```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


data =pd.read_csv('/test_loan.csv')
```

Double-click (or enter) to edit

```
#to show the data
data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	C
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	

```
#to show last five
data.tail()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome
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

```
#to know no.of rows and colums
data.shape
```

```
(367, 12)
```

```
#to show columns
data.columns
```

```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area'],
      dtype='object')
```

```
#data information
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 367 entries, 0 to 366
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            367 non-null    object 
 1   Gender             356 non-null    object 
 2   Married            367 non-null    object 
 3   Dependents         357 non-null    object 
 4   Education          367 non-null    object 
 5   Self_Employed      344 non-null    object 
 6   ApplicantIncome    367 non-null    int64  
 7   CoapplicantIncome  367 non-null    int64  
 8   LoanAmount         362 non-null    float64
 9   Loan_Amount_Term   361 non-null    float64
 10  Credit_History     338 non-null    float64
 11  Property_Area      367 non-null    object 
dtypes: float64(3), int64(2), object(7)
memory usage: 34.5+ KB
```

```
#statistical summary
data.describe()
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_Histo
count	367.000000	367.000000	362.000000	361.000000	338.0000
mean	4805.599455	1569.577657	136.132597	342.537396	0.8254
std	4910.685399	2334.232099	61.366652	65.156643	0.3801
min	0.000000	0.000000	28.000000	6.000000	0.0000
25%	2864.000000	0.000000	100.250000	360.000000	1.0000
50%	3786.000000	1025.000000	125.000000	360.000000	1.0000
75%	5060.000000	2430.500000	158.000000	360.000000	1.0000
max	72529.000000	24000.000000	550.000000	480.000000	1.0000

```
#a)missing value handling
data.isna().sum()
```

```
Loan_ID              0
Gender              11
Married              0
Dependents          10
Education            0
Self_Employed       23
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           5
Loan_Amount_Term     6
Credit_History      29
Property_Area        0
dtype: int64
```

```
data.dtypes
```

```
Loan_ID              object
Gender               object
```

```
Gender               object
Married              object
Dependents           object
Education            object
Self_Employed        object
ApplicantIncome       int64
CoapplicantIncome     int64
LoanAmount          float64
Loan_Amount_Term    float64
Credit_History      float64
Property_Area        object
dtype: object
```

```
#mssing values in 3-categorical and 3-numerical columns
#numerical colums-mean/median
#categorical columns-mode
```

```
#values in Gender
data['Gender'].unique()
```

```
array(['Male', 'Female', nan], dtype=object)
```

```
data['Gender'].value_counts()
```

```
Male      286
Female     70
Name: Gender, dtype: int64
```

```
data['Gender']=data['Gender'].fillna('Male')
```

```
data.isna().sum()
```

```
Loan_ID               0
Gender                0
Married               0
Dependents           10
Education             0
Self_Employed        23
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            5
Loan_Amount_Term      6
Credit_History       29
Property_Area         0
dtype: int64
```

```
data['Dependents'].unique()
```

```
array(['0', '1', '2', '3+', nan], dtype=object)
```

```
data['Dependents'].mode()
```

```
0    0
Name: Dependents, dtype: object
```

```
data['Dependents']=data['Dependents'].fillna(data['Dependents'].mode()[0])
```

```
data.isna().sum()
```

```
Loan_ID               0
Gender                0
Married               0
Dependents            0
Education             0
Self_Employed        23
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            5
Loan_Amount_Term      6
Credit_History       29
Property_Area         0
dtype: int64
```

