In [9]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [10]:
```python
loan_data=pd.read_csv(r"F:\FSDS\Data Files\train_ctrUa4k.csv")
```

In [11]:
```python
loan_data
```

Out[11]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplicai
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]:
```python
loan_data['Dependents'].unique
```

Out[5]:
```
<bound method Series.unique of 0       0
1       1
2       0
3       0
4       0
       ..
609     0
610    3+
611     1
612     2
613     0
Name: Dependents, Length: 614, dtype: object>
```

separate the data

In [5]:
```python
loan_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            614 non-null    object 
 1   Gender             601 non-null    object 
 2   Married            611 non-null    object 
 3   Dependents         599 non-null    object 
 4   Education          614 non-null    object 
 5   Self_Employed      582 non-null    object 
 6   ApplicantIncome    614 non-null    int64  
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object 
 12  Loan_Status        614 non-null    object 
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

```
In [6]: cat=loan_data.select_dtypes(include='object').columns
        num=loan_data.select_dtypes(exclude='object').columns
```

```
In [7]: cat
```

```
Out[7]: Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
               'Self_Employed', 'Property_Area', 'Loan_Status'],
              dtype='object')
```

```
In [8]: num
```

```
Out[8]: Index(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
               'Loan_Amount_Term', 'Credit_History'],
              dtype='object')
```

Data quick checks

```
In [9]: loan_data.head()
```

Out[9]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantI
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	

```
In [10]: loan_data.tail()
```

Out[10]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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	

In [11]:

```
loan_data.shape
```

Out[11]:

```
(614, 13)
```

In [12]:

```
loan_data.describe()
```

Out[12]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

In [13]:

```
loan_data.isnull().sum()
```

Out[13]:

```
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
```

Fill missing values

- To fill the numerical columns we use median or KNN-Imputer
- To fill the categorical columns we use mode

In [14]:

```
num
```

Out[14]:

```
Index(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History'],
      dtype='object')
```

In [15]:

```
loan_data['LoanAmount']
```

Out[15]:

```
0        NaN
1      128.0
2       66.0
3      120.0
4      141.0
       ...  
609     71.0
610     40.0
611    253.0
612    187.0
613    133.0
Name: LoanAmount, Length: 614, dtype: float64
```

In [16]:

```
loan_data['LoanAmount'].isnull().sum()
```

Out[16]:

```
22
```

In [17]:

```
median_amt=loan_data['LoanAmount'].median()
loan_data['LoanAmount'].fillna(median_amt, inplace=True)
```

In [18]:

```
loan_data['LoanAmount'].isnull().sum()
```

Out[18]:

```
0
```

In [19]:

```
median_amt=loan_data['Loan_Amount_Term'].median()
loan_data['Loan_Amount_Term'].fillna(median_amt, inplace=True)
```

In [20]:

```
loan_data['Loan_Amount_Term'].isnull().sum()
```

Out[20]:

```
0
```

In [21]:

```
median_amt=loan_data['Credit_History'].median()
loan_data['Credit_History'].fillna(median_amt, inplace=True)
```

In [22]:

```
loan_data['Credit_History'].isnull().sum()
```

Out[22]:

```
0
```

Fill the categorical columns

In [23]:

```
cat
```

Out[23]:

```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [24]:

```
mode=loan_data['Gender'].mode()[0]
loan_data['Gender'].fillna(mode,inplace=True)
```

In [25]:

```
loan_data['Gender'].isnull().sum()
```

Out[25]:

```
0
```

In [26]:

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

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

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

In [27]:

```
loan_data.isnull().sum()
```

Out[27]:

```
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
```

In [28]:

```
#step-6 now drop the columns
loan_data.head()
```

Out[28]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantI
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	

In [29]:

```
loan_data.drop('Loan_ID', axis=1, inplace=True)
```

In [30]:

```
loan_data.head()
```

Out[30]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	Lo
0	Male	No	0	Graduate	No	5849	0.0	
1	Male	Yes	1	Graduate	No	4583	1508.0	
2	Male	Yes	0	Graduate	Yes	3000	0.0	
3	Male	Yes	0	Not Graduate	No	2583	2358.0	
4	Male	No	0	Graduate	No	6000	0.0	

Categorical columns Analysis

In [31]:
```
cat=loan_data.select_dtypes(include='object').columns
```

In [32]:
```
cat
```

Out[32]:
```
Index(['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed',
       'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [33]:
```
unique=loan_data['Gender'].unique()
unique
```

Out[33]:
```
array(['Male', 'Female'], dtype=object)
```

In [34]:
```
loan_data[['Gender']].value_counts()
```

Out[34]:
```
Gender
Male      502
Female    112
dtype: int64
```

In [35]:
```
count=[]
for i in unique:
    con=loan_data['Gender']==i
    count.append(len(loan_data[con]))

count
```

Out[35]:
```
[502, 112]
```

In [36]:
```
df=pd.DataFrame(zip(unique,count), columns=['labels', 'count'], index=[1,2])
df
```

Out[36]:

	labels	count
1	Male	502
2	Female	112

In [37]:
```
loan_data
```

Out[37]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
0	Male	No	0	Graduate	No	5849	0.0
1	Male	Yes	1	Graduate	No	4583	1508.0
2	Male	Yes	0	Graduate	Yes	3000	0.0
3	Male	Yes	0	Not Graduate	No	2583	2358.0
4	Male	No	0	Graduate	No	6000	0.0
...	...	...	...	...	...	...	...
609	Female	No	0	Graduate	No	2900	0.0
610	Male	Yes	3+	Graduate	No	4106	0.0
611	Male	Yes	1	Graduate	No	8072	240.0
612	Male	Yes	2	Graduate	No	7583	0.0
613	Female	No	0	Graduate	Yes	4583	0.0

614 rows × 12 columns

In [38]:

```
cat
```

Out[38]:

```
Index(['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed',
       'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [39]:

```
import os
new_folder='project2'
cwd=os.getcwd()
new_dir=os.path.join(cwd, new_folder)
try:
    os.makedirs(new_dir)
except Exception as e:
    print(e)
```

```
[WinError 183] Cannot create a file when that file already exists: 'C:\\Users\\NEHA
\\project2'
```

In [40]:

```
for i in cat:
    keys=loan_data[i].value_counts().keys()
    values=loan_data[i].value_counts().values
    cols=['labels', 'count']
    df1=pd.DataFrame(zip(keys, values),columns=cols)
#     df1

    file_name=f'{i}_table.csv'
    new_path1=os.path.join(new_dir, file_name)
    df1.to_csv(new_path1)
```

In [41]:

```
for col in cat:
    keys=loan_data[col].value_counts().keys()
    values=loan_data[col].value_counts().values
    print(keys)
    print(values)
```

```
Index(['Male', 'Female'], dtype='object')
[502 112]
Index(['Yes', 'No'], dtype='object')
[401 213]
Index(['0', '1', '2', '3+'], dtype='object')
[360 102 101  51]
Index(['Graduate', 'Not Graduate'], dtype='object')
[480 134]
Index(['No', 'Yes'], dtype='object')
[532  82]
Index(['Semiurban', 'Urban', 'Rural'], dtype='object')
[233 202 179]
Index(['Y', 'N'], dtype='object')
[422 192]
```

Barchart

In [42]:

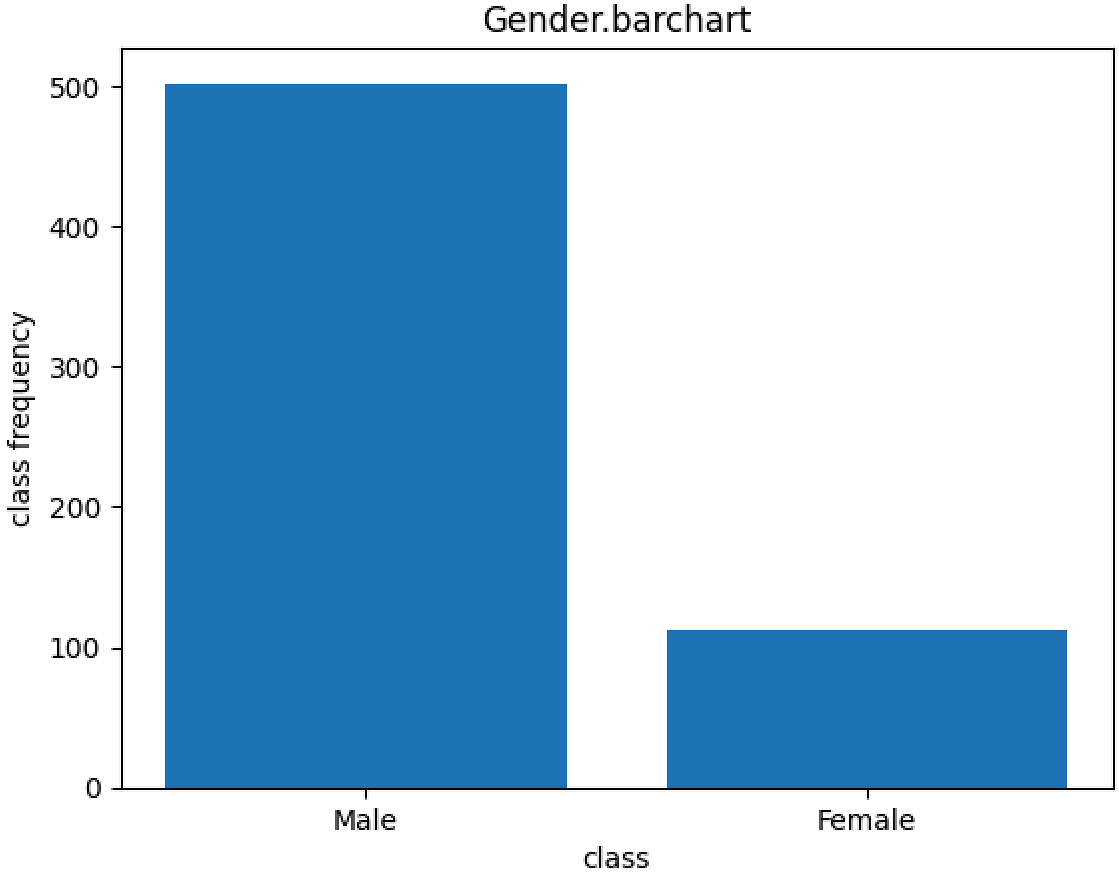
```
cat
```

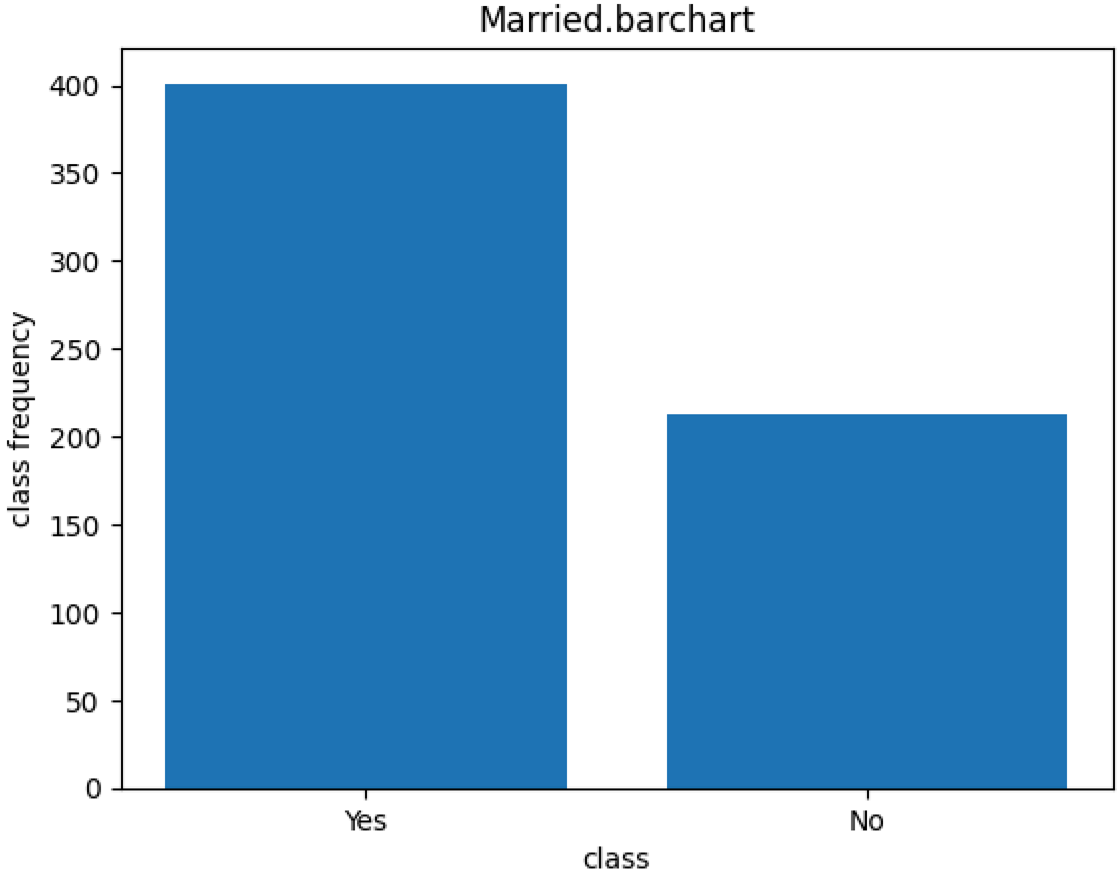
Out[42]:

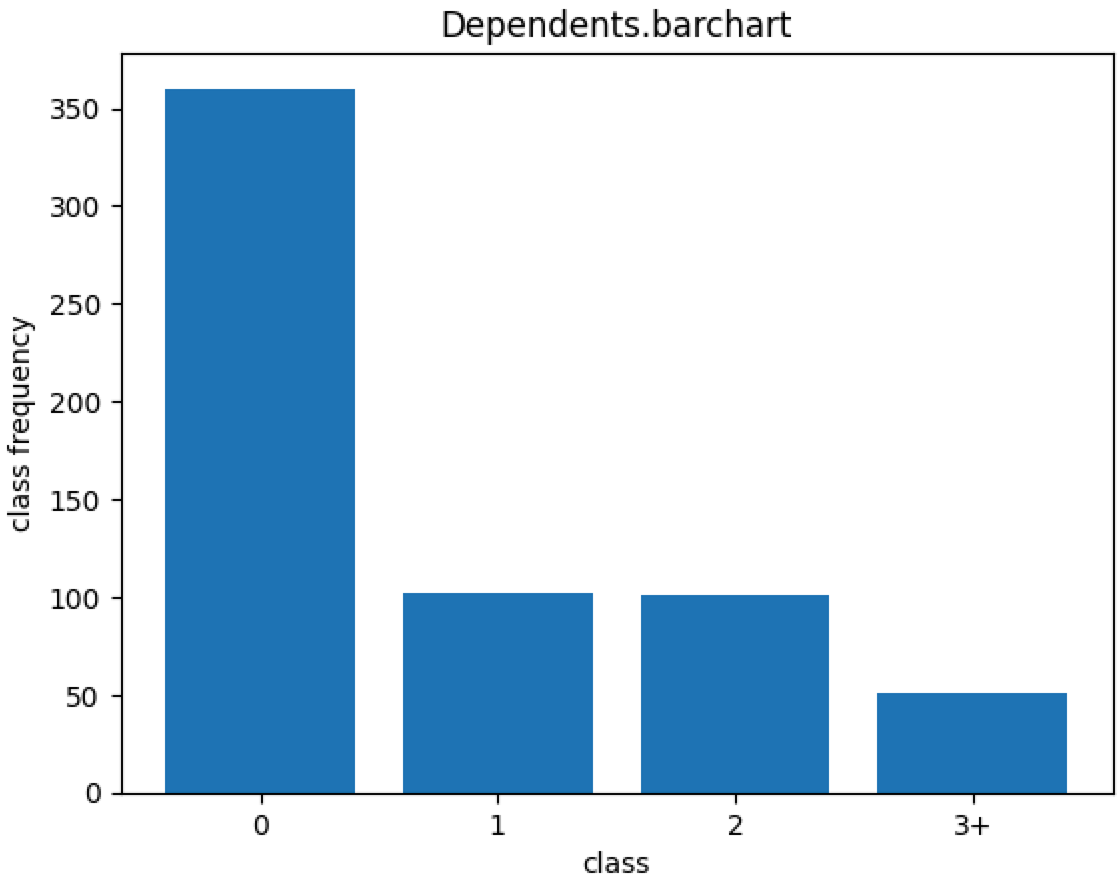
```
Index(['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed',
       'Property_Area', 'Loan_Status'],
      dtype='object')
```

