

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

In [81]: data=pd.read_csv("E:/Data Analytics course/ML/train.csv")
data

Out[81]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y
...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	360.0	1.0	Rural	Y
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	180.0	1.0	Rural	Y
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	360.0	1.0	Urban	Y
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	360.0	1.0	Urban	Y
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	360.0	0.0	Semurburb	N

614 rows x 13 columns

```
In [94]: data.head()
```

```
Out[94]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y

```
In [96]: data.describe()
```

```
Out[96]:
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.000000	564.000000
mean	5403.458283	1621.245788	146.412162	342.000000	0.842189
std	6108.041673	2926.248389	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.000000	0.000000
25%	2877.500000	0.000000	100.000000	360.000000	1.000000
50%	3812.500000	1188.000000	128.000000	360.000000	1.000000
75%	5795.000000	2297.250000	188.000000	360.000000	1.000000
max	81006.000000	41667.000000	708.000000	480.000000	1.000000

```
In [97]: data.info()
```

```
Out[97]:
```

```
In [98]: data.isnull().sum()
```

```
Out[98]:
```

```
In [82]: data["LoanAmount"] = data["LoanAmount"].fillna(data["LoanAmount"].mean())
data["Credit_History"] = data["Credit_History"].fillna(data["Credit_History"].median())
data
```

```
Out[82]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	146.412162	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.000000	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.000000	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.000000	360.0	1.0	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.000000	360.0	1.0	Urban	Y
...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.000000	360.0	1.0	Rural	Y
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.000000	180.0	1.0	Rural	Y
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.000000	360.0	1.0	Urban	Y
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.000000	360.0	1.0	Urban	Y
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.000000	360.0	0.0	Semurburb	N

614 rows x 13 columns

```
In [40]: (data.isnull().sum())
```

```
Out[40]:
```

```
In [63]: data.shape
```

```
Out[63]: (614, 13)
```

```
In [43]: data.droptna(inplace=True)
```

```
In [37]: data.shape
```

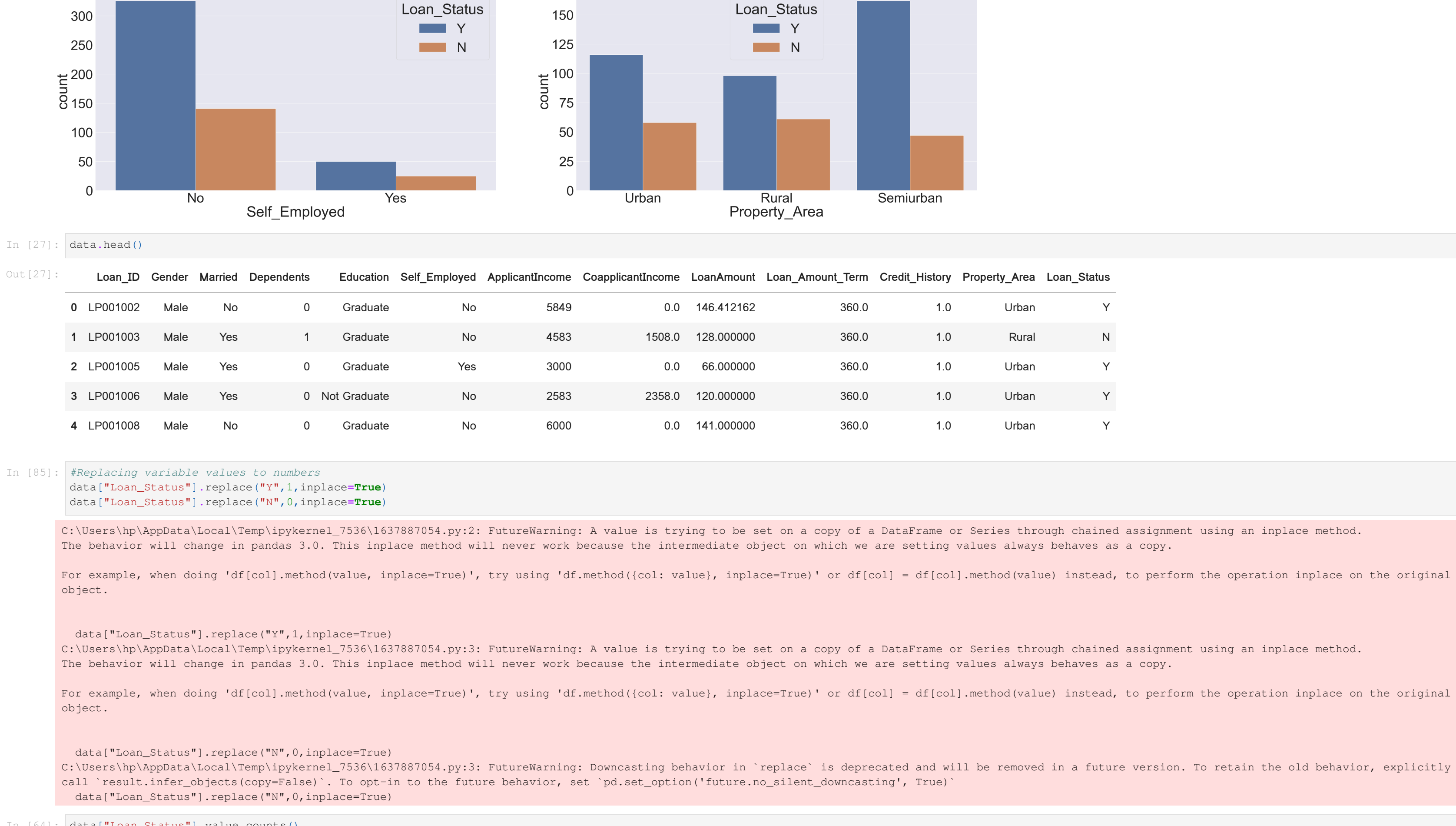
```
Out[37]: (542, 13)
```