```
data['Self_Employed'].unique()
```

```
array(['No', 'Yes', nan], dtype=object)
```

```
data['Self_Employed'].mode()
```

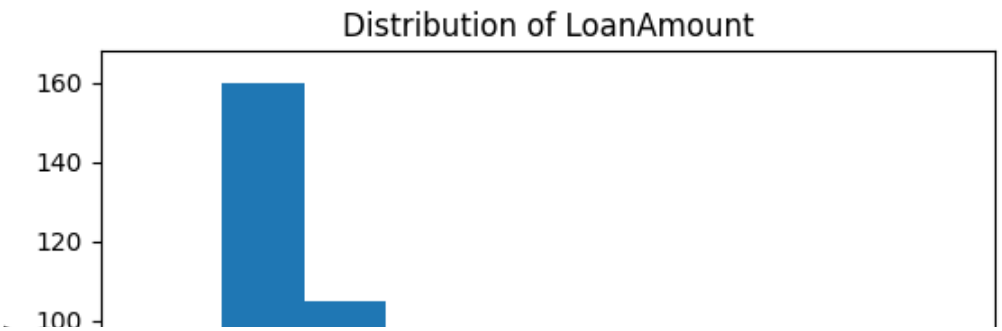
```
0    No
Name: Self_Employed, dtype: object
```

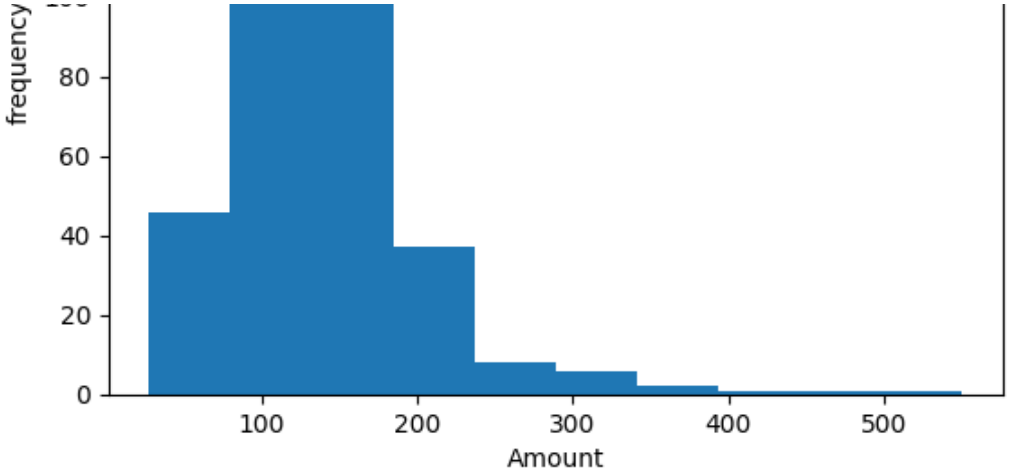
```
data['Self_Employed']=data['Self_Employed'].fillna(data['Self_Employed'].mode()[0])
```

```
data.isna().sum()
```

```
Loan_ID               0
Gender                0
Married               0
Dependents            0
Education             0
Self_Employed         0
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            5
Loan_Amount_Term      6
Credit_History       29
Property_Area         0
dtype: int64
```