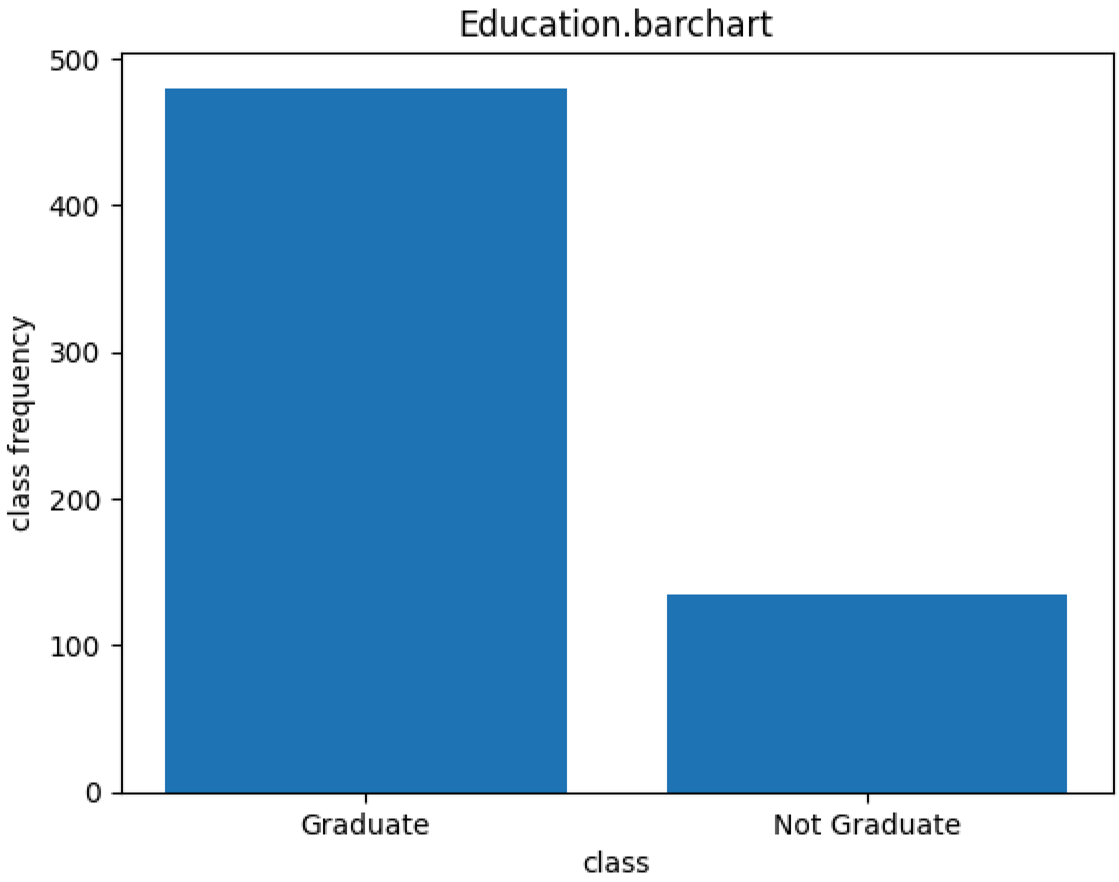
In [43]:

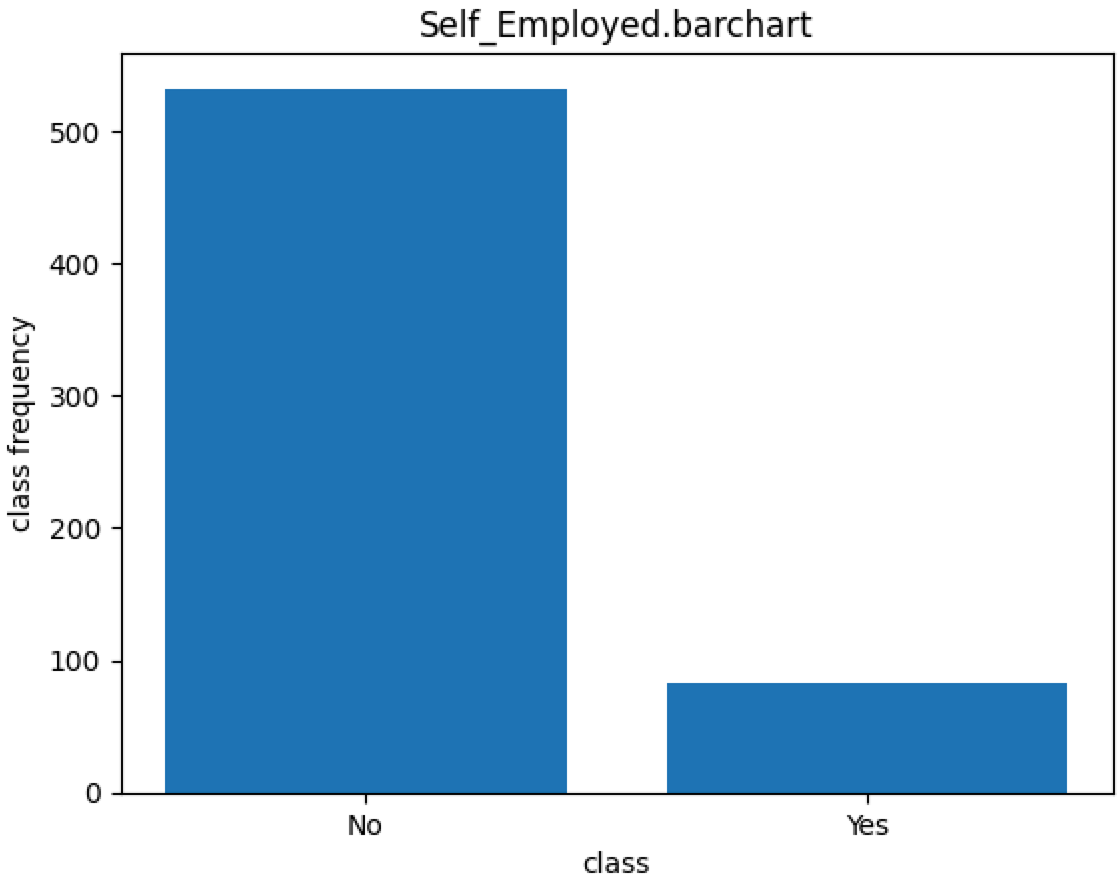
```
for col in cat:
    keys=loan_data[col].value_counts().keys()
    values=loan_data[col].value_counts().values
    plt.bar(keys, values)
    plt.title(f'{col}.barchart')
    plt.xlabel('class')
    plt.ylabel('class frequency')
    plt.savefig('col.png')
    plt.show()
```

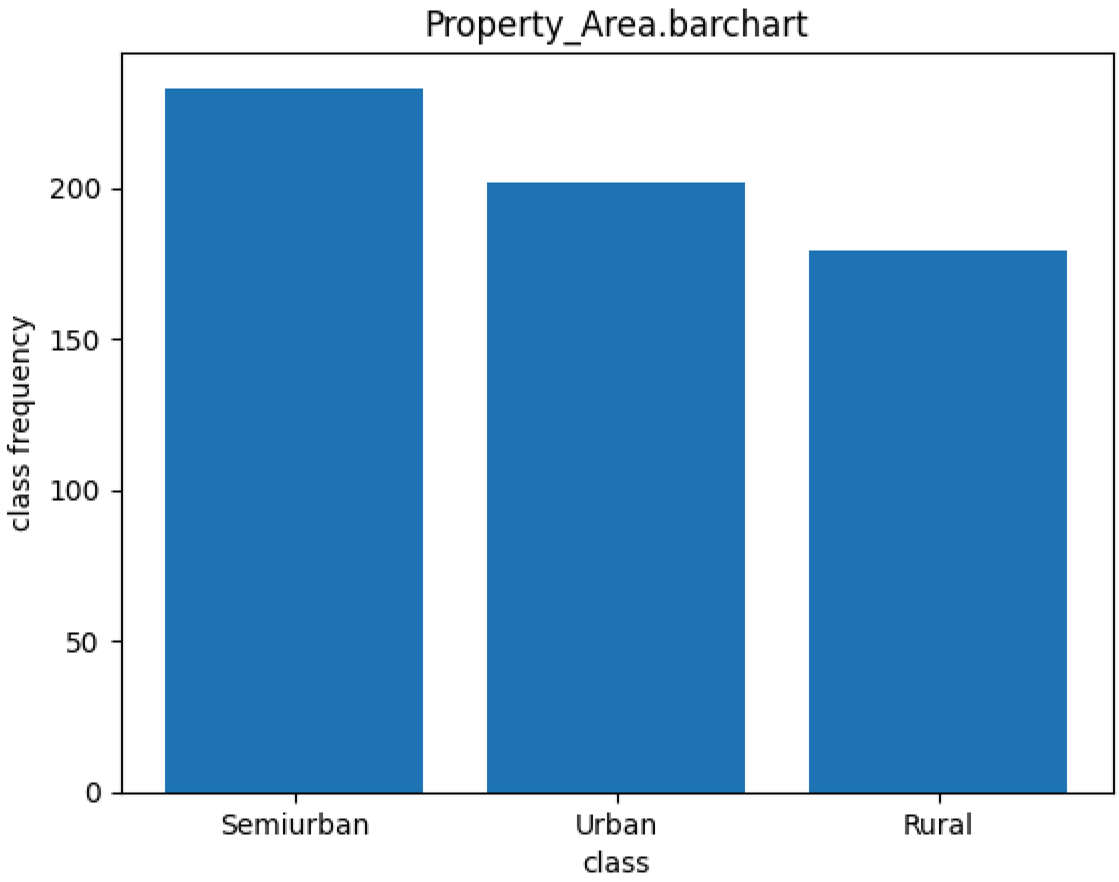

Gender.barchart
class frequency
500
400
300
200
100
0
Male
Female
class

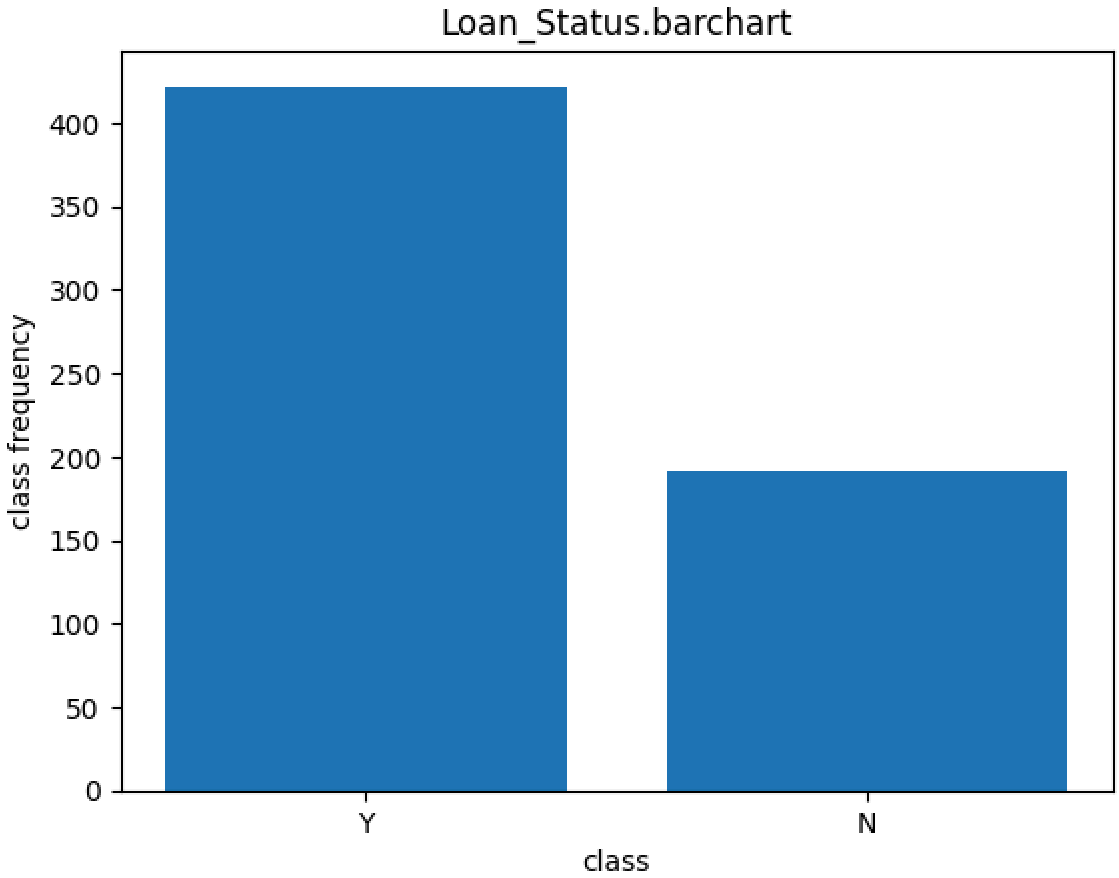

Married.barchart
class frequency
400
350
300
250
200
150
100
50
0
Yes
No
class


Dependents.barchart
class frequency
0
50
100
150
200
250
300
350
0
1
2
3+
class


Education.barchart
class frequency
0
100
200
300
400
500
Graduate
Not Graduate
class

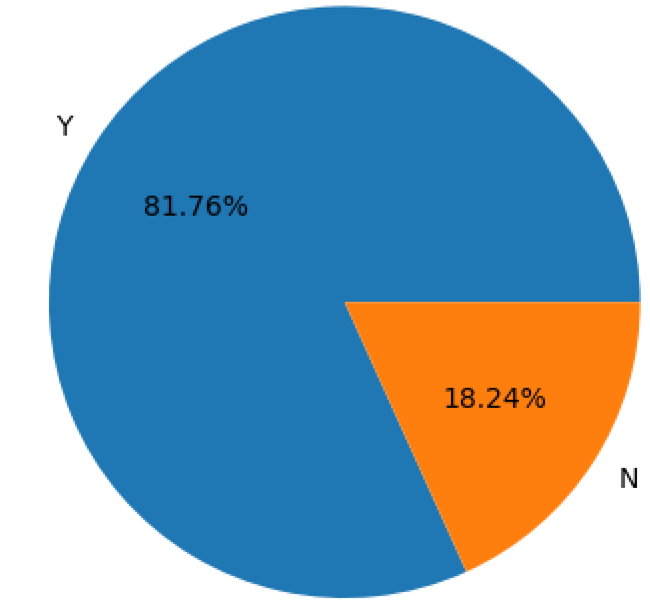

Self_Employed.barchart


Property_Area.barchart



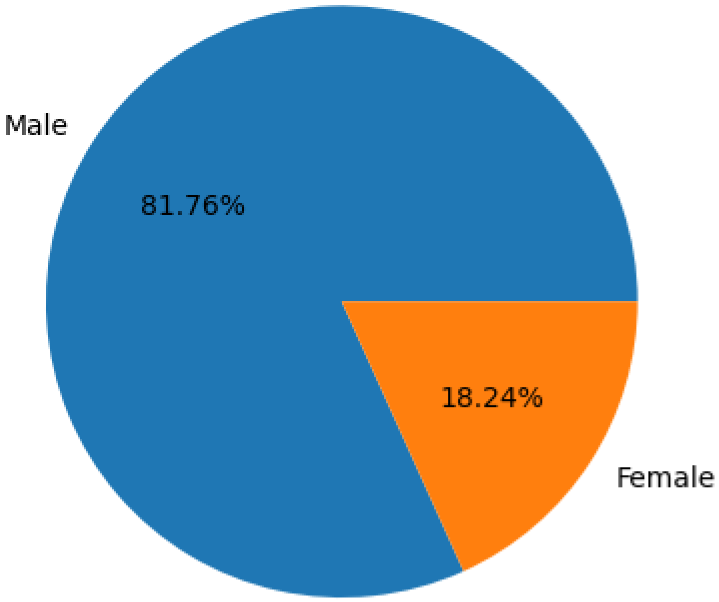
piechart

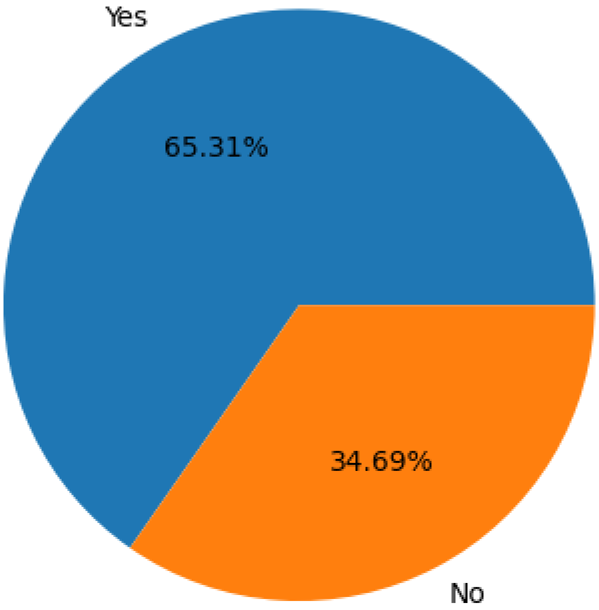
```
In [44]: keys==loan_data['Gender'].value_counts().keys()
         values=loan_data['Gender'].value_counts().values
         plt.pie(x=values, labels=keys, autopct='%0.2f%%')
         plt.show()
```