```
In [76]: data.isnull().sum()
```

```
Out[76]:
```

```
In [84]: #DATA VISUALIZATION
plt.figure(figsize=(10,5))
sns.set(font_scale=5)
plt.subplot(131)
sns.countplot(x="Gender", hue=data["Loan_Status"], data=data)
plt.subplot(132)
sns.countplot(x="Married", hue=data["Loan_Status"], data=data)
plt.subplot(133)
sns.countplot(x="Education", hue=data["Loan_Status"], data=data)
plt.subplot(134)
sns.countplot(x="Self_Employed", hue=data["Loan_Status"], data=data)
plt.subplot(135)
sns.countplot(x="Property_Area", hue=data["Loan_Status"], data=data)
plt.subplot(136)
sns.countplot(x="Loan_Status", data=data)
```

```
Out[84]:
```



```
In [27]: data.head()
```

```
Out[27]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	146.412162	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.000000	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.000000	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.000000	360.0	1.0	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.000000	360.0	1.0	Urban	Y
...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.000000	360.0	1.0	Rural	Y
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.000000	180.0	1.0	Rural	Y
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.000000	360.0	1.0	Urban	Y
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.000000	360.0	1.0	Urban	Y
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.000000	360.0	0.0	Semurburb	N

```
In [85]: #Replacing variable values to numbers
data["Loan_Status"] = data["Loan_Status"].replace("Y",1,inplace=True)
data["Loan_Status"] = data["Loan_Status"].replace("N",0,inplace=True)

C:\Users\hp\AppData\Local\Temp\ipykernel_7536\1931877054.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.
For example, when doing "df[col].method(value, inplace=True)", try using "df.method(col: value, inplace=True)" or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

data["Loan_Status"] = data["Loan_Status"].replace("Y",1,inplace=True)
C:\Users\hp\AppData\Local\Temp\ipykernel_7536\1931877054.py:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.
For example, when doing "df[col].method(value, inplace=True)", try using "df.method(col: value, inplace=True)" or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

data["Loan_Status"] = data["Loan_Status"].replace("Y",1,inplace=True)
C:\Users\hp\AppData\Local\Temp\ipykernel_7536\1931877054.py:3: FutureWarning: Downcasting behavior in 'replace' is deprecated and will be removed in a future version. To retain the old behavior, explicitly call 'result.infer_objects(copy=False)'. To opt-in to the future behavior, set 'pd.set_option('future.no_silent_downcasting', True)'

data["Loan_Status"] = data["Loan_Status"].replace("Y",1,inplace=True)

In [44]: data["Loan_Status"].value_counts()
```

```
Out[44]:
```

```
In [64]: Loan_Status
0    422
1    192
Name: count, dtype: int64
```

```
In [93]: data.Married = data.Married.map({"Yes":1,"No":0})
```

```
In [94]: data["Married"].value_counts()
```

```
Out[94]:
```

```
In [96]:
```

	Married
1	355
0	187

Name: count, dtype: int64

```
In [71]: data
```

```
Out[71]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y
...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	360.0	1.0	Rural	Y
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	180.0	1.0	Rural	Y
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	360.0	1.0	Urban	Y
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	360.0	1.0	Urban	Y
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	360.0	0.0	Semurburb	N

614 rows x 13 columns

```
In [83]: data
```

```
Out[83]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	146.412162	360.0	1.0	Urban	1
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.000000	360.0	1.0	Rural	0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.000000	360.0	1.0	Urban	1
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.000000	360.0	1.0	Urban	1
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.000000	360.0	1.0	Urban	1
...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.000000	360.0	1.0	Rural	1
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.000000	180.0	1.0	Rural	1
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.000000	360.0	1.0	Urban	1
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.000000	360.0	1.0	Urban	1
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.000000	360.0	0.0	Semurburb	0