```
plt.hist(data['LoanAmount'])
plt.title('Distribution of LoanAmount')
plt.xlabel('Amount')
plt.ylabel('frequency')
plt.show()
```



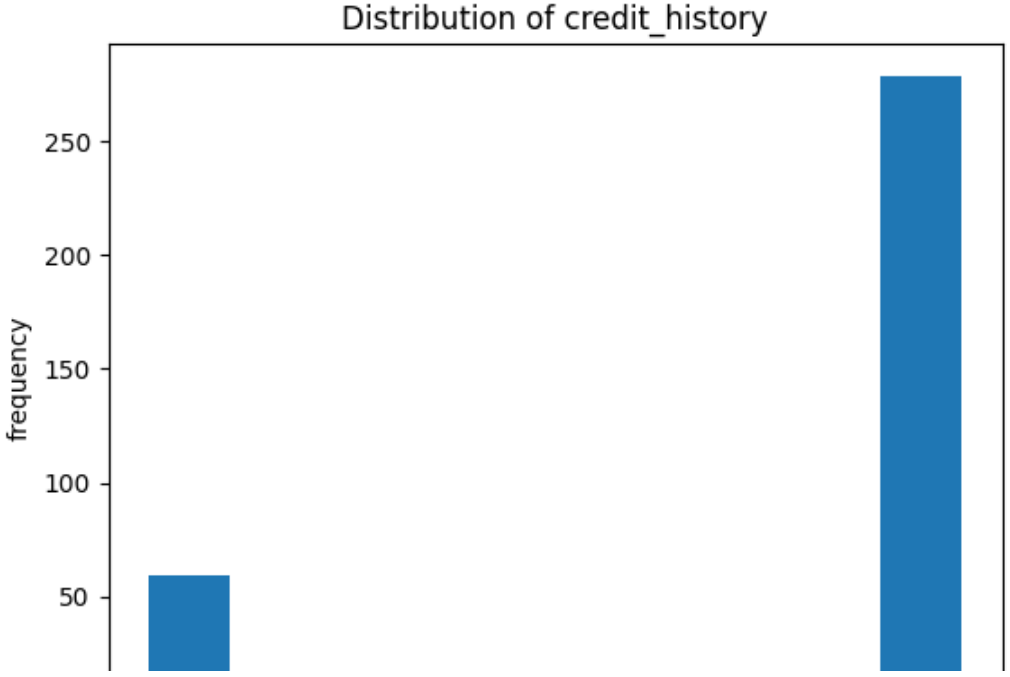


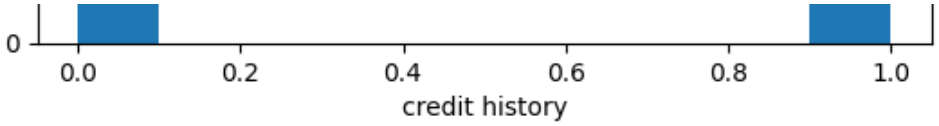

```
data['LoanAmount']=data['LoanAmount'].fillna(data['LoanAmount'].mean())
```

```
data.isna().sum()
```

```
Loan_ID               0
Gender                0
Married               0
Dependents            0
Education             0
Self_Employed         0
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            0
Loan_Amount_Term      6
Credit_History       29
Property_Area         0
dtype: int64
```

```
plt.hist(data['Credit_History'])
plt.title('Distribution of credit_history')
plt.xlabel('credit history')
plt.ylabel('frequency')
plt.show()
```



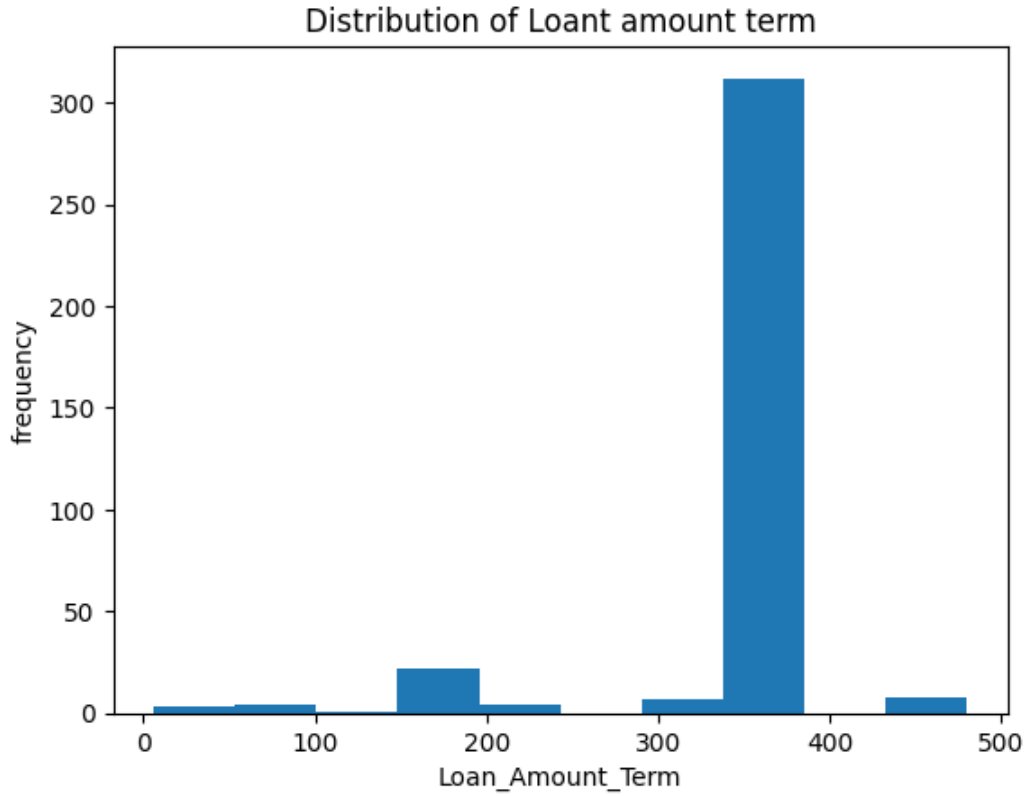



