Self_Employed_No              bool
Self_Employed_Yes             bool
Property_Area_Rural           bool
Property_Area_Semiurban       bool
Property_Area_Urban           bool
Loan_Status_N                 bool
Loan_Status_Y                 bool
dtype: object
```

In [45]:
```
df.drop("Gender_Female",axis=1,inplace=True)
df.drop("Married_No",axis=1,inplace=True)
df.drop("Education_Graduate",axis=1,inplace=True)
df.drop("Self_Employed_No",axis=1,inplace=True)
df.drop("Property_Area_Rural",axis=1,inplace=True)
df.drop("Loan_Status_N",axis=1,inplace=True)
df.drop("Dependents_3+",axis=1,inplace=True)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[45], line 1
----> 1 df.drop("Gender_Female",axis=1,inplace=True)
      2 df.drop("Married_No",axis=1,inplace=True)
      3 df.drop("Education_Graduate",axis=1,inplace=True)

File ~\AppData\Roaming\Python\Python312\site-packages\pandas\core\frame.py:5581, in DataFram
e.drop(self, labels, axis, index, columns, level, inplace, errors)
   5433 def drop(
   5434     self,
   5435     labels: IndexLabel | None = None,
   (...)
   5442     errors: IgnoreRaise = "raise",
   5443 ) -> DataFrame | None:
   5444     """
   5445     Drop specified labels from rows or columns.
   5446 
   (...)
   5579             weight  1.0     0.8
   5580     """
-> 5581     return super().drop(
   5582         labels=labels,
   5583         axis=axis,
   5584         index=index,
   5585         columns=columns,
   5586         level=level,
   5587         inplace=inplace,
   5588         errors=errors,
   5589     )

File ~\AppData\Roaming\Python\Python312\site-packages\pandas\core\generic.py:4788, in NDFram
e.drop(self, labels, axis, index, columns, level, inplace, errors)
   4786 for axis, labels in axes.items():
   4787     if labels is not None:
-> 4788         obj = obj._drop_axis(labels, axis, level=level, errors=errors)
   4790 if inplace:
   4791     self._update_inplace(obj)

File ~\AppData\Roaming\Python\Python312\site-packages\pandas\core\generic.py:4830, in NDFram
e._drop_axis(self, labels, axis, level, errors, only_slice)
   4828         new_axis = axis.drop(labels, level=level, errors=errors)
   4829     else:
-> 4830         new_axis = axis.drop(labels, errors=errors)
   4831     indexer = axis.get_indexer(new_axis)
   4833 # Case for non-unique axis
   4834 else:

File ~\AppData\Roaming\Python\Python312\site-packages\pandas\core\indexes\base.py:7070, in I
ndex.drop(self, labels, errors)
   7068 if mask.any():
   7069     if errors != "ignore":
-> 7070         raise KeyError(f"{labels[mask].tolist()} not found in axis")
   7071     indexer = indexer[~mask]
   7072 return self.delete(indexer)

KeyError: "['Gender_Female'] not found in axis"
```

In [47]:
```
x=df.iloc[:,2:15]
y=df.iloc[:,-1]
y
```

```
Out[47]: 0       True
         1      False
         2       True
         3       True
         4       True
                ...
         593     True
         594     True
         595     True
         596     True
         597    False
         Name: Loan_Status_Y, Length: 598, dtype: bool
```

```
In [51]: from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.75,random_state=(0))
```

```
In [55]: from sklearn.linear_model import LogisticRegression
         model=LogisticRegression()
         model.fit(x_train,y_train)
```

```
C:\Users\Dell\anaconda3\Lib\site-packages\sklearn\linear_model\_logistic.py:469: Convergence
Warning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
Out[55]: ▾ LogisticRegression

         LogisticRegression()
```

```
In [65]: y_hat=model.predict(x_test)
```

```
In [67]: model.score(x,y)
```

```
Out[67]: 0.8043478260869565
```

```
In [69]: from sklearn.metrics import accuracy_score
         acc=accuracy_score(y_hat,y_test)
         print(acc)
```

```
0.7933333333333333
```