542 rows x 13 columns

```
In [84]: data["Gender"] = data["Gender"].replace("Male",1,inplace=True)
data["Gender"] = data["Gender"].replace("Female",0,inplace=True)
data

C:\Users\hp\AppData\Local\Temp\ipykernel_7536\1931877054.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.
For example, when doing "df[col].method(value, inplace=True)", try using "df.method(col: value, inplace=True)" or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

data["Gender"] = data["Gender"].replace("Male",1,inplace=True)
C:\Users\hp\AppData\Local\Temp\ipykernel_7536\1931877054.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.
For example, when doing "df[col].method(value, inplace=True)", try using "df.method(col: value, inplace=True)" or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

data["Gender"] = data["Gender"].replace("Female",0,inplace=True)
C:\Users\hp\AppData\Local\Temp\ipykernel_7536\1931877054.py:2: FutureWarning: Downcasting behavior in 'replace' is deprecated and will be removed in a future version. To retain the old behavior, explicitly call 'result.infer_objects(copy=False)'. To opt-in to the future behavior, set 'pd.set_option('future.no_silent_downcasting', True)'

data["Gender"] = data["Gender"].replace("Female",0,inplace=True)

In [87]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	1	No	0	Graduate	No	5849	0.0	146.412162	360.0	1.0	Urban	1
1	LP001003	1	Yes	1	Graduate	No	4583	1508.0	128.000000	360.0	1.0	Rural	0
2	LP001005	1	Yes	0	Graduate	Yes	3000	0.0	66.000000	360.0	1.0	Urban	1
3	LP001006	1	Yes	0	Not Graduate	No	2583	2358.0	120.000000	360.0	1.0	Urban	1
4	LP001008	1	No	0	Graduate	No	6000	0.0	141.000000	360.0	1.0	Urban	1
...
609	LP002978	0	No	0	Graduate	No	2900	0.0	71.000000	360.0	1.0	Rural	1
610	LP002979	1	Yes	3+	Graduate	No	4106	0.0	40.000000	180.0	1.0	Rural	1
611	LP002983	1	Yes	1	Graduate	No	8072	240.0	253.000000	360.0	1.0	Urban	1
612	LP002984	1	Yes	2	Graduate	No	7583	0.0	187.000000	360.0	1.0	Urban	1
613	LP002990	0	No	0	Graduate	Yes	4583	0.0	133.000000	360.0	0.0	Semurburb	0

542 rows x 13 columns

```
In [89]: data["Education"] = data["Education"].replace("Not Graduate",0,inplace=True)
data["Education"] = data["Education"].replace("Not Graduate",0,inplace=True)
data["Education"].value_counts()
```

```
Out[89]:
```

```
In [89]: data["Education"] = data["Education"].replace("Not Graduate",0,inplace=True)
data["Education"] = data["Education"].replace("Not Graduate",0,inplace=True)
data["Education"].value_counts()
```

```
Out[89]:
```

```
In [90]: data["Self_Employed"] = data["Self_Employed"].replace("Yes",1,inplace=True)
data["Self_Employed"] = data["Self_Employed"].replace("No",0,inplace=True)
data["Self_Employed"].value_counts()
```

```
Out[90]:
```

```
In [91]: data["Property_Area"] = data["Property_Area"].replace("Urban",0,inplace=True)
data["Property_Area"] = data["Property_Area"].replace("Suburban",2,inplace=True)
data["Property_Area"].value_counts()
```

```
Out[91]:
```

```
In [92]: data["Property_Area"] = data["Property_Area"].replace("Urban",0,inplace=True)
data["Property_Area"] = data["Property_Area"].replace("Suburban",2,inplace=True)
data["Property_Area"].value_counts()
```

```
Out[92]:
```

```
In [93]: data["Property_Area"] = data["Property_Area"].replace("Urban",0,inplace=True)
data["Property_Area"] = data["Property_Area"].replace("Suburban",2,inplace=True)
data["Property_Area"].value_counts()
```

```
Out[93]:
```

```
In [94]: data["Property_Area"] = data["Property_Area"].replace("Urban",0,inplace=True)
data["Property_Area"] = data["Property_Area"].replace("Suburban",2,inplace=True)
data["Property_Area"].value_counts()
```

```
Out[94]:
```

```
In [95]: data.head()
```

```
Out[95]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	1	0	0	1	0	5849	0.0	146.412162	360.0	1.0	0	1
1	LP001003	1	1	1	1	0	4583	1508.0	128.000000	360.0	1.0	1	0
2	LP001005	1	1	0	1	1	3000	0.0	66.000000	360.0	1.0	0	1
3	LP001006	1	1	0	0	0	2583	23					