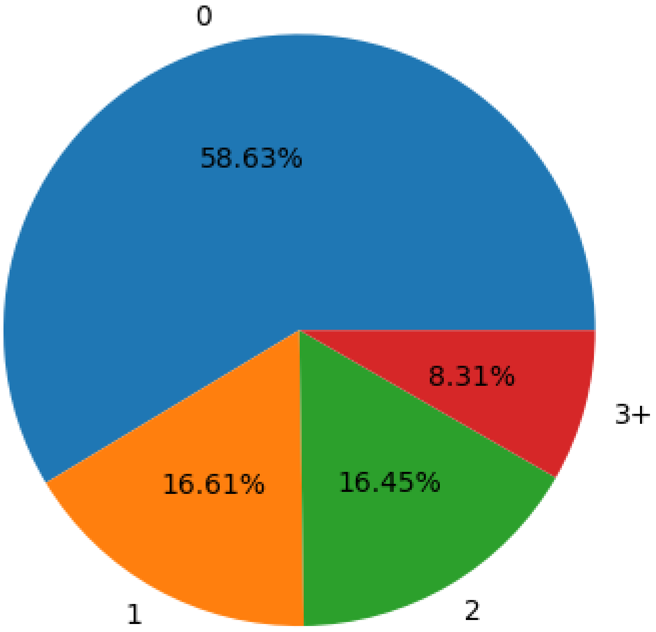


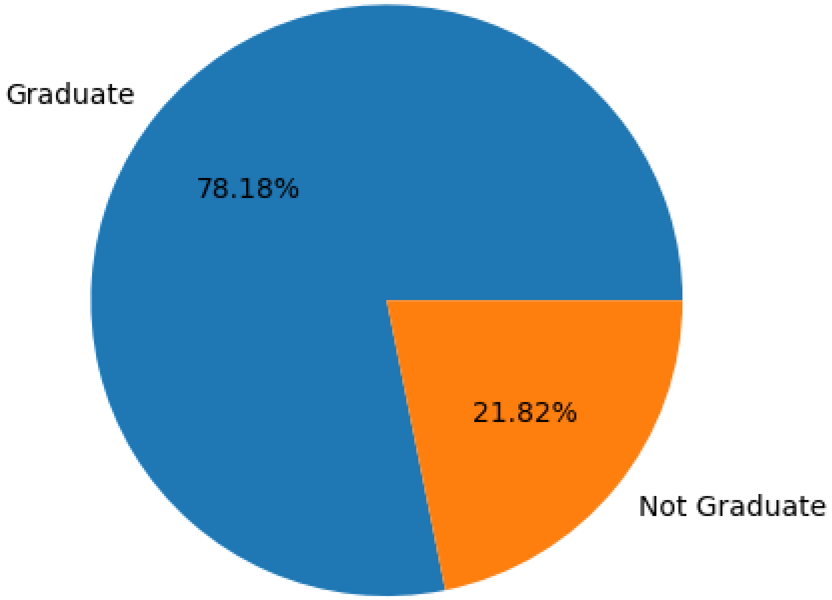
In [53]:

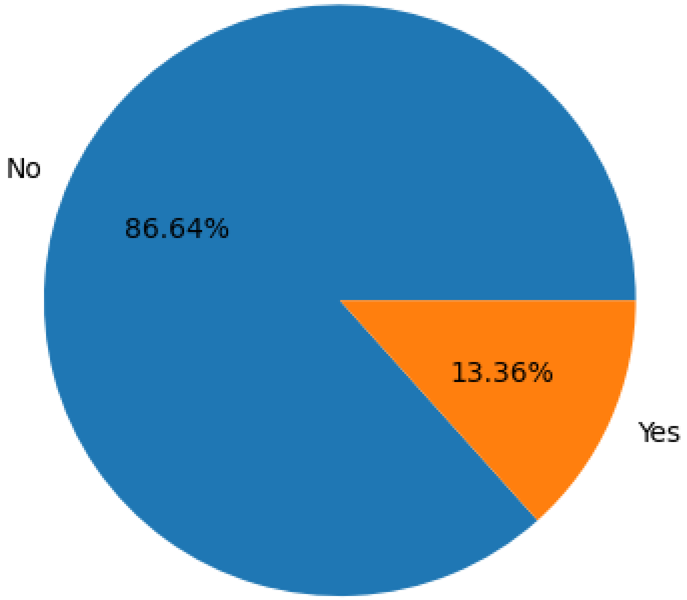
```
for col in cat:
    keys=loan_data[col].value_counts().keys()
    values=loan_data[col].value_counts().values
    plt.pie(values, labels=keys, autopct='%0.2f%%')
    plt.savefig('col.png')
    plt.show()
```

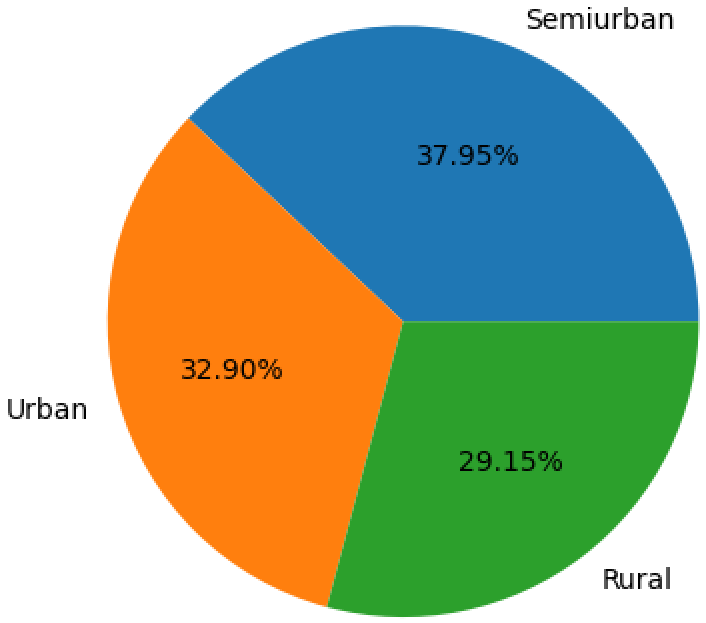


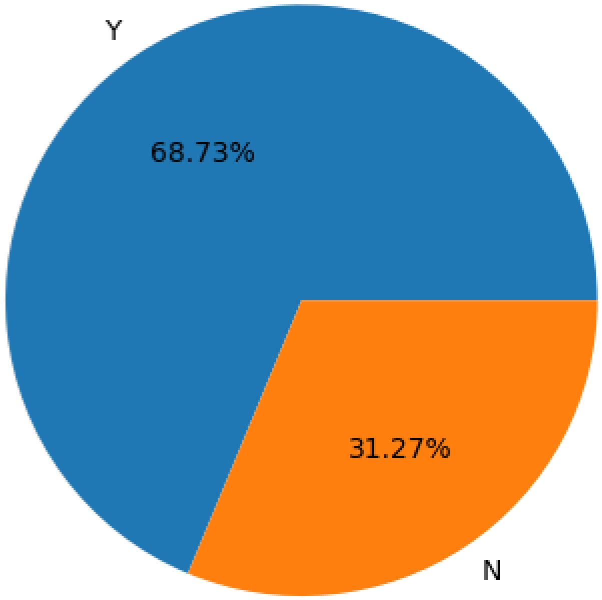

Yes
65.31%
34.69%
No


0
58.63%
8.31%
3+
16.61%
16.45%
1
2


Graduate
78.18%
21.82%
Not Graduate


No
86.64%
13.36%
Yes





Numerical column Analysis

- Histogram

```
In [45]: num
```

```
Out[45]: Index(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
                'Loan_Amount_Term', 'Credit_History'],
               dtype='object')
```

In [46]:

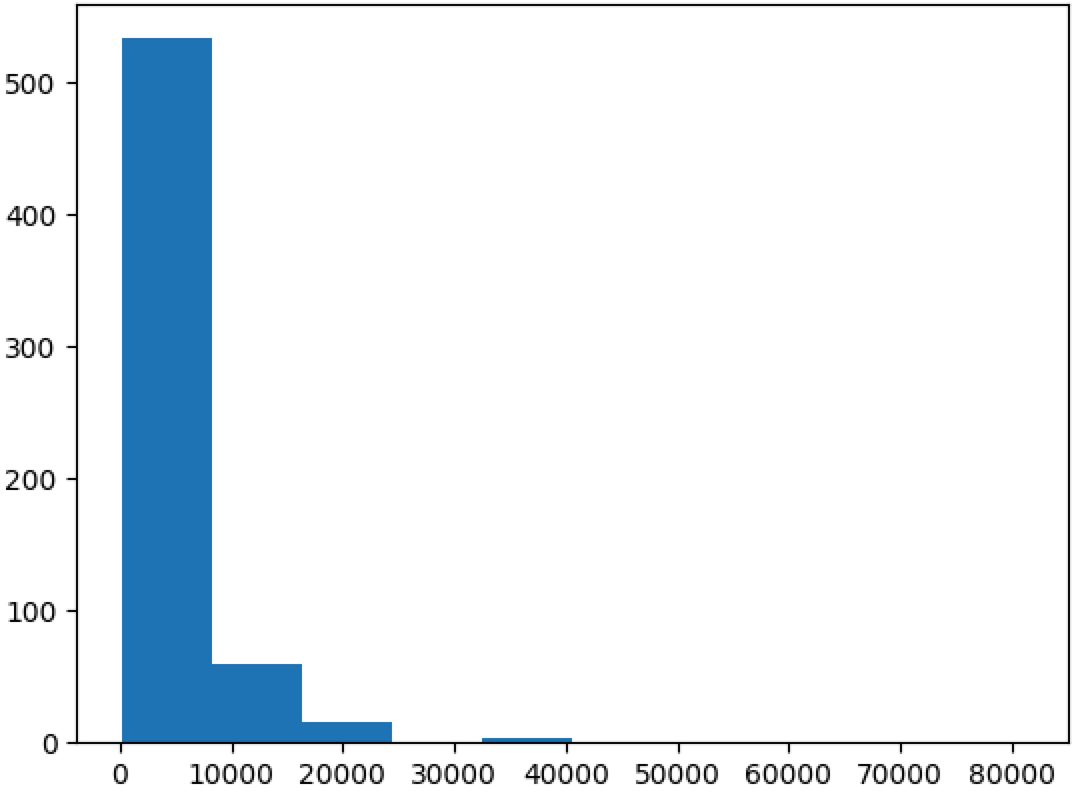
```
loan_data.describe()
```

Out[46]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	614.000000	614.000000	614.000000
mean	5403.459283	1621.245798	145.752443	342.410423	0.855049
std	6109.041673	2926.248369	84.107233	64.428629	0.352339
min	150.000000	0.000000	9.000000	12.000000	0.000000
25%	2877.500000	0.000000	100.250000	360.000000	1.000000
50%	3812.500000	1188.500000	128.000000	360.000000	1.000000
75%	5795.000000	2297.250000	164.750000	360.000000	1.000000
max	81000.000000	41667.000000	700.000000	480.000000	1.000000

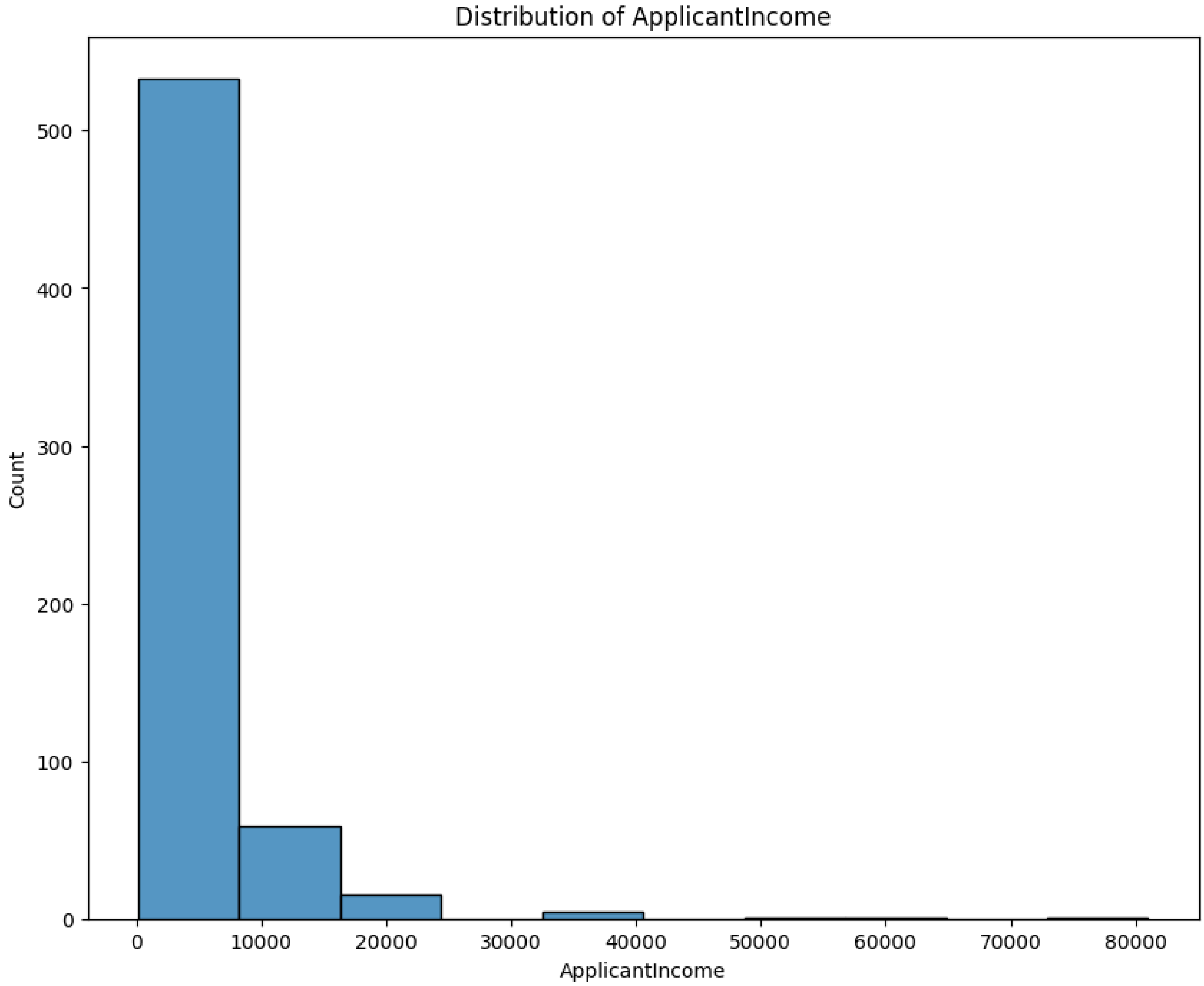
In [47]:

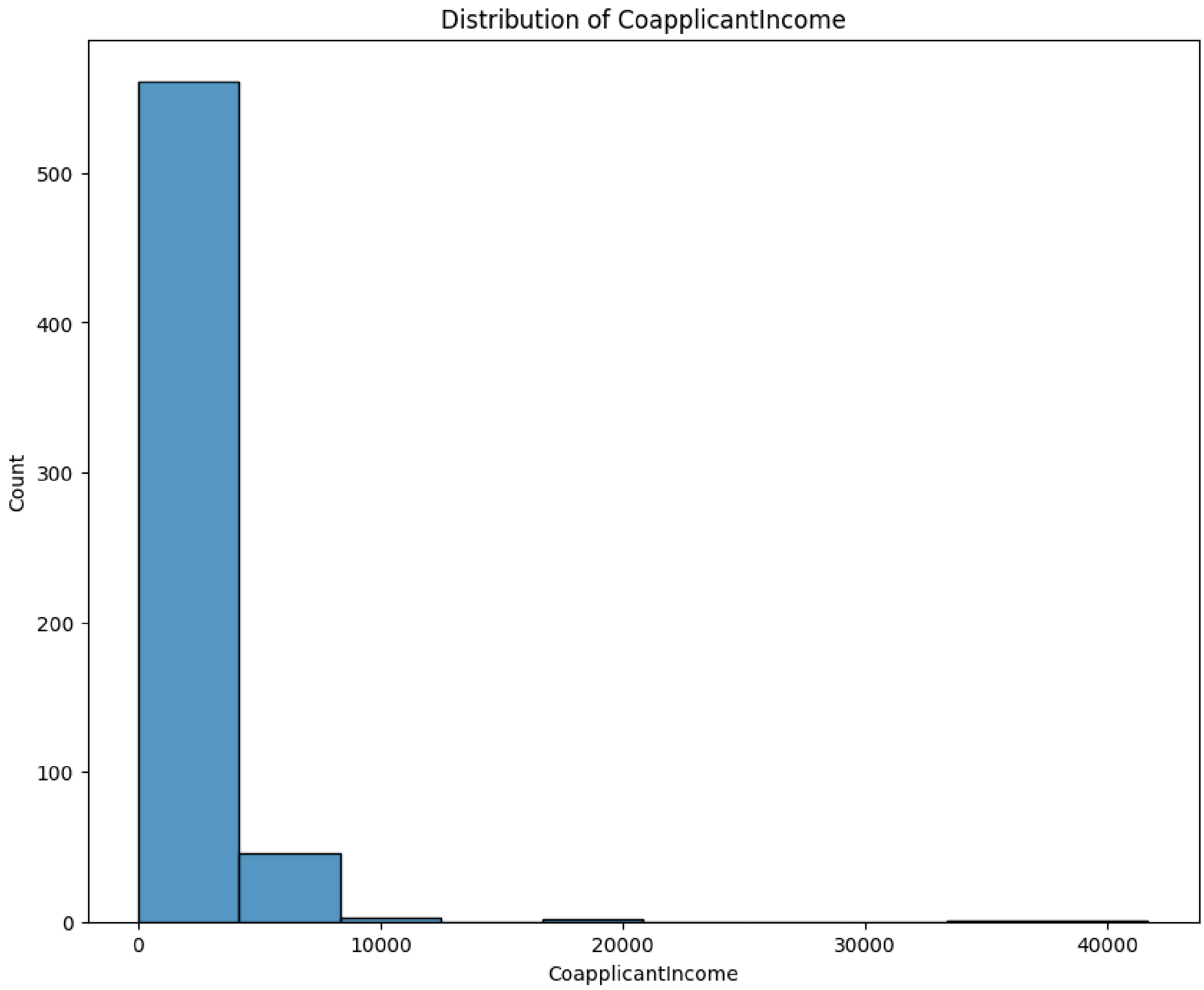
```
# Histplot
plt.hist(loan_data['ApplicantIncome'],bins=10)
plt.show()
```