```
data['Credit_History']=data['Credit_History'].fillna(data['Credit_History'].median())
```

```
data.isna().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
dtype: int64
```

```
plt.hist(data['Loan_Amount_Term'])
plt.title('Distribution of Loant amount term')
plt.xlabel('Loan_Amount_Term')
plt.ylabel('frequency')
plt.show()
```



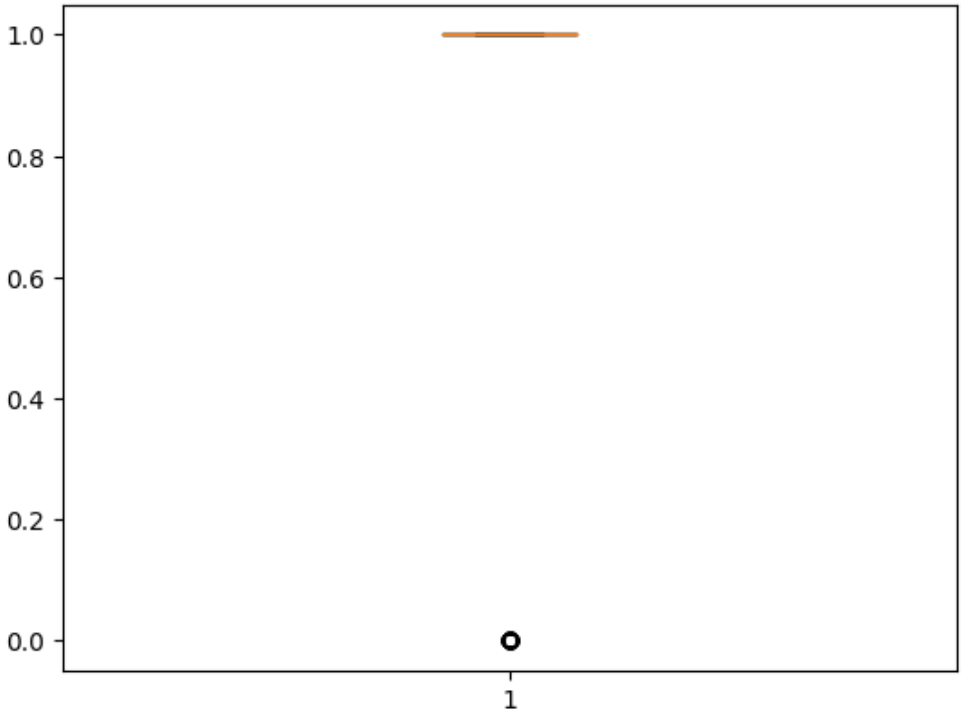

```
data['Loan_Amount_Term']=data['Loan_Amount_Term'].fillna(data['Loan_Amount_Term'].median(
```

