

```
import pandas as pd
```

```
data=pd.read_csv("/content/loan_data.csv")
```

```
data
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome
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

```
#FIRST FEW ROWS OF THE DATASET
```

```
data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome
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


```
data.info
```

```

<bound method DataFrame.info of
0    LP001002    Male    No    ...    1.0    Urban    Y
1    LP001003    Male    Yes   ...    1.0    Rural    N
2    LP001005    Male    Yes   ...    1.0    Urban    Y
3    LP001006    Male    Yes   ...    1.0    Urban    Y
4    LP001008    Male    No    ...    1.0    Urban    Y
..      ...      ...      ...      ...      ...      ...
609  LP002978    Female   No    ...    1.0    Rural    Y
610  LP002979    Male    Yes   ...    1.0    Rural    Y
611  LP002983    Male    Yes   ...    1.0    Urban    Y
612  LP002984    Male    Yes   ...    1.0    Urban    Y
613  LP002990    Female   No    ...    0.0    Semiurban  N

[614 rows x 13 columns]>

```



```
data.shape
```

```
(614, 13)
```

```
#checking missing values inthe data
```

```
data.isnull().sum()
```

```

Loan_ID           0
Gender            13
Married           3
Dependents        15
Education          0
Self_Employed     32
ApplicantIncome   0
CoapplicantIncome  0
LoanAmount        22
Loan_Amount_Term  14
Credit_History    50
Property_Area      0
Loan_Status        0
dtype: int64

```

```
#checking datatypes
```

```
data.dtypes
```

```

Loan_ID           object
Gender            object
Married           object
Dependents        object
Education          object
Self_Employed     object
ApplicantIncome   int64
CoapplicantIncome float64
LoanAmount        float64
Loan_Amount_Term  float64
Credit_History    float64
Property_Area      object

```

```
Loan_Status      object
dtype: object
```

```
#filling missing values with categorical variable mode
data["Gender"].fillna(data["Gender"].mode()[0],inplace=True)

data['Married'].fillna(data['Married'].mode()[0], inplace=True)

data['Dependents'].fillna(data['Dependents'].mode()[0], inplace=True)

data['Self_Employed'].fillna(data['Self_Employed'].mode()[0], inplace=True)

data['Loan_Amount_Term'].fillna(data['Loan_Amount_Term'].mode()[0], inplace=True)

data['Credit_History'].fillna(data['Credit_History'].mode()[0], inplace=True)
```

```
#filling missing values of continous variable with mean
data["LoanAmount"].fillna(data["LoanAmount"].mean(),inplace=True)
```

```
#checking missing values
data.isnull().sum()
```

```
Loan_ID          0
Gender           0
Married          0
Dependents       0
Education        0
Self_Employed    0
ApplicantIncome  0
CoapplicantIncome 0
LoanAmount       0
Loan_Amount_Term 0
Credit_History  0
Property_Area    0
Loan_Status      0
dtype: int64
```

```
#converting categorical into numbers
data['Gender'] = data['Gender'].map({'Male': 0, 'Female': 1})
data['Married'] = data['Married'].map({'No': 0, 'Yes': 1})
data['Dependents'] = data['Dependents'].map({'0': 0, '1': 1, '2': 2, '3+': 3})
data['Education'] = data['Education'].map({'Graduate': 1, 'Not Graduate': 0})
data['Self_Employed'] = data['Self_Employed'].map({'No': 0, 'Yes': 1})
data['Property_Area'] = data['Property_Area'].map({'Rural': 0, 'Semiurban': 1, 'Urban': 2})
data['Loan_Status'] = data['Loan_Status'].map({'N': 0, 'Y': 1})
```

```
#checking data values
data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome
0	LP001002	0	0	0	1	0	58421
1	LP001003	0	1	1	1	0	45728
2	LP001005	0	1	0	1	1	30492
3	LP001006	0	1	0	0	0	25668
4	LP001008	0	0	0	1	0	60136

```
#data normalaization
```

```
data["Dependents"] = data["Dependents"] - data["Dependents"].min() / data["Dependents"].max() - data["Dependents"].min()
```

```
#using for loop we can convert the all the values in the range between 0 to 1
```

```
for i in data.columns[1::]:
```

```
    data[i] = (data[i] - data[i].min()) / (data[i].max() - data[i].min())
```

```
#checking values
```

```
data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome
0	LP001002	0.0	0.0	0.000000	1.0	0.0	0.07041
1	LP001003	0.0	1.0	0.333333	1.0	0.0	0.05481
2	LP001005	0.0	1.0	0.000000	1.0	1.0	0.03521
3	LP001006	0.0	1.0	0.000000	0.0	0.0	0.03001
4	LP001008	0.0	0.0	0.000000	1.0	0.0	0.07231

```
#saving the proprocessed data
```

```
data.to_csv("Loan_Prediction_New_Data.csv", index=False)
```

✓ 0s completed at 3:27 PM

● ✕