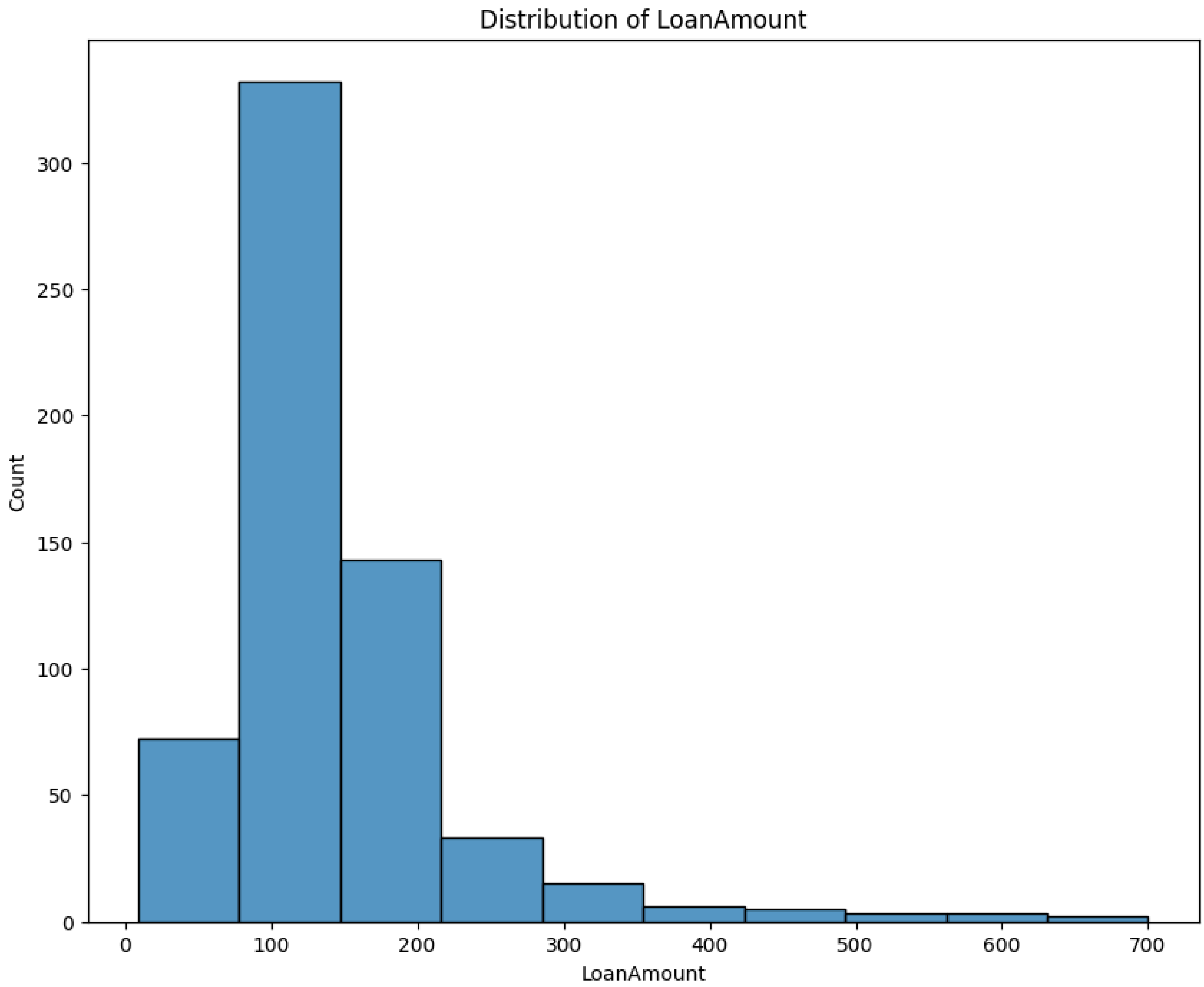


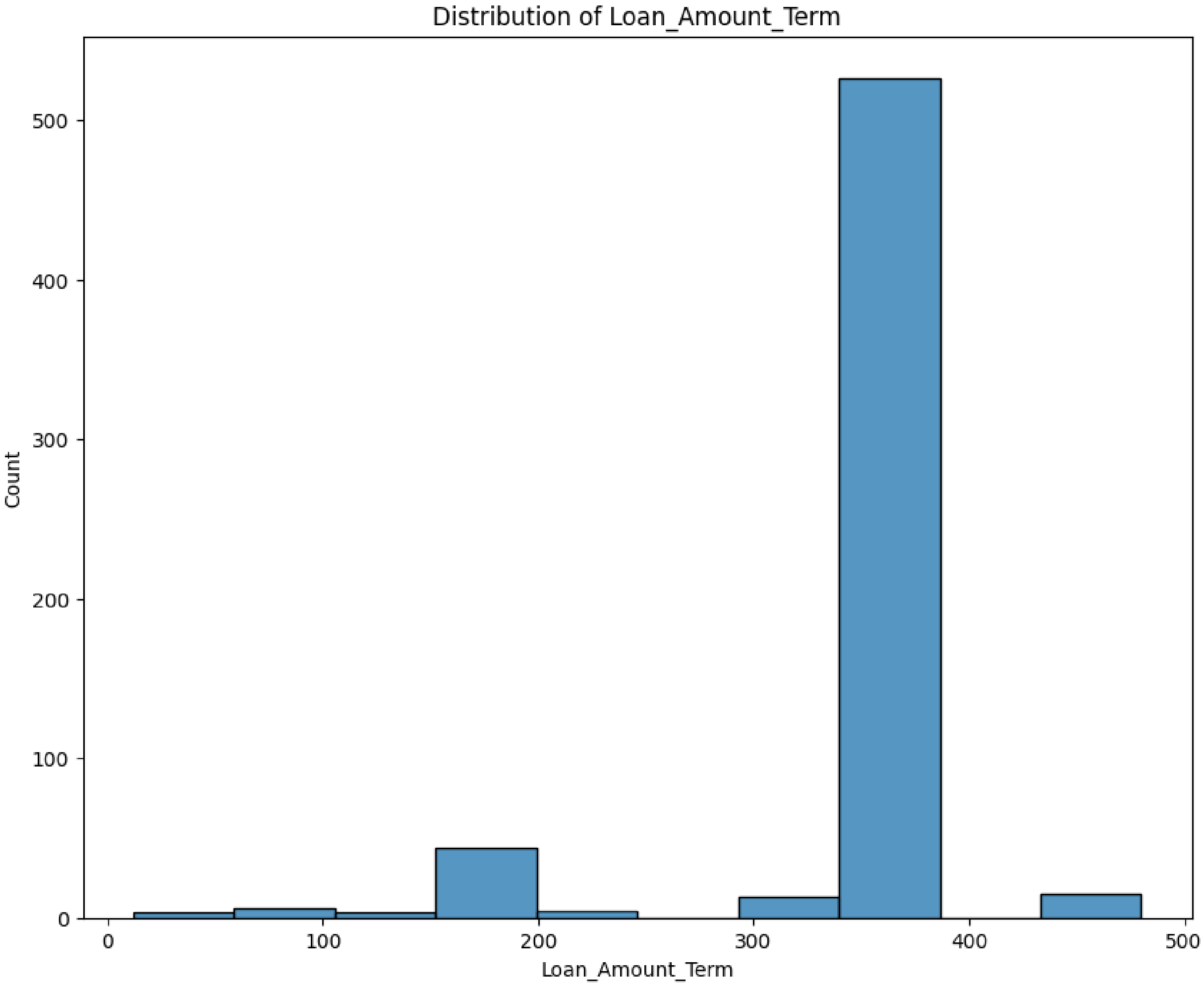
In [48]:

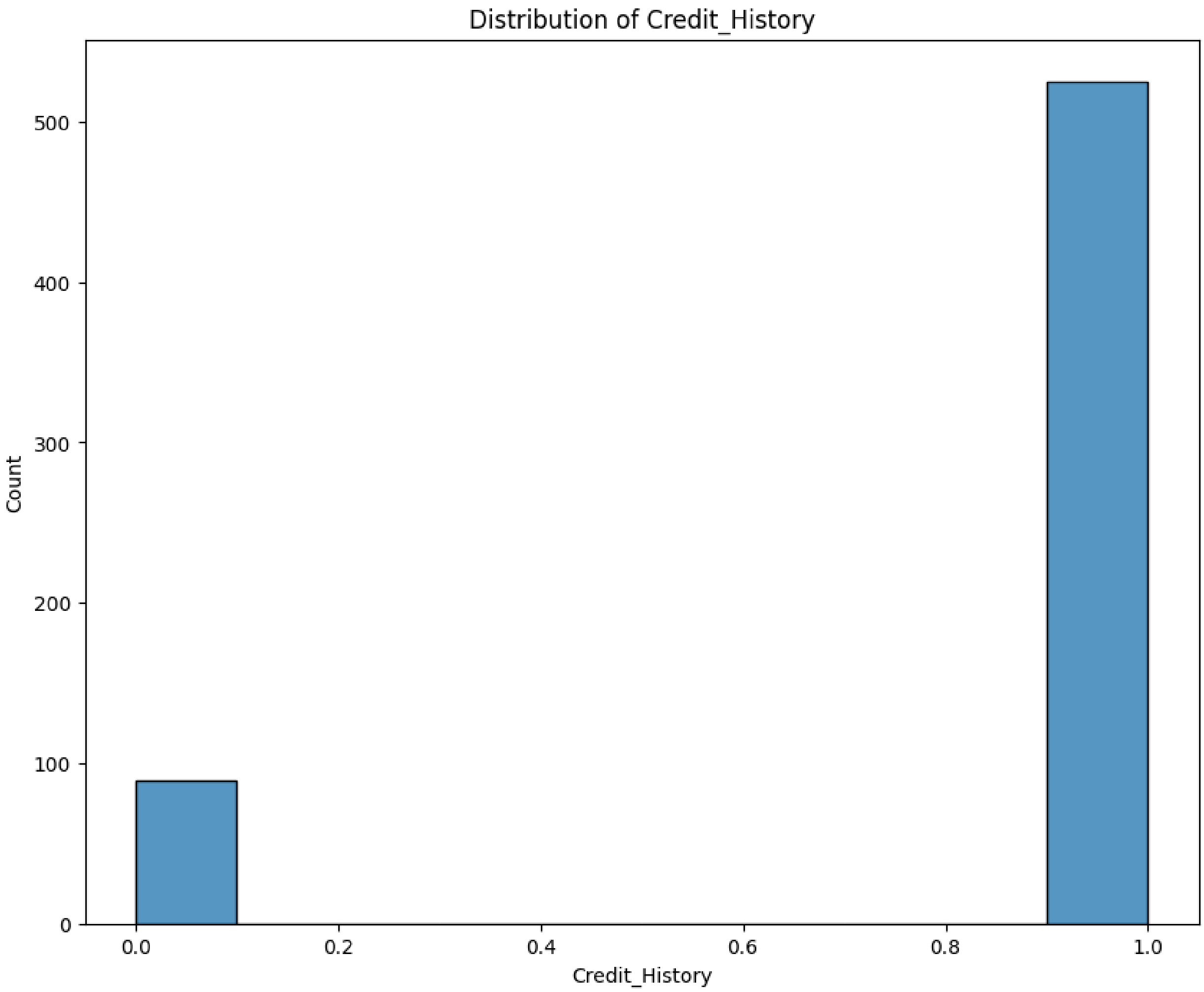
```
for col in num:
    plt.figure(figsize=(10,8))
    sns.histplot(loan_data[col], bins=10, kde=False)
    plt.title(f'Distribution of {col}')
    plt.savefig(f'{col}.png')
    plt.show()
```


Distribution of ApplicantIncome
Count
0
100
200
300
400
500
0
10000
20000
30000
40000
50000
60000
70000
80000
ApplicantIncome


Distribution of CoapplicantIncome
Count
0
100
200
300
400
500
0
10000
20000
30000
40000
CoapplicantIncome


Distribution of LoanAmount
Count
0
50
100
150
200
250
300
0
100
200
300
400
500
600
700
LoanAmount

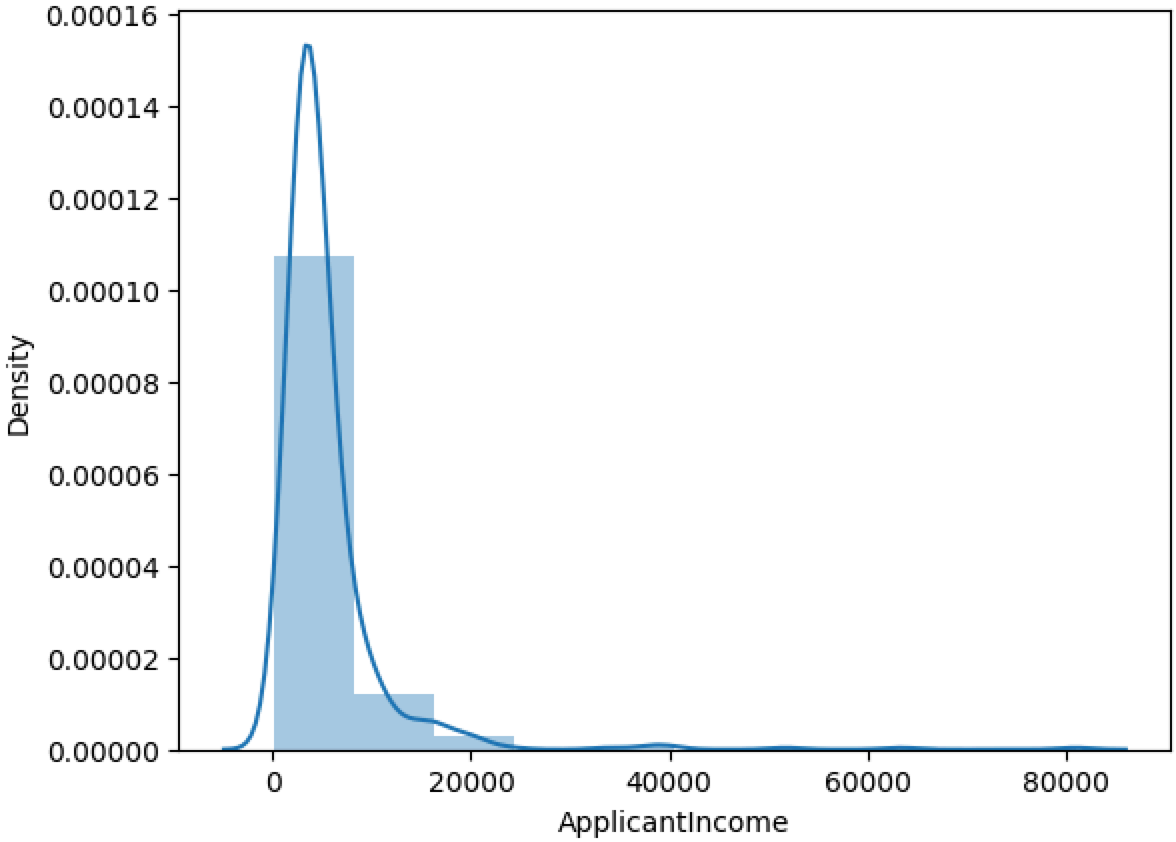

Distribution of Loan_Amount_Term
Count
0
100
200
300
400
500
0
100
200
300
400
500
Loan_Amount_Term



In [6]:

```
# Distplot
import warnings
warnings.filterwarnings('ignore')

sns.distplot(loan_data['ApplicantIncome'], bins=10)
plt.show()
```



```
In [ ]: #Barplot
```

Outlier Analysis

```
In [50]: cat
```

```
Out[50]: Index(['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed',
                'Property_Area', 'Loan_Status'],
               dtype='object')
```

```
In [51]: num
```

```
Out[51]: Index(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
                'Loan_Amount_Term', 'Credit_History'],
               dtype='object')
```

```
In [52]: loan_data.head()
```

Out[52]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	Lo
0	Male	No	0	Graduate	No	5849	0.0	
1	Male	Yes	1	Graduate	No	4583	1508.0	
2	Male	Yes	0	Graduate	Yes	3000	0.0	
3	Male	Yes	0	Not Graduate	No	2583	2358.0	
4	Male	No	0	Graduate	No	6000	0.0	