```
data.isna().sum()
```

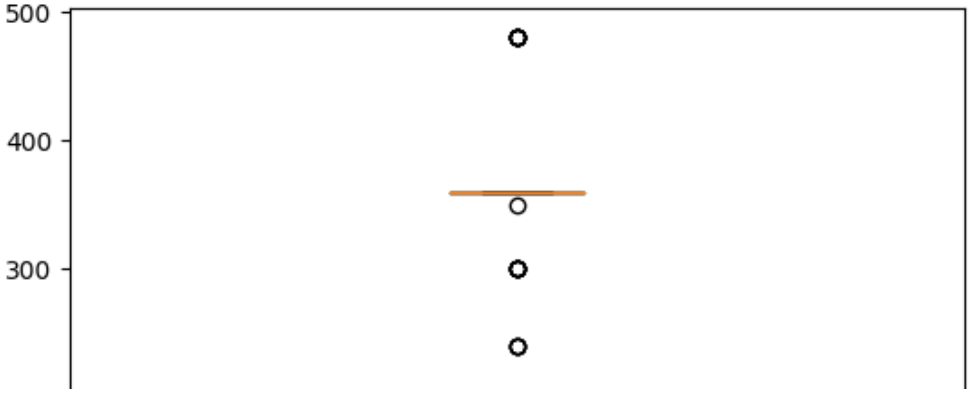
```
Loan_ID              0
Gender               0
Married              0
Dependents           0
Education            0
Self_Employed        0
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term     0
Credit_History       0
Property_Area        0
dtype: int64
```

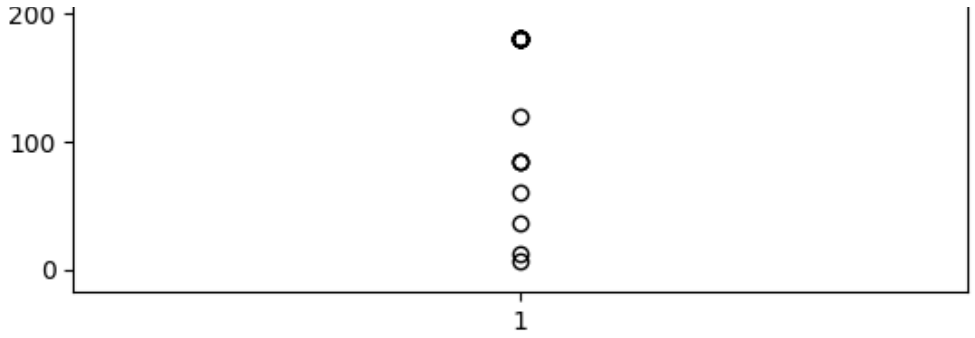
```
#b)outlier detection
```

```
plt.boxplot(data['Credit_History'])
plt.show()
```

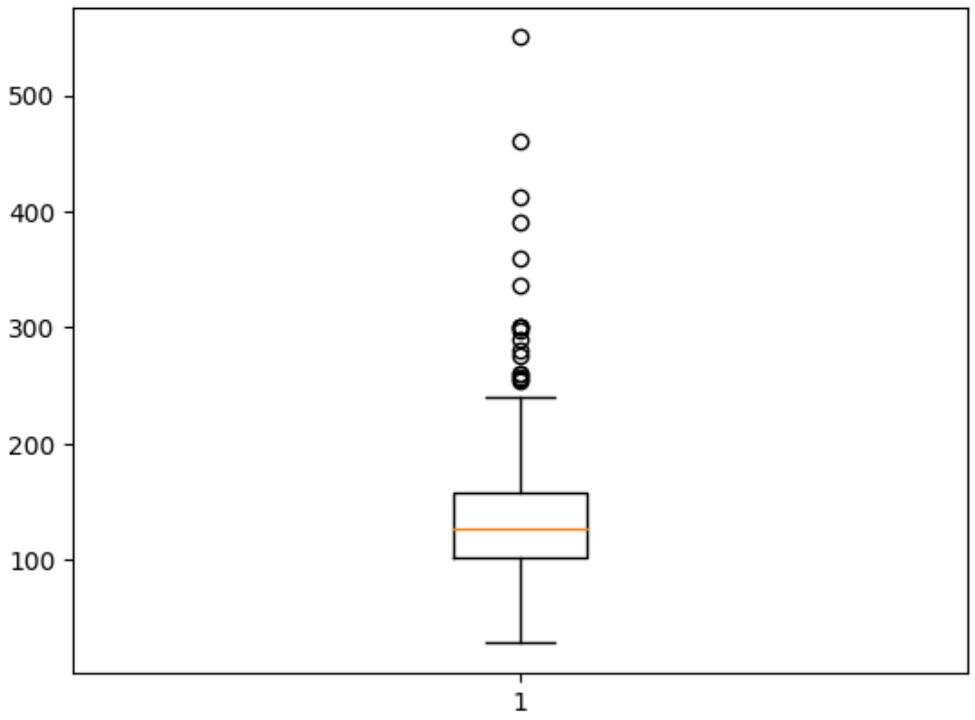