In [54]:

```
income_df=loan_data['ApplicantIncome']
q1=round(np.quantile(income_df, 0.25), 2)
q3=round(np.quantile(income_df, 0.75), 2)
IQR= q3-q1
lb=q1-1.5*IQR
ub=q3+1.5*IQR
con1=loan_data['ApplicantIncome']>lb
con2=loan_data['ApplicantIncome']<ub
con3=con1&con2
count=len(loan_data[con3])
non_outliers_data=loan_data[con3]
non_outliers_data
```

Out[54]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
0	Male	No	0	Graduate	No	5849	0.0
1	Male	Yes	1	Graduate	No	4583	1508.0
2	Male	Yes	0	Graduate	Yes	3000	0.0
3	Male	Yes	0	Not Graduate	No	2583	2358.0
4	Male	No	0	Graduate	No	6000	0.0
...	...	...	...	...	...	...	...
609	Female	No	0	Graduate	No	2900	0.0
610	Male	Yes	3+	Graduate	No	4106	0.0
611	Male	Yes	1	Graduate	No	8072	240.0
612	Male	Yes	2	Graduate	No	7583	0.0
613	Female	No	0	Graduate	Yes	4583	0.0

564 rows × 12 columns

In [59]:

```
income_df=loan_data['ApplicantIncome']
q1=round(np.quantile(income_df,0.25),2)
q3=round(np.quantile(income_df, 0.75),2)
IQR=q3-q1
lb=q1-1.5*IQR
ub=q3+1.5*IQR
```

```
median=income_df.median()
new_df=[]
```

In [60]:

```
for i in income_df:
    if i<lb or i>ub:
        new_df.append(median)
    else:
        new_df.append(i)
```

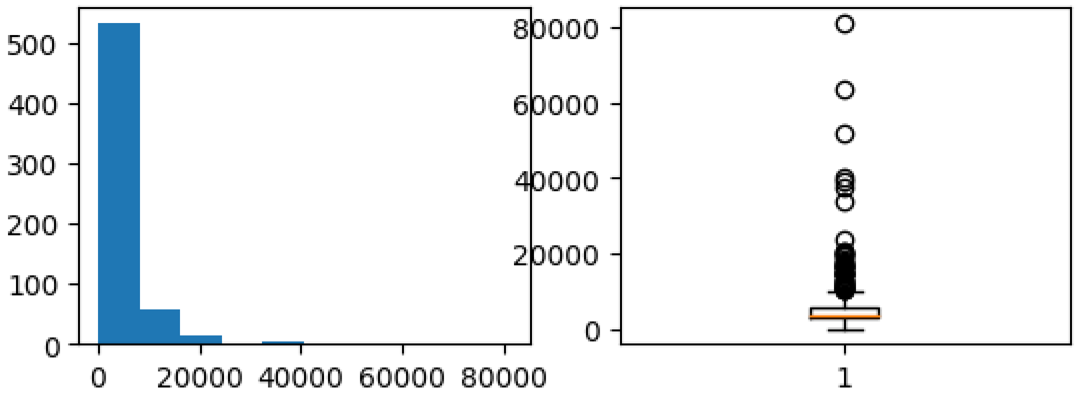
In [61]:

```
new_df
```

```
Out[61]: [5849,
           4583,
           3000,
           2583,
           6000,
           5417,
           2333,
           3036,
           4006,
           3812.5,
           3200,
           2500,
           3073,
           1853,
           1299,
           4950,
           3596,
           3510,
           4887,
           2600,
           7660,
           5955,
           2600,
           3365,
           3717,
           9560,
           2799,
           4226,
           1442,
           3750,
           4166,
           3167,
           4692,
           3500,
           3812.5,
           2275,
           1828,
           3667,
           4166,
           3748,
           3600,
           1800,
           2400,
           3941,
           4695,
           3410,
           5649,
           5821,
           2645,
           4000,
           1928,
           3086,
           4230,
           4616,
           3812.5,
           2708,
           2132,
3366,
8080,
3357,
2500,
3029,
2609,
4945,
4166,
5726,
3200,
3812.5,
7100,
4300,
3208,
1875,
3500,
4755,
5266,
3750,
3750,
1000,
3167,
3333,
3846,
2395,
1378,
6000,
3988,
2366,
3333,
2500,
8566,
5695,
2958,
6250,
3273,
4133,
3620,
6782,
2484,
1977,
4188,
1759,
4288,
4843,
3812.5,
4652,
3816,
3052,
3812.5,
7333,
3800,
2071,
5316,
2929,
3572,
7451,
5050,
3812.5,
3167,
2214,
5568,
3812.5,
5667,
4166,
2137,
2957,
4300,
3692,
3812.5,
3865,
3812.5,
6080,
3812.5,
2014,
2718,
3459,
4895,
4000,
4583,
3316,
3812.5,
4200,
5042,
5417,
6950,
2698,
3812.5,
2330,
3812.5,
1538,
10000,
4860,
6277,
2577,
9166,
2281,
3254,
3812.5,
6000,
9538,
2980,
4583,
1863,
7933,
3089,
4167,
9323,
3707,
4583,
2439,
2237,
8000,
1820,
3812.5,
3522,
5708,
4344,
3497,
2045,
5516,
3750,
2333,
6400,
1916,
4600,
3812.5,
3625,
3812.5,
2178,
2383,
674,
9328,
4885,
3812.5,
6033,
3858,
4191,
3125,
8333,
1907,
3416,
3812.5,
2600,
4923,
3992,
3500,
3917,
4408,
3244,
3975,
2479,
3418,
10000,
3430,
7787,
5703,
3173,
3850,
150,
3727,
5000,
4283,
2221,
4009,
2971,
7578,
6250,
3250,
4735,
6250,
4758,
6400,
2491,
3716,
3189,
8333,
3155,
5500,
5746,
3463,
3812,
3315,
5819,
2510,
2965,
6250,
3406,
6050,
9703,
6608,
2882,
1809,
1668,
3427,
2583,
2661,
3812.5,
3083,
6045,
5250,
3812.5,
4931,
6083,
2060,
3481,
7200,
5166,
4095,
4708,
4333,
3418,
2876,
3237,
3812.5,
2833,
2620,
3900,
2750,
3993,
3103,
3812.5,
4100,
4053,
3927,
2301,
1811,
3812.5,
    3158,
    2600,
    3704,
    4124,
    9508,
    3075,
    4400,
    3153,
    5417,
    2383,
    4416,
    6875,
    4666,
    5000,
    2014,
    1800,
    2875,
    5000,
    1625,
    4000,
    2000,
    3762,
    2400,
    3812.5,
    7667,
    2917,
    2927,
    2507,
    5746,
    2473,
    3399,
    3717,
    2058,
    3541,
    10000,
    2400,
    4342,
    3601,
    3166,
    3812.5,
    8666,
    4917,
    5818,
    4333,
    2500,
    4384,
    2935,
    2833,
    3812.5,
    9833,
    5503,
    5250,
    2500,
    1830,
    4160,
    2647,
    2378,
4554,
3173,
2583,
2499,
3523,
3083,
6333,
2625,
9083,
8750,
2666,
5500,
2423,
3813,
8333,
3875,
3000,
5167,
4723,
5000,
4750,
3013,
6822,
6216,
2500,
5124,
6325,
3812.5,
3812.5,
5185,
9323,
3062,
2764,
4817,
8750,
4310,
3069,
5391,
3333,
5941,
6000,
7167,
4566,
3667,
2346,
3010,
2333,
5488,
9167,
9504,
2583,
1993,
3100,
3276,
3180,
3033,
3902,
1500,
2889,
2755,
2500,
1963,
7441,
4547,
2167,
2213,
8300,
3812.5,
3867,
6256,
6096,
2253,
2149,
2995,
2600,
1600,
1025,
3246,
5829,
2720,
1820,
7250,
3812.5,
2666,
4606,
5935,
2920,
2717,
8624,
6500,
3812.5,
2425,
3750,
10047,
1926,
2213,
3812.5,
7142,
3660,
7901,
4707,
3812.5,
7333,
3466,
4652,
3539,
3340,
2769,
2309,
1958,
3948,
2483,
7085,
3859,
4301,
3708,
4354,
8334,
2083,
7740,
3015,
5191,
4166,
6000,
2947,
3812.5,
210,
4333,
3450,
2653,
4691,
2500,
5532,
3812.5,
6700,
2873,
3812.5,
2947,
4350,
3095,
2083,
3812.5,
8333,
1958,
3547,
3812.5,
4583,
2435,
2699,
5333,
3691,
3812.5,
3597,
3326,
2600,
4625,
2895,
6283,
645,
3159,
4865,
4050,
3814,
3510,
3812.5,
3583,
2479,
3812.5,
3598,
6065,
3283,
2130,
5815,
3466,
2031,
3074,
4683,
3400,
2192,
2500,
5677,
7948,
4680,
3812.5,
3775,
5285,
2679,
6783,
1025,
4281,
3588,
3812.5,
3812.5,
2550,
6133,
3617,
2917,
6417,
4608,
2138,
3652,
2239,
3017,
2768,
3358,
2526,
5000,
2785,
6633,
2492,
3333,
2454,
3593,
5468,
2667,
10139,
3887,
4180,
3675,
3812.5,
5923,
5800,
8799,
4467,
3333,
3400,
2378,
3166,
 3417,
 5116,
 3812.5,
 6125,
 6406,
 3159,
 3087,
 3229,
 1782,
 3182,
 6540,
 1836,
 3166,
 1880,
 2787,
 4283,
 2297,
 2165,
 4750,
 2726,
 3000,
 6000,
 9357,
 3859,
 3812.5,
 3833,
 6383,
 2987,
 9963,
 5780,
 416,
 2894,
 5703,
 3676,
 3812.5,
 2400,
 3400,
 3987,
 3232,
 2900,
 4106,
 8072,
 7583,
 4583]
```

In [62]:

```
plt.subplot(2,2,1).hist(loan_data['ApplicantIncome'])
plt.subplot(2,2,2).boxplot(loan_data['ApplicantIncome'])
plt.show()
```



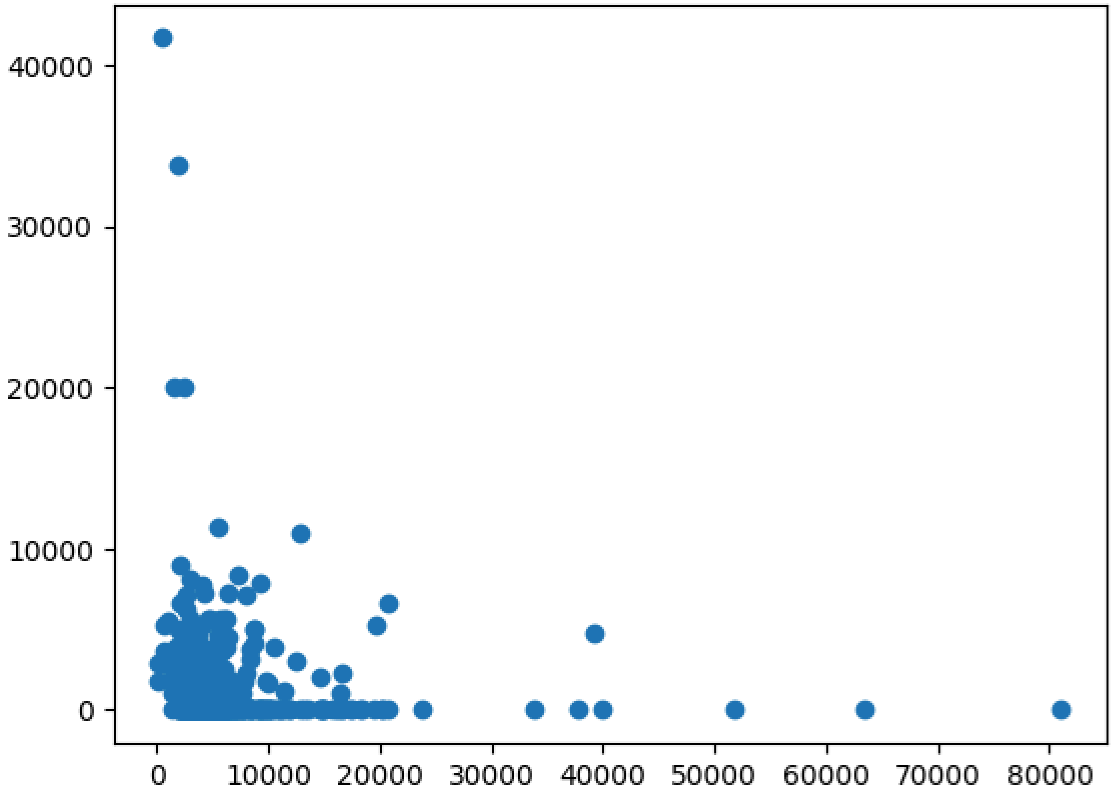
Correlation

```
In [63]: num
         # find the correlation between numerical columns
```

```
Out[63]: Index(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
                'Loan_Amount_Term', 'Credit_History'],
               dtype='object')
```

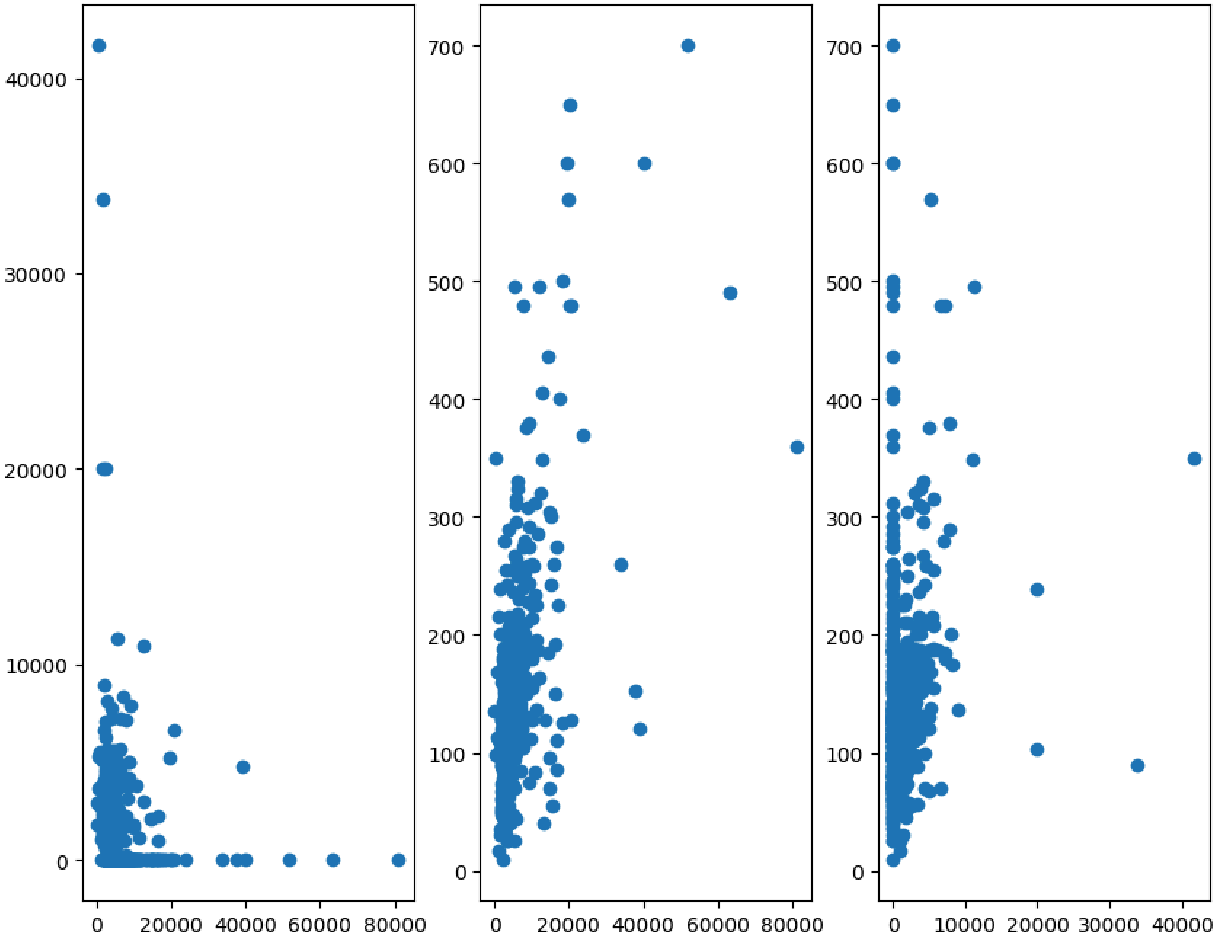
```
In [64]: col1=loan_data['ApplicantIncome']
         col2=loan_data['CoapplicantIncome']
         col3=loan_data['LoanAmount']
         plt.scatter(col1, col2)
```

```
Out[64]: <matplotlib.collections.PathCollection at 0x1f1d8aa05b0>
```



```
In [65]: plt.figure(figsize=(10,8))
         plt.subplot(1,3,1).scatter(col1,col2)
         plt.subplot(1,3,2).scatter(col1,col3)
         plt.subplot(1,3,3).scatter(col2,col3)
```

Out[65]:
```
<matplotlib.collections.PathCollection at 0x1f1d692cd00>
```



In [66]:
```
loan_data.corr(numeric_only=True)
```

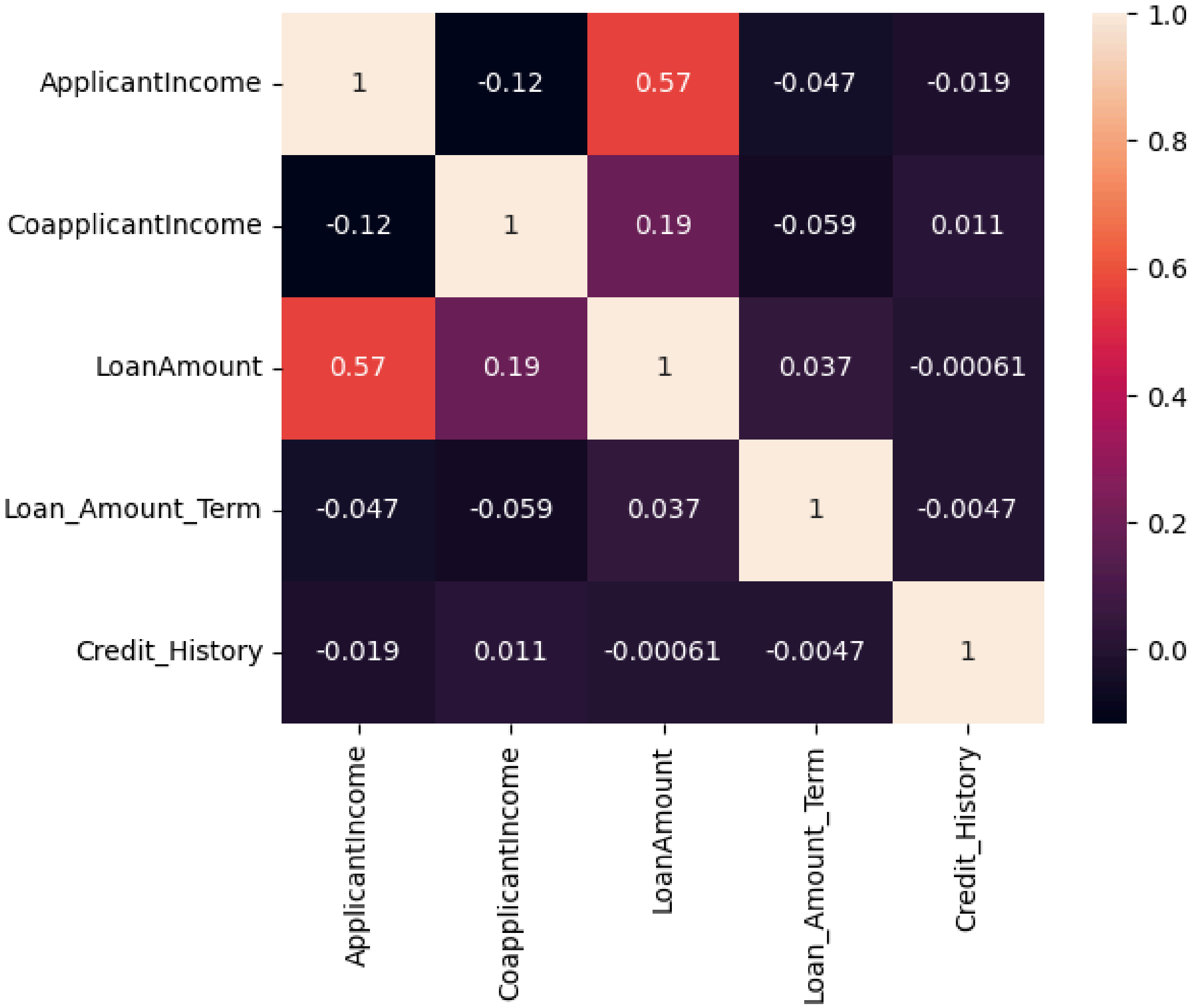
Out[66]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_Hi
ApplicantIncome	1.000000	-0.116605	0.565181	-0.046531	-0.01
CoapplicantIncome	-0.116605	1.000000	0.189218	-0.059383	0.01
LoanAmount	0.565181	0.189218	1.000000	0.036960	-0.00
Loan_Amount_Term	-0.046531	-0.059383	0.036960	1.000000	-0.00
Credit_History	-0.018615	0.011134	-0.000607	-0.004705	1.00

Heatmap

In [67]:
```
corr=loan_data.corr(numeric_only=True)
sns.heatmap(corr, annot=True)
```

Out[67]:
```
<AxesSubplot: >
```



Convert categorical to numerical collumn

- labelEncoder

```
In [68]: cat
```

```
Out[68]: Index(['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed',
                'Property_Area', 'Loan_Status'],
               dtype='object')
```

```
In [69]: from sklearn.preprocessing import LabelEncoder
```

```
In [71]: le=LabelEncoder()
```

```
In [72]: loan_data['Gender']=le.fit_transform(loan_data['Gender'])
```

```
In [73]: loan_data
```

Out[73]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
0	1	No	0	Graduate	No	5849	0.0
1	1	Yes	1	Graduate	No	4583	1508.0
2	1	Yes	0	Graduate	Yes	3000	0.0
3	1	Yes	0	Not Graduate	No	2583	2358.0
4	1	No	0	Graduate	No	6000	0.0
...	...	...	...	...	...	...	...
609	0	No	0	Graduate	No	2900	0.0
610	1	Yes	3+	Graduate	No	4106	0.0
611	1	Yes	1	Graduate	No	8072	240.0
612	1	Yes	2	Graduate	No	7583	0.0
613	0	No	0	Graduate	Yes	4583	0.0

614 rows × 12 columns

In [76]:

```python
for col in cat:
    loan_data['Gender']=le.fit_transform(loan_data[col])
```

In [75]:

```python
loan_data
```

Out[75]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
0	1	No	0	Graduate	No	5849	0.0
1	1	Yes	1	Graduate	No	4583	1508.0
2	1	Yes	0	Graduate	Yes	3000	0.0
3	1	Yes	0	Not Graduate	No	2583	2358.0
4	1	No	0	Graduate	No	6000	0.0
...	...	...	...	...	...	...	...
609	1	No	0	Graduate	No	2900	0.0
610	1	Yes	3+	Graduate	No	4106	0.0
611	1	Yes	1	Graduate	No	8072	240.0
612	1	Yes	2	Graduate	No	7583	0.0
613	0	No	0	Graduate	Yes	4583	0.0

614 rows × 12 columns

One hot-Encoder

In [77]:
```python
loan_data=pd.read_csv(r"F:\FSDS\Data Files\train_ctrUa4k.csv")
pd.get_dummies(loan_data['Gender'],prefix='Gender',dtype='int')
```

Out[77]:

	Gender_Female	Gender_Male
0	0	1
1	0	1
2	0	1
3	0	1
4	0	1
...	...	...
609	1	0
610	0	1
611	0	1
612	0	1
613	1	0

614 rows × 2 columns

In [78]:
```python
loan_data=pd.read_csv(r"F:\FSDS\Data Files\train_ctrUa4k.csv")
pd.get_dummies(loan_data, dtype='int')
```

Out[78]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Loan_ID_
0	5849	0.0	NaN	360.0	1.0	
1	4583	1508.0	128.0	360.0	1.0	
2	3000	0.0	66.0	360.0	1.0	
3	2583	2358.0	120.0	360.0	1.0	
4	6000	0.0	141.0	360.0	1.0	
...	...	...	...	...	...	
609	2900	0.0	71.0	360.0	1.0	
610	4106	0.0	40.0	180.0	1.0	
611	8072	240.0	253.0	360.0	1.0	
612	7583	0.0	187.0	360.0	1.0	
613	4583	0.0	133.0	360.0	0.0	

614 rows × 636 columns

In [ ]:
```python
# step-12: Scale the data
# Standarization
```

Standarization

In [12]:
```
loan_data
```

Out[12]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplicai
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 [13]:
```
income_data=loan_data['ApplicantIncome']
mean=income_data.mean()
std=income_data.std()
data=(income_data-mean)/std
data
```

Out[13]:
```
0      0.072931
1     -0.134302
2     -0.393427
3     -0.461686
4      0.097649
         ...
609   -0.409796
610   -0.212383
611    0.436818
612    0.356773
613   -0.134302
Name: ApplicantIncome, Length: 614, dtype: float64
```