```
plt.boxplot(data['Loan_Amount_Term'])
plt.show()
```






```
plt.boxplot(data['LoanAmount'])
plt.show()
```




```
#Encoding
data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	C
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	

```
data.dtypes
```

```
Loan_ID              object
Gender               object
Married              object
Dependents           object
```

```
Dependents          object
Education           object
Self_Employed       object
ApplicantIncome      int64
CoapplicantIncome    int64
LoanAmount         float64
Loan_Amount_Term   float64
Credit_History     float64
Property_Area       object
dtype: object
```

```
data['Dependents'].unique()
```

```
array(['0', '1', '2', '3+'], dtype=object)
```

```
dict={'0':0,'1':1,'2':2,'3+':3}
```

```
data['Dependents']=data['Dependents'].map(dict)
```

```
data.dtypes
```

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

```
data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	C
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	

```
data['Self_Employed'].unique()
```

```
array(['No', 'Yes'], dtype=object)
```

```
#Label Encoding
```

```
from sklearn.preprocessing import LabelEncoder
```

```
le=LabelEncoder()
```

```
data['Self_Employed']=le.fit_transform(data['Self_Employed'])
```

```
data.dtypes
```

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

```
data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	C
0	LP001015	Male	Yes	0	Graduate	0	5720	
1	LP001022	Male	Yes	1	Graduate	0	3076	
2	LP001031	Male	Yes	2	Graduate	0	5000	
3	LP001035	Male	Yes	2	Graduate	0	2340	
4	LP001051	Male	No	0	Not Graduate	0	3276	

```
#one hot encoding
```

```
data=pd.get_dummies(data)
```

```
data.head()
```

	Dependents	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Am
0	0	0	5720	0	110.0	
1	1	0	3076	1500	126.0	
2	2	0	5000	1800	208.0	
3	2	0	2340	2546	100.0	
4	0	0	3276	0	78.0	

5 rows × 383 columns

```
#split data to independent and dependent column
```

```
data.dtypes
```

```
Dependents                   int64
Self_Employed                int64
ApplicantIncome              int64
CoapplicantIncome            int64
LoanAmount                 float64
                            ...
Education_Graduate           uint8
Education_Not Graduate       uint8
Property_Area_Rural          uint8
Property_Area_Semiurban      uint8
Property_Area_Urban          uint8
Length: 383, dtype: object
```

```
x=data.drop('Self_Employed',axis=1)
y=data['Self_Employed']
```

```
#minmax scalar
```

```
from sklearn.preprocessing import MinMaxScaler
min_max= MinMaxScaler()
```

```
x.loc[:,['LoanAmount','Dependents','ApplicantIncome','CoapplicantIncome']]=min_max.fit_tr
```

```
data.head()
```

	Dependents	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Am
0	0	0	5720	0	110.0	
1	1	0	3076	1500	126.0	
2	2	0	5000	1800	208.0	
3	2	0	2340	2546	100.0	
4	0	0	3276	0	78.0	

5 rows × 383 columns