In [14]:
```
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
ss.fit_transform(loan_data[['ApplicantIncome']])
```

```
Out[14]: array([[ 7.29908229e-02],
               [-1.34411954e-01],
               [-3.93747337e-01],
               [-4.62062470e-01],
               [ 9.77284368e-02],
               [ 2.21831139e-03],
               [-5.03018784e-01],
               [-3.87849628e-01],
               [-2.28939128e-01],
               [ 1.21845703e+00],
               [-3.60982286e-01],
               [-4.75659966e-01],
               [-3.81788093e-01],
               [-5.81654908e-01],
               [-6.72414101e-01],
               [-7.42880841e-02],
               [-2.96107483e-01],
               [-3.10196456e-01],
               [-8.46090753e-02],
               [-4.59277440e-01],
               [ 3.69678365e-01],
               [ 9.03563002e-02],
               [-4.59277440e-01],
               [-3.33951118e-01],
               [-2.76284627e-01],
               [ 6.80946355e-01],
               [-4.26676214e-01],
               [-1.92897571e-01],
               [-6.48987089e-01],
               [-2.70878394e-01],
               [-2.02727086e-01],
               [-3.66388519e-01],
               [-1.16555001e-01],
               [-3.11834708e-01],
               [ 1.16259261e+00],
               [-5.12520649e-01],
               [-5.85750540e-01],
               [-2.84475890e-01],
               [-2.02727086e-01],
               [-2.71206044e-01],
               [-2.95452182e-01],
               [-5.90337647e-01],
               [-4.92042492e-01],
               [-2.39587769e-01],
               [-1.16063525e-01],
               [-3.26578981e-01],
               [ 4.02257712e-02],
               [ 6.84037156e-02],
               [-4.51905304e-01],
               [-2.29922079e-01],
               [-5.69368014e-01],
               [-3.79658365e-01],
               [-1.92242270e-01],
               [-1.29005720e-01],
               [ 9.98767356e-01],
               [-4.41584313e-01],
               [-5.35947661e-01],
[-3.33787293e-01],
[ 4.38484973e-01],
[-3.35261720e-01],
[-4.75659966e-01],
[-3.88996405e-01],
[-4.57803013e-01],
[-7.51072104e-02],
[-2.02727086e-01],
[ 5.28403161e-02],
[-3.60982286e-01],
[ 8.75898412e-01],
[ 2.77936221e-01],
[-1.80774502e-01],
[-3.59671684e-01],
[-5.78050752e-01],
[-3.11834708e-01],
[-1.06234009e-01],
[-2.25193026e-02],
[-2.70878394e-01],
[-2.70878394e-01],
[-7.21397853e-01],
[-3.66388519e-01],
[-3.39193526e-01],
[-2.55151169e-01],
[-4.92861618e-01],
[-6.59471906e-01],
[ 9.77284368e-02],
[-2.31887982e-01],
[-4.97612551e-01],
[-3.39193526e-01],
[-4.75659966e-01],
[ 5.18104049e-01],
[ 4.77617331e-02],
[-4.00627998e-01],
[ 1.38684751e-01],
[-3.49023042e-01],
[-2.08133320e-01],
[-2.92175677e-01],
[ 2.25839789e-01],
[-4.78281170e-01],
[-5.61340576e-01],
[-1.99122931e-01],
[-5.97054482e-01],
[-1.82740405e-01],
[-9.18173867e-02],
[ 1.35099166e+00],
[-1.23108011e-01],
[-2.60065927e-01],
[-3.85228424e-01],
[ 9.85169859e-01],
[ 3.16107506e-01],
[-2.62687131e-01],
[-5.45941002e-01],
[-1.43280397e-02],
[-4.05378931e-01],
[-3.00039290e-01],
[ 3.35438886e-01],
[-5.79055583e-02],
[ 1.50384063e+00],
[-3.66388519e-01],
[-5.22513990e-01],
[ 2.69559254e-02],
[ 8.19870174e-01],
[ 4.31746259e-02],
[-2.02727086e-01],
[-5.35128535e-01],
[-4.00791823e-01],
[-1.80774502e-01],
[-2.80380259e-01],
[ 3.01430951e+00],
[-2.52038489e-01],
[ 8.37071826e-01],
[ 1.10834457e-01],
[ 2.41847704e+00],
[-5.55279042e-01],
[-4.39946060e-01],
[-3.18551544e-01],
[-8.32984733e-02],
[-2.29922079e-01],
[-1.34411954e-01],
[-3.41978556e-01],
[ 1.57199193e+00],
[-1.97157028e-01],
[-5.92161604e-02],
[ 2.21831139e-03],
[ 2.53362432e-01],
[-4.43222565e-01],
[ 1.04087045e+00],
[-5.03510260e-01],
[ 1.55020317e+00],
[-6.33259864e-01],
[ 7.53029469e-01],
[-8.90323573e-02],
[ 1.43108033e-01],
[-4.63045421e-01],
[ 6.16399204e-01],
[-5.11537698e-01],
[-3.52135722e-01],
[ 5.66762338e+00],
[ 9.77284368e-02],
[ 6.77342200e-01],
[-3.97023842e-01],
[-1.34411954e-01],
[-5.80016656e-01],
[ 4.14402661e-01],
[-3.79166889e-01],
[-2.02563261e-01],
[ 6.42119769e-01],
[-2.77922880e-01],
[-1.34411954e-01],
[-4.85653307e-01],
[-5.18746009e-01],
[ 4.25378953e-01],
[-5.87061142e-01],
[ 7.59486372e+00],
[-3.08230553e-01],
[ 4.98914615e-02],
[-1.73566190e-01],
[-3.12326184e-01],
[-5.50200459e-01],
[ 1.84370119e-02],
[-2.70878394e-01],
[-5.03018784e-01],
[ 1.63258540e-01],
[-5.71333917e-01],
[-1.31626924e-01],
[ 4.65960657e+00],
[-2.91356551e-01],
[ 5.52804426e+00],
[-5.28411699e-01],
[-4.94827521e-01],
[-7.74804887e-01],
[ 6.42938895e-01],
[-8.49367259e-02],
[ 1.08067998e+00],
[ 1.03134670e-01],
[-2.53185266e-01],
[-1.98631455e-01],
[-3.73269180e-01],
[ 4.79932764e-01],
[-5.72808344e-01],
[-3.25596030e-01],
[ 9.16854727e-01],
[-4.59277440e-01],
[-7.87113661e-02],
[-2.31232681e-01],
[-3.11834708e-01],
[-2.43519576e-01],
[-1.63081374e-01],
[-3.53773974e-01],
[-2.34017711e-01],
[-4.79100297e-01],
[-3.25268379e-01],
[ 7.53029469e-01],
[-3.23302476e-01],
[ 3.90484173e-01],
[ 4.90723352e-02],
[-3.65405568e-01],
[-2.54495868e-01],
[-8.60649323e-01],
[-2.74646375e-01],
[-6.60968212e-02],
[-1.83559531e-01],
[-5.21367213e-01],
[-2.28447652e-01],
[-3.98498270e-01],
[ 3.56244694e-01],
[ 1.38684751e-01],
[-3.52791023e-01],
[-1.09510515e-01],
[ 1.38684751e-01],
[-1.05742534e-01],
[ 1.63258540e-01],
[-4.77134394e-01],
[-2.76448452e-01],
[-3.62784363e-01],
[ 4.79932764e-01],
[-3.68354422e-01],
[ 1.58158078e-02],
[ 5.61168213e-02],
[-3.17896243e-01],
[-2.60721228e-01],
[-3.42142381e-01],
[ 6.80760651e-02],
[-4.74021714e-01],
[-3.99481221e-01],
[ 1.38684751e-01],
[-3.27234282e-01],
[ 1.05919700e-01],
[ 7.04373367e-01],
[ 1.97334194e-01],
[-4.13078718e-01],
[-5.88863219e-01],
[-6.11962581e-01],
[-3.23793952e-01],
[-4.62062470e-01],
[-4.49284100e-01],
[ 1.77693733e+00],
[-3.80149841e-01],
[ 1.05100573e-01],
[-2.51405067e-02],
[ 1.52022315e+00],
[-7.74007640e-02],
[ 1.11325933e-01],
[-5.47743080e-01],
[-3.14947388e-01],
[ 2.94318746e-01],
[-3.89018284e-02],
[-2.14358680e-01],
[-1.13933797e-01],
[-1.75368268e-01],
[-3.25268379e-01],
[-4.14061669e-01],
[-3.54920751e-01],
[ 9.40773215e-01],
[-4.21106155e-01],
[-4.56000935e-01],
[-2.46304605e-01],
[-4.34703652e-01],
[-2.31068856e-01],
[-3.76873336e-01],
[ 1.50384063e+00],
[-2.13539553e-01],
[-2.21239341e-01],
[-2.41881323e-01],
[-5.08261193e-01],
[-5.88535569e-01],
[ 2.50055350e+00],
[-3.67862946e-01],
[-4.59277440e-01],
[-2.78414356e-01],
[-2.09607747e-01],
[ 6.72427442e-01],
[-3.81460443e-01],
[-1.64391976e-01],
[-3.68682073e-01],
[ 2.21831139e-03],
[-4.94827521e-01],
[-1.61770772e-01],
[ 2.41075538e-01],
[-1.20814457e-01],
[-6.60968212e-02],
[-5.55279042e-01],
[-5.90337647e-01],
[-4.14225494e-01],
[-6.60968212e-02],
[-6.19007067e-01],
[-2.29922079e-01],
[-5.57572595e-01],
[-2.68912491e-01],
[-4.92042492e-01],
[ 2.42945333e+00],
[ 3.70825142e-01],
[-4.07344834e-01],
[-4.05706581e-01],
[-4.74513189e-01],
[ 5.61168213e-02],
[-4.80083248e-01],
[-3.28381059e-01],
[-2.76284627e-01],
[-5.48070730e-01],
[-3.05117873e-01],
[ 7.53029469e-01],
[-4.92042492e-01],
[-1.73893841e-01],
[-2.95288357e-01],
[-3.66552344e-01],
[ 1.57215576e+00],
[ 5.34486575e-01],
[-7.96943176e-02],
[ 6.79122399e-02],
[-1.75368268e-01],
[-4.75659966e-01],
[-1.67013180e-01],
[-4.04395979e-01],
[-4.21106155e-01],
[ 9.49097726e+00],
[ 7.25670651e-01],
[ 1.63072836e-02],
[-2.51405067e-02],
[-4.75659966e-01],
[-5.85422889e-01],
[-2.03710038e-01],
[-4.51577653e-01],
[-4.95646648e-01],
[-1.39162886e-01],
[-3.65405568e-01],
[-4.62062470e-01],
[-4.75823791e-01],
[-3.08066727e-01],
[-3.80149841e-01],
[ 1.52282248e-01],
[-4.55181809e-01],
[ 6.02801707e-01],
[ 5.48247896e-01],
[-4.48464973e-01],
[ 1.58158078e-02],
[-4.88274511e-01],
[-2.60557402e-01],
[ 4.79932764e-01],
[-2.50400236e-01],
[-3.93747337e-01],
[-3.87380031e-02],
[-1.11476418e-01],
[-6.60968212e-02],
[-1.07053136e-01],
[-3.91617609e-01],
[ 2.32392799e-01],
[ 1.33114693e-01],
[-4.75659966e-01],
[-4.57824892e-02],
[ 1.50971646e-01],
[ 2.34704923e+00],
[ 1.69649913e+00],
[-3.57891485e-02],
[ 6.42119769e-01],
[-3.83590171e-01],
[-4.32410098e-01],
[-9.60768434e-02],
[ 5.48247896e-01],
[-1.79136249e-01],
[-3.82443394e-01],
[-2.04114532e-03],
[-3.39193526e-01],
[ 8.80627466e-02],
[ 9.77284368e-02],
[ 2.88912513e-01],
[-1.37196983e-01],
[-2.84475890e-01],
[-5.00889056e-01],
[-3.92109085e-01],
[-5.03018784e-01],
[ 1.38499047e-02],
[ 6.16563029e-01],
[ 6.71772141e-01],
[-4.62062470e-01],
[-5.58719372e-01],
[-3.77364811e-01],
[-3.48531566e-01],
[-3.64258791e-01],
[-3.88341104e-01],
[-2.45976954e-01],
[-6.39485224e-01],
[-4.11931941e-01],
[-4.33884525e-01],
[-4.75659966e-01],
[-5.63634130e-01],
[ 3.33800634e-01],
[-1.40309663e-01],
[-5.30213777e-01],
[-5.22677815e-01],
[ 4.74526530e-01],
[ 1.23846228e+01],
[-2.51710839e-01],
[ 1.39667703e-01],
[ 1.13455662e-01],
[-5.16124805e-01],
[-5.33162632e-01],
[-3.94566463e-01],
[-4.59277440e-01],
[-6.23102698e-01],
[-7.17302222e-01],
[-3.53446324e-01],
[ 6.97143177e-02],
[-4.39618409e-01],
[-5.87061142e-01],
[ 3.02510009e-01],
[ 1.55249673e+00],
[-4.48464973e-01],
[-1.30643973e-01],
[ 8.70797950e-02],
[-4.06853358e-01],
[-4.40109885e-01],
[ 5.27605914e-01],
[ 1.79641066e-01],
[ 1.22419091e+00],
[-4.87946861e-01],
[-2.70878394e-01],
[ 7.60729256e-01],
[-5.69695664e-01],
[-5.22677815e-01],
[ 8.21180776e-01],
[ 2.84816881e-01],
[-2.85622667e-01],
[ 4.09160252e-01],
[-1.14097622e-01],
[ 5.29410180e+00],
[ 3.16107506e-01],
[-3.17404767e-01],
[-1.23108011e-01],
[-3.05445523e-01],
[-3.38046749e-01],
[-4.31590972e-01],
[-5.06950590e-01],
[-5.64453256e-01],
[-2.38440993e-01],
[-4.78444996e-01],
[ 2.75478842e-01],
[-2.53021441e-01],
[-1.80610677e-01],
[-2.77759055e-01],
[-1.71927938e-01],
[ 4.80096589e-01],
[-5.43975099e-01],
[ 3.82784386e-01],
[-3.91289958e-01],
[-3.48061969e-02],
[-2.02727086e-01],
[ 9.77284368e-02],
[-4.02430076e-01],
[ 1.84934810e+00],
[-8.50819807e-01],
[-1.75368268e-01],
[-3.20025971e-01],
[-4.50594702e-01],
[-1.16718826e-01],
[-4.75659966e-01],
[ 2.10582161e-02],
[ 1.82198928e+00],
[ 2.12406117e-01],
[-4.14553145e-01],
[ 1.84525246e+00],
[-4.02430076e-01],
[-1.72583239e-01],
[-3.78183938e-01],
[-5.43975099e-01],
[ 8.89495909e-01],
[ 4.79932764e-01],
[-5.64453256e-01],
[-3.04134921e-01],
[ 2.11818534e+00],
[-1.34411954e-01],
[-4.86308608e-01],
[-4.43058740e-01],
[-1.15430103e-02],
[-2.80544084e-01],
[ 1.94289232e+00],
[-2.95943658e-01],
[-3.40340303e-01],
[-4.59277440e-01],
[-1.27531293e-01],
[-4.10948989e-01],
[ 1.44090985e-01],
[-7.79555820e-01],
[-3.67699121e-01],
[-8.82132310e-02],
[-2.21730816e-01],
[-2.60393577e-01],
[-3.10196456e-01],
[ 2.52774849e+00],
[-2.98237212e-01],
[-4.79100297e-01],
[ 1.28742746e+00],
[-2.95779833e-01],
[ 1.08377079e-01],
[-3.47384789e-01],
       [-5.36275312e-01],
       [ 6.74207641e-02],
       [-3.17404767e-01],
       [-5.52494012e-01],
       [-3.81624268e-01],
       [-1.18029428e-01],
       [-3.28217234e-01],
       [-5.26118146e-01],
       [-4.75659966e-01],
       [ 4.48128785e-02],
       [ 4.16860039e-01],
       [-1.18520904e-01],
       [ 1.98171890e+00],
       [-2.66782762e-01],
       [-1.94066227e-02],
       [-4.46335245e-01],
       [ 2.26003614e-01],
       [-7.17302222e-01],
       [-1.83887182e-01],
       [-2.97418085e-01],
       [ 9.57811041e-01],
       [ 2.09066270e+00],
       [-4.67468703e-01],
       [ 1.19517196e-01],
       [-2.92667153e-01],
       [-4.07344834e-01],
       [ 1.66043569e-01],
       [-1.30316322e-01],
       [-5.34964710e-01],
       [-2.86933269e-01],
       [-5.18418359e-01],
       [-3.90962308e-01],
       [-4.31754797e-01],
       [-3.35097895e-01],
       [-4.71400509e-01],
       [-6.60968212e-02],
       [-4.28969768e-01],
       [ 2.01429825e-01],
       [-4.76970568e-01],
       [-3.39193526e-01],
       [-4.83195928e-01],
       [-2.96598959e-01],
       [ 1.05733996e-02],
       [-4.48301148e-01],
       [ 7.75801180e-01],
       [-2.48434333e-01],
       [-2.00433533e-01],
       [-2.83165288e-01],
       [ 2.30674822e+00],
       [ 8.51138919e-02],
       [ 6.49633852e-02],
       [ 5.56275334e-01],
       [-1.53415684e-01],
       [-3.39193526e-01],
       [-3.28217234e-01],
       [-4.95646648e-01],
       [-3.66552344e-01],
       [-3.25432205e-01],
       [-4.70930913e-02],
       [ 1.84508864e+00],
       [ 1.18206594e-01],
       [ 1.64241492e-01],
       [-3.67699121e-01],
       [-3.79494540e-01],
       [-3.56231353e-01],
       [-5.93286501e-01],
       [-3.63931140e-01],
       [ 1.86194076e-01],
       [-5.84439938e-01],
       [-3.66552344e-01],
       [-5.77231626e-01],
       [-4.28642117e-01],
       [-1.83559531e-01],
       [-5.08916494e-01],
       [-5.30541428e-01],
       [-1.07053136e-01],
       [-4.38635458e-01],
       [-3.93747337e-01],
       [ 9.77284368e-02],
       [ 6.47689828e-01],
       [-2.53021441e-01],
       [ 1.75564005e+00],
       [-2.57280897e-01],
       [ 1.60473511e-01],
       [-3.95877066e-01],
       [ 7.46967934e-01],
       [ 6.16868800e-02],
       [-8.17071804e-01],
       [-4.11112815e-01],
       [ 4.90723352e-02],
       [-2.83001463e-01],
       [ 1.08067998e+00],
       [-4.92042492e-01],
       [-3.28217234e-01],
       [-2.32051808e-01],
       [-3.55739877e-01],
       [-4.10129863e-01],
       [-2.12556602e-01],
       [ 4.37174371e-01],
       [ 3.57063820e-01],
       [-1.34411954e-01]])
```

In [16]:

```
loan_data[['ApplicantIncome']]
```

Out[16]:

	ApplicantIncome
0	5849
1	4583
2	3000
3	2583
4	6000
...	...
609	2900
610	4106
611	8072
612	7583
613	4583

614 rows × 1 columns

In [17]:

```
d=loan_data['ApplicantIncome'].values.reshape(-1,1)
ss.fit_transform(d)
```

```
Out[17]: array([[ 7.29908229e-02],
               [-1.34411954e-01],
               [-3.93747337e-01],
               [-4.62062470e-01],
               [ 9.77284368e-02],
               [ 2.21831139e-03],
               [-5.03018784e-01],
               [-3.87849628e-01],
               [-2.28939128e-01],
               [ 1.21845703e+00],
               [-3.60982286e-01],
               [-4.75659966e-01],
               [-3.81788093e-01],
               [-5.81654908e-01],
               [-6.72414101e-01],
               [-7.42880841e-02],
               [-2.96107483e-01],
               [-3.10196456e-01],
               [-8.46090753e-02],
               [-4.59277440e-01],
               [ 3.69678365e-01],
               [ 9.03563002e-02],
               [-4.59277440e-01],
               [-3.33951118e-01],
               [-2.76284627e-01],
               [ 6.80946355e-01],
               [-4.26676214e-01],
               [-1.92897571e-01],
               [-6.48987089e-01],
               [-2.70878394e-01],
               [-2.02727086e-01],
               [-3.66388519e-01],
               [-1.16555001e-01],
               [-3.11834708e-01],
               [ 1.16259261e+00],
               [-5.12520649e-01],
               [-5.85750540e-01],
               [-2.84475890e-01],
               [-2.02727086e-01],
               [-2.71206044e-01],
               [-2.95452182e-01],
               [-5.90337647e-01],
               [-4.92042492e-01],
               [-2.39587769e-01],
               [-1.16063525e-01],
               [-3.26578981e-01],
               [ 4.02257712e-02],
               [ 6.84037156e-02],
               [-4.51905304e-01],
               [-2.29922079e-01],
               [-5.69368014e-01],
               [-3.79658365e-01],
               [-1.92242270e-01],
               [-1.29005720e-01],
               [ 9.98767356e-01],
               [-4.41584313e-01],
               [-5.35947661e-01],
       [-3.33787293e-01],
       [ 4.38484973e-01],
       [-3.35261720e-01],
       [-4.75659966e-01],
       [-3.88996405e-01],
       [-4.57803013e-01],
       [-7.51072104e-02],
       [-2.02727086e-01],
       [ 5.28403161e-02],
       [-3.60982286e-01],
       [ 8.75898412e-01],
       [ 2.77936221e-01],
       [-1.80774502e-01],
       [-3.59671684e-01],
       [-5.78050752e-01],
       [-3.11834708e-01],
       [-1.06234009e-01],
       [-2.25193026e-02],
       [-2.70878394e-01],
       [-2.70878394e-01],
       [-7.21397853e-01],
       [-3.66388519e-01],
       [-3.39193526e-01],
       [-2.55151169e-01],
       [-4.92861618e-01],
       [-6.59471906e-01],
       [ 9.77284368e-02],
       [-2.31887982e-01],
       [-4.97612551e-01],
       [-3.39193526e-01],
       [-4.75659966e-01],
       [ 5.18104049e-01],
       [ 4.77617331e-02],
       [-4.00627998e-01],
       [ 1.38684751e-01],
       [-3.49023042e-01],
       [-2.08133320e-01],
       [-2.92175677e-01],
       [ 2.25839789e-01],
       [-4.78281170e-01],
       [-5.61340576e-01],
       [-1.99122931e-01],
       [-5.97054482e-01],
       [-1.82740405e-01],
       [-9.18173867e-02],
       [ 1.35099166e+00],
       [-1.23108011e-01],
       [-2.60065927e-01],
       [-3.85228424e-01],
       [ 9.85169859e-01],
       [ 3.16107506e-01],
       [-2.62687131e-01],
       [-5.45941002e-01],
       [-1.43280397e-02],
       [-4.05378931e-01],
       [-3.00039290e-01],
       [ 3.35438886e-01],
[-5.79055583e-02],
[ 1.50384063e+00],
[-3.66388519e-01],
[-5.22513990e-01],
[ 2.69559254e-02],
[ 8.19870174e-01],
[ 4.31746259e-02],
[-2.02727086e-01],
[-5.35128535e-01],
[-4.00791823e-01],
[-1.80774502e-01],
[-2.80380259e-01],
[ 3.01430951e+00],
[-2.52038489e-01],
[ 8.37071826e-01],
[ 1.10834457e-01],
[ 2.41847704e+00],
[-5.55279042e-01],
[-4.39946060e-01],
[-3.18551544e-01],
[-8.32984733e-02],
[-2.29922079e-01],
[-1.34411954e-01],
[-3.41978556e-01],
[ 1.57199193e+00],
[-1.97157028e-01],
[-5.92161604e-02],
[ 2.21831139e-03],
[ 2.53362432e-01],
[-4.43222565e-01],
[ 1.04087045e+00],
[-5.03510260e-01],
[ 1.55020317e+00],
[-6.33259864e-01],
[ 7.53029469e-01],
[-8.90323573e-02],
[ 1.43108033e-01],
[-4.63045421e-01],
[ 6.16399204e-01],
[-5.11537698e-01],
[-3.52135722e-01],
[ 5.66762338e+00],
[ 9.77284368e-02],
[ 6.77342200e-01],
[-3.97023842e-01],
[-1.34411954e-01],
[-5.80016656e-01],
[ 4.14402661e-01],
[-3.79166889e-01],
[-2.02563261e-01],
[ 6.42119769e-01],
[-2.77922880e-01],
[-1.34411954e-01],
[-4.85653307e-01],
[-5.18746009e-01],
[ 4.25378953e-01],
[-5.87061142e-01],
[ 7.59486372e+00],
[-3.08230553e-01],
[ 4.98914615e-02],
[-1.73566190e-01],
[-3.12326184e-01],
[-5.50200459e-01],
[ 1.84370119e-02],
[-2.70878394e-01],
[-5.03018784e-01],
[ 1.63258540e-01],
[-5.71333917e-01],
[-1.31626924e-01],
[ 4.65960657e+00],
[-2.91356551e-01],
[ 5.52804426e+00],
[-5.28411699e-01],
[-4.94827521e-01],
[-7.74804887e-01],
[ 6.42938895e-01],
[-8.49367259e-02],
[ 1.08067998e+00],
[ 1.03134670e-01],
[-2.53185266e-01],
[-1.98631455e-01],
[-3.73269180e-01],
[ 4.79932764e-01],
[-5.72808344e-01],
[-3.25596030e-01],
[ 9.16854727e-01],
[-4.59277440e-01],
[-7.87113661e-02],
[-2.31232681e-01],
[-3.11834708e-01],
[-2.43519576e-01],
[-1.63081374e-01],
[-3.53773974e-01],
[-2.34017711e-01],
[-4.79100297e-01],
[-3.25268379e-01],
[ 7.53029469e-01],
[-3.23302476e-01],
[ 3.90484173e-01],
[ 4.90723352e-02],
[-3.65405568e-01],
[-2.54495868e-01],
[-8.60649323e-01],
[-2.74646375e-01],
[-6.60968212e-02],
[-1.83559531e-01],
[-5.21367213e-01],
[-2.28447652e-01],
[-3.98498270e-01],
[ 3.56244694e-01],
[ 1.38684751e-01],
[-3.52791023e-01],
[-1.09510515e-01],
[ 1.38684751e-01],
[-1.05742534e-01],
[ 1.63258540e-01],
[-4.77134394e-01],
[-2.76448452e-01],
[-3.62784363e-01],
[ 4.79932764e-01],
[-3.68354422e-01],
[ 1.58158078e-02],
[ 5.61168213e-02],
[-3.17896243e-01],
[-2.60721228e-01],
[-3.42142381e-01],
[ 6.80760651e-02],
[-4.74021714e-01],
[-3.99481221e-01],
[ 1.38684751e-01],
[-3.27234282e-01],
[ 1.05919700e-01],
[ 7.04373367e-01],
[ 1.97334194e-01],
[-4.13078718e-01],
[-5.88863219e-01],
[-6.11962581e-01],
[-3.23793952e-01],
[-4.62062470e-01],
[-4.49284100e-01],
[ 1.77693733e+00],
[-3.80149841e-01],
[ 1.05100573e-01],
[-2.51405067e-02],
[ 1.52022315e+00],
[-7.74007640e-02],
[ 1.11325933e-01],
[-5.47743080e-01],
[-3.14947388e-01],
[ 2.94318746e-01],
[-3.89018284e-02],
[-2.14358680e-01],
[-1.13933797e-01],
[-1.75368268e-01],
[-3.25268379e-01],
[-4.14061669e-01],
[-3.54920751e-01],
[ 9.40773215e-01],
[-4.21106155e-01],
[-4.56000935e-01],
[-2.46304605e-01],
[-4.34703652e-01],
[-2.31068856e-01],
[-3.76873336e-01],
[ 1.50384063e+00],
[-2.13539553e-01],
[-2.21239341e-01],
[-2.41881323e-01],
[-5.08261193e-01],
[-5.88535569e-01],
[ 2.50055350e+00],
[-3.67862946e-01],
[-4.59277440e-01],
[-2.78414356e-01],
[-2.09607747e-01],
[ 6.72427442e-01],
[-3.81460443e-01],
[-1.64391976e-01],
[-3.68682073e-01],
[ 2.21831139e-03],
[-4.94827521e-01],
[-1.61770772e-01],
[ 2.41075538e-01],
[-1.20814457e-01],
[-6.60968212e-02],
[-5.55279042e-01],
[-5.90337647e-01],
[-4.14225494e-01],
[-6.60968212e-02],
[-6.19007067e-01],
[-2.29922079e-01],
[-5.57572595e-01],
[-2.68912491e-01],
[-4.92042492e-01],
[ 2.42945333e+00],
[ 3.70825142e-01],
[-4.07344834e-01],
[-4.05706581e-01],
[-4.74513189e-01],
[ 5.61168213e-02],
[-4.80083248e-01],
[-3.28381059e-01],
[-2.76284627e-01],
[-5.48070730e-01],
[-3.05117873e-01],
[ 7.53029469e-01],
[-4.92042492e-01],
[-1.73893841e-01],
[-2.95288357e-01],
[-3.66552344e-01],
[ 1.57215576e+00],
[ 5.34486575e-01],
[-7.96943176e-02],
[ 6.79122399e-02],
[-1.75368268e-01],
[-4.75659966e-01],
[-1.67013180e-01],
[-4.04395979e-01],
[-4.21106155e-01],
[ 9.49097726e+00],
[ 7.25670651e-01],
[ 1.63072836e-02],
[-2.51405067e-02],
[-4.75659966e-01],
[-5.85422889e-01],
[-2.03710038e-01],
[-4.51577653e-01],
[-4.95646648e-01],
[-1.39162886e-01],
[-3.65405568e-01],
[-4.62062470e-01],
[-4.75823791e-01],
[-3.08066727e-01],
[-3.80149841e-01],
[ 1.52282248e-01],
[-4.55181809e-01],
[ 6.02801707e-01],
[ 5.48247896e-01],
[-4.48464973e-01],
[ 1.58158078e-02],
[-4.88274511e-01],
[-2.60557402e-01],
[ 4.79932764e-01],
[-2.50400236e-01],
[-3.93747337e-01],
[-3.87380031e-02],
[-1.11476418e-01],
[-6.60968212e-02],
[-1.07053136e-01],
[-3.91617609e-01],
[ 2.32392799e-01],
[ 1.33114693e-01],
[-4.75659966e-01],
[-4.57824892e-02],
[ 1.50971646e-01],
[ 2.34704923e+00],
[ 1.69649913e+00],
[-3.57891485e-02],
[ 6.42119769e-01],
[-3.83590171e-01],
[-4.32410098e-01],
[-9.60768434e-02],
[ 5.48247896e-01],
[-1.79136249e-01],
[-3.82443394e-01],
[-2.04114532e-03],
[-3.39193526e-01],
[ 8.80627466e-02],
[ 9.77284368e-02],
[ 2.88912513e-01],
[-1.37196983e-01],
[-2.84475890e-01],
[-5.00889056e-01],
[-3.92109085e-01],
[-5.03018784e-01],
[ 1.38499047e-02],
[ 6.16563029e-01],
[ 6.71772141e-01],
[-4.62062470e-01],
[-5.58719372e-01],
[-3.77364811e-01],
[-3.48531566e-01],
[-3.64258791e-01],
[-3.88341104e-01],
[-2.45976954e-01],
[-6.39485224e-01],
[-4.11931941e-01],
[-4.33884525e-01],
[-4.75659966e-01],
[-5.63634130e-01],
[ 3.33800634e-01],
[-1.40309663e-01],
[-5.30213777e-01],
[-5.22677815e-01],
[ 4.74526530e-01],
[ 1.23846228e+01],
[-2.51710839e-01],
[ 1.39667703e-01],
[ 1.13455662e-01],
[-5.16124805e-01],
[-5.33162632e-01],
[-3.94566463e-01],
[-4.59277440e-01],
[-6.23102698e-01],
[-7.17302222e-01],
[-3.53446324e-01],
[ 6.97143177e-02],
[-4.39618409e-01],
[-5.87061142e-01],
[ 3.02510009e-01],
[ 1.55249673e+00],
[-4.48464973e-01],
[-1.30643973e-01],
[ 8.70797950e-02],
[-4.06853358e-01],
[-4.40109885e-01],
[ 5.27605914e-01],
[ 1.79641066e-01],
[ 1.22419091e+00],
[-4.87946861e-01],
[-2.70878394e-01],
[ 7.60729256e-01],
[-5.69695664e-01],
[-5.22677815e-01],
[ 8.21180776e-01],
[ 2.84816881e-01],
[-2.85622667e-01],
[ 4.09160252e-01],
[-1.14097622e-01],
[ 5.29410180e+00],
[ 3.16107506e-01],
[-3.17404767e-01],
[-1.23108011e-01],
[-3.05445523e-01],
[-3.38046749e-01],
[-4.31590972e-01],
[-5.06950590e-01],
[-5.64453256e-01],
[-2.38440993e-01],
[-4.78444996e-01],
[ 2.75478842e-01],
[-2.53021441e-01],
[-1.80610677e-01],
[-2.77759055e-01],
[-1.71927938e-01],
[ 4.80096589e-01],
[-5.43975099e-01],
[ 3.82784386e-01],
[-3.91289958e-01],
[-3.48061969e-02],
[-2.02727086e-01],
[ 9.77284368e-02],
[-4.02430076e-01],
[ 1.84934810e+00],
[-8.50819807e-01],
[-1.75368268e-01],
[-3.20025971e-01],
[-4.50594702e-01],
[-1.16718826e-01],
[-4.75659966e-01],
[ 2.10582161e-02],
[ 1.82198928e+00],
[ 2.12406117e-01],
[-4.14553145e-01],
[ 1.84525246e+00],
[-4.02430076e-01],
[-1.72583239e-01],
[-3.78183938e-01],
[-5.43975099e-01],
[ 8.89495909e-01],
[ 4.79932764e-01],
[-5.64453256e-01],
[-3.04134921e-01],
[ 2.11818534e+00],
[-1.34411954e-01],
[-4.86308608e-01],
[-4.43058740e-01],
[-1.15430103e-02],
[-2.80544084e-01],
[ 1.94289232e+00],
[-2.95943658e-01],
[-3.40340303e-01],
[-4.59277440e-01],
[-1.27531293e-01],
[-4.10948989e-01],
[ 1.44090985e-01],
[-7.79555820e-01],
[-3.67699121e-01],
[-8.82132310e-02],
[-2.21730816e-01],
[-2.60393577e-01],
[-3.10196456e-01],
[ 2.52774849e+00],
[-2.98237212e-01],
[-4.79100297e-01],
[ 1.28742746e+00],
[-2.95779833e-01],
[ 1.08377079e-01],
[-3.47384789e-01],
[-5.36275312e-01],
[ 6.74207641e-02],
[-3.17404767e-01],
[-5.52494012e-01],
[-3.81624268e-01],
[-1.18029428e-01],
[-3.28217234e-01],
[-5.26118146e-01],
[-4.75659966e-01],
[ 4.48128785e-02],
[ 4.16860039e-01],
[-1.18520904e-01],
[ 1.98171890e+00],
[-2.66782762e-01],
[-1.94066227e-02],
[-4.46335245e-01],
[ 2.26003614e-01],
[-7.17302222e-01],
[-1.83887182e-01],
[-2.97418085e-01],
[ 9.57811041e-01],
[ 2.09066270e+00],
[-4.67468703e-01],
[ 1.19517196e-01],
[-2.92667153e-01],
[-4.07344834e-01],
[ 1.66043569e-01],
[-1.30316322e-01],
[-5.34964710e-01],
[-2.86933269e-01],
[-5.18418359e-01],
[-3.90962308e-01],
[-4.31754797e-01],
[-3.35097895e-01],
[-4.71400509e-01],
[-6.60968212e-02],
[-4.28969768e-01],
[ 2.01429825e-01],
[-4.76970568e-01],
[-3.39193526e-01],
[-4.83195928e-01],
[-2.96598959e-01],
[ 1.05733996e-02],
[-4.48301148e-01],
[ 7.75801180e-01],
[-2.48434333e-01],
[-2.00433533e-01],
[-2.83165288e-01],
[ 2.30674822e+00],
[ 8.51138919e-02],
[ 6.49633852e-02],
[ 5.56275334e-01],
[-1.53415684e-01],
[-3.39193526e-01],
[-3.28217234e-01],
[-4.95646648e-01],
[-3.66552344e-01],
       [-3.25432205e-01],
       [-4.70930913e-02],
       [ 1.84508864e+00],
       [ 1.18206594e-01],
       [ 1.64241492e-01],
       [-3.67699121e-01],
       [-3.79494540e-01],
       [-3.56231353e-01],
       [-5.93286501e-01],
       [-3.63931140e-01],
       [ 1.86194076e-01],
       [-5.84439938e-01],
       [-3.66552344e-01],
       [-5.77231626e-01],
       [-4.28642117e-01],
       [-1.83559531e-01],
       [-5.08916494e-01],
       [-5.30541428e-01],
       [-1.07053136e-01],
       [-4.38635458e-01],
       [-3.93747337e-01],
       [ 9.77284368e-02],
       [ 6.47689828e-01],
       [-2.53021441e-01],
       [ 1.75564005e+00],
       [-2.57280897e-01],
       [ 1.60473511e-01],
       [-3.95877066e-01],
       [ 7.46967934e-01],
       [ 6.16868800e-02],
       [-8.17071804e-01],
       [-4.11112815e-01],
       [ 4.90723352e-02],
       [-2.83001463e-01],
       [ 1.08067998e+00],
       [-4.92042492e-01],
       [-3.28217234e-01],
       [-2.32051808e-01],
       [-3.55739877e-01],
       [-4.10129863e-01],
       [-2.12556602e-01],
       [ 4.37174371e-01],
       [ 3.57063820e-01],
       [-1.34411954e-01]])
```

In [19]:

```
loan_data=pd.read_csv(r"F:\FSDS\Data Files\train_ctrUa4k.csv")
income_data=loan_data['ApplicantIncome']
mean=income_data.mean()
std=income_data.std()

loan_data['ApplicantIncome_z']=(income_data-mean)/std
loan_data
```

Out[19]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapplica
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 × 14 columns

In [20]:

```
loan_data[['ApplicantIncome','ApplicantIncome_z']]
```

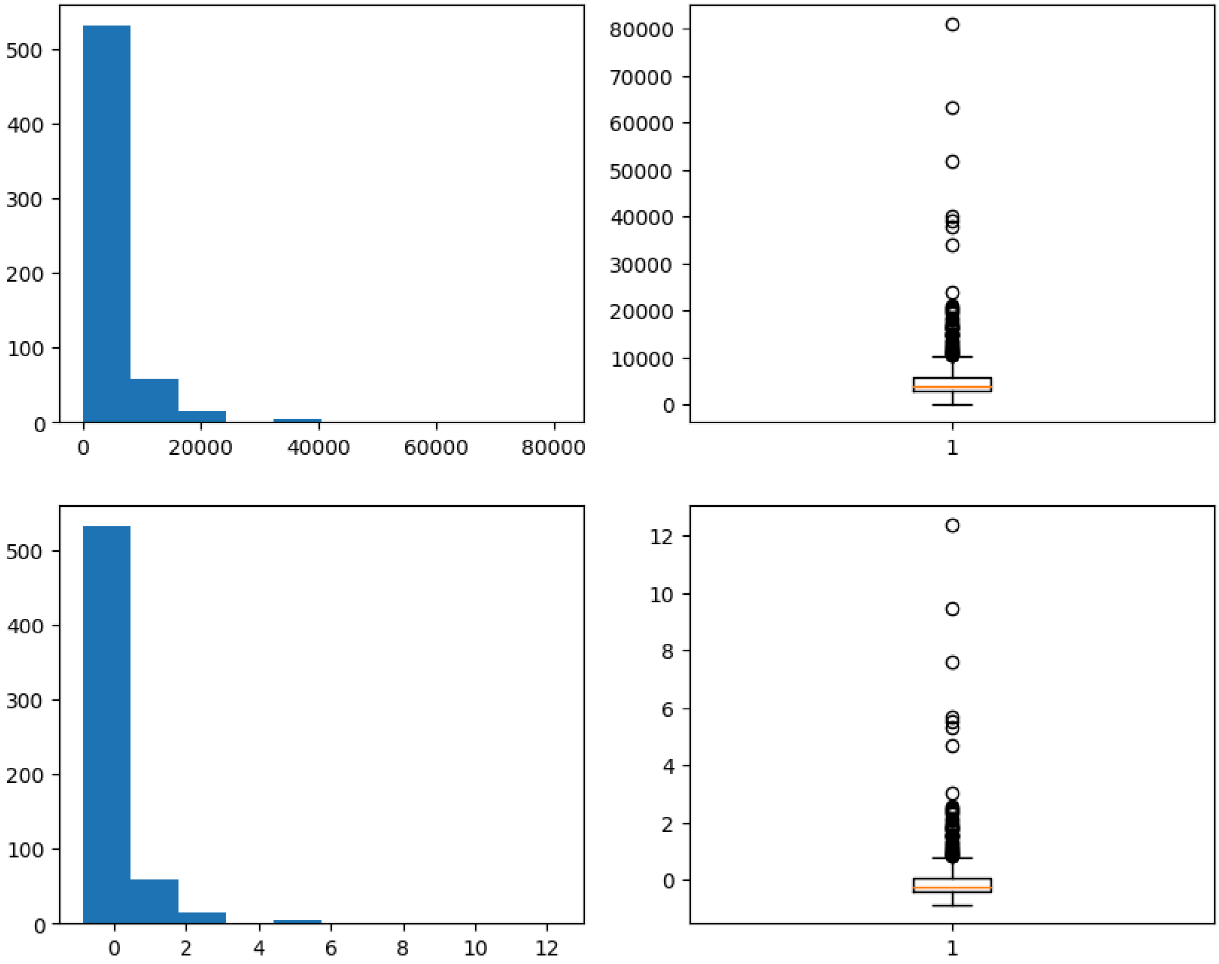
Out[20]:

	ApplicantIncome	ApplicantIncome_z
0	5849	0.072931
1	4583	-0.134302
2	3000	-0.393427
3	2583	-0.461686
4	6000	0.097649
...	...	...
609	2900	-0.409796
610	4106	-0.212383
611	8072	0.436818
612	7583	0.356773
613	4583	-0.134302

614 rows × 2 columns

In [21]:

```
plt.figure(figsize=(10,8))
plt.subplot(2,2,1).hist(loan_data['ApplicantIncome'])
plt.subplot(2,2,2).boxplot(loan_data['ApplicantIncome'])
plt.subplot(2,2,3).hist(loan_data['ApplicantIncome_z'])
plt.subplot(2,2,4).boxplot(loan_data['ApplicantIncome_z'])
plt.show()
```



500
400
300
200
100
0
0
20000
40000
60000
80000
80000
70000
60000
50000
40000
30000
20000
10000
0
1
500
400
300
200
100
0
0
2
4
6
8
10
12
12
10
8
6
4
2
0
1

In [ ]: