

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sn
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

```
In [2]: df=pd.read_csv("loan_data_set.csv")
df
```

Out[2]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
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 [3]: df.isnull().sum()
```

Out[3]:

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	

```
In [4]: df["Gender"] = df["Gender"].map({"Male": 1, "Female": 0})
```

```
In [5]: df
```

Out[5]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
0	LP001002	1.0	No	0	Graduate	No	5849	
1	LP001003	1.0	Yes	1	Graduate	No	4583	
2	LP001005	1.0	Yes	0	Graduate	Yes	3000	
3	LP001006	1.0	Yes	0	Not Graduate	No	2583	
4	LP001008	1.0	No	0	Graduate	No	6000	
...	...	...	...	...	...	...	...	...
609	LP002978	0.0	No	0	Graduate	No	2900	
610	LP002979	1.0	Yes	3+	Graduate	No	4106	
611	LP002983	1.0	Yes	1	Graduate	No	8072	
612	LP002984	1.0	Yes	2	Graduate	No	7583	
613	LP002990	0.0	No	0	Graduate	Yes	4583	

614 rows × 13 columns

In [6]: `df["Gender"].value_counts()`

Out[6]:

```
Gender
1.0    489
0.0    112
Name: count, dtype: int64
```

In [7]: `df.Gender.fillna(1,inplace=True)`

In [8]: `df`

Out[8]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
0	LP001002	1.0	No	0	Graduate	No	5849	
1	LP001003	1.0	Yes	1	Graduate	No	4583	
2	LP001005	1.0	Yes	0	Graduate	Yes	3000	
3	LP001006	1.0	Yes	0	Not Graduate	No	2583	
4	LP001008	1.0	No	0	Graduate	No	6000	
...	...	...	...	...	...	...	...	...
609	LP002978	0.0	No	0	Graduate	No	2900	
610	LP002979	1.0	Yes	3+	Graduate	No	4106	
611	LP002983	1.0	Yes	1	Graduate	No	8072	
612	LP002984	1.0	Yes	2	Graduate	No	7583	
613	LP002990	0.0	No	0	Graduate	Yes	4583	

614 rows × 13 columns

In [9]: `df.isnull().sum()`

Out[9]:

Loan_ID	0
Gender	0
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

In [10]: `df["Dependents"] = df["Dependents"].map({"1":1, "0":0, "2":2, "3+":3})`

In [11]: `df`

Out[11]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
0	LP001002	1.0	No	0.0	Graduate	No	5849	
1	LP001003	1.0	Yes	1.0	Graduate	No	4583	
2	LP001005	1.0	Yes	0.0	Graduate	Yes	3000	
3	LP001006	1.0	Yes	0.0	Not Graduate	No	2583	
4	LP001008	1.0	No	0.0	Graduate	No	6000	
...	...	...	...	...	...	...	...	...
609	LP002978	0.0	No	0.0	Graduate	No	2900	
610	LP002979	1.0	Yes	3.0	Graduate	No	4106	
611	LP002983	1.0	Yes	1.0	Graduate	No	8072	
612	LP002984	1.0	Yes	2.0	Graduate	No	7583	
613	LP002990	0.0	No	0.0	Graduate	Yes	4583	

614 rows × 13 columns

In [12]: `df["Dependents"].value_counts()`

Out[12]:

Dependents	
0.0	345
1.0	102
2.0	101
3.0	51
Name: count, dtype:	int64

In [13]: `df.Dependents.fillna(0,inplace=True)`

In [14]: `df.isnull().sum()`

```
Out[14]: 
Loan_ID          0
Gender           0
Married          3
Dependents       0
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
```

```
In [15]: df["Self_Employed"] = df["Self_Employed"].map({"Yes":1, "No":0})
```

```
In [16]: df
```

```
Out[16]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
0	LP001002	1.0	No	0.0	Graduate	0.0	5849	
1	LP001003	1.0	Yes	1.0	Graduate	0.0	4583	
2	LP001005	1.0	Yes	0.0	Graduate	1.0	3000	
3	LP001006	1.0	Yes	0.0	Not Graduate	0.0	2583	
4	LP001008	1.0	No	0.0	Graduate	0.0	6000	
...	...	...	...	...	...	...	...	...
609	LP002978	0.0	No	0.0	Graduate	0.0	2900	
610	LP002979	1.0	Yes	3.0	Graduate	0.0	4106	
611	LP002983	1.0	Yes	1.0	Graduate	0.0	8072	
612	LP002984	1.0	Yes	2.0	Graduate	0.0	7583	
613	LP002990	0.0	No	0.0	Graduate	1.0	4583	

614 rows × 13 columns

```
In [17]: df["Loan_Status"] = df["Loan_Status"].map({"Y":1, "N":0})
```

```
In [18]: df
```

Out[18]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
<b>0</b>	LP001002	1.0	No	0.0	Graduate	0.0	5849	
<b>1</b>	LP001003	1.0	Yes	1.0	Graduate	0.0	4583	
<b>2</b>	LP001005	1.0	Yes	0.0	Graduate	1.0	3000	
<b>3</b>	LP001006	1.0	Yes	0.0	Not Graduate	0.0	2583	
<b>4</b>	LP001008	1.0	No	0.0	Graduate	0.0	6000	
...	...	...	...	...	...	...	...	...
<b>609</b>	LP002978	0.0	No	0.0	Graduate	0.0	2900	
<b>610</b>	LP002979	1.0	Yes	3.0	Graduate	0.0	4106	
<b>611</b>	LP002983	1.0	Yes	1.0	Graduate	0.0	8072	
<b>612</b>	LP002984	1.0	Yes	2.0	Graduate	0.0	7583	
<b>613</b>	LP002990	0.0	No	0.0	Graduate	1.0	4583	

614 rows × 13 columns

In [19]: `df["Self_Employed"].value_counts()`

Out[19]:

```
Self_Employed
0.0    500
1.0     82
Name: count, dtype: int64
```

In [20]: `df['Self_Employed'].fillna(0,inplace=True)`

In [21]: `df`

Out[21]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
<b>0</b>	LP001002	1.0	No	0.0	Graduate	0.0	5849	
<b>1</b>	LP001003	1.0	Yes	1.0	Graduate	0.0	4583	
<b>2</b>	LP001005	1.0	Yes	0.0	Graduate	1.0	3000	
<b>3</b>	LP001006	1.0	Yes	0.0	Not Graduate	0.0	2583	
<b>4</b>	LP001008	1.0	No	0.0	Graduate	0.0	6000	
...	...	...	...	...	...	...	...	...
<b>609</b>	LP002978	0.0	No	0.0	Graduate	0.0	2900	
<b>610</b>	LP002979	1.0	Yes	3.0	Graduate	0.0	4106	
<b>611</b>	LP002983	1.0	Yes	1.0	Graduate	0.0	8072	
<b>612</b>	LP002984	1.0	Yes	2.0	Graduate	0.0	7583	
<b>613</b>	LP002990	0.0	No	0.0	Graduate	1.0	4583	

614 rows × 13 columns

In [22]: `df.isnull().sum()`

Out[22]:

Loan_ID	0
Gender	0
Married	3
Dependents	0
Education	0
Self_Employed	0
ApplicantIncome	0
CoapplicantIncome	0
LoanAmount	22
Loan_Amount_Term	14
Credit_History	50
Property_Area	0
Loan_Status	0

dtype: int64

In [23]: `k=df.Credit_History.mean()  
k=round(k,1)  
k`

Out[23]: 0.8

In [24]: `df.Credit_History.fillna(k,inplace=True)`

In [25]: `df`

Out[25]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
0	LP001002	1.0	No	0.0	Graduate		0.0	5849
1	LP001003	1.0	Yes	1.0	Graduate		0.0	4583
2	LP001005	1.0	Yes	0.0	Graduate		1.0	3000
3	LP001006	1.0	Yes	0.0	Not Graduate		0.0	2583
4	LP001008	1.0	No	0.0	Graduate		0.0	6000
...	...	...	...	...	...	...	...	...
609	LP002978	0.0	No	0.0	Graduate		0.0	2900
610	LP002979	1.0	Yes	3.0	Graduate		0.0	4106
611	LP002983	1.0	Yes	1.0	Graduate		0.0	8072
612	LP002984	1.0	Yes	2.0	Graduate		0.0	7583
613	LP002990	0.0	No	0.0	Graduate		1.0	4583

614 rows × 13 columns

In [26]: `df.isnull().sum()`

```
Out[26]: 
Loan_ID          0
Gender           0
Married          3
Dependents       0
Education         0
Self_Employed    0
ApplicantIncome   0
CoapplicantIncome 0
LoanAmount        22
Loan_Amount_Term 14
Credit_History    0
Property_Area     0
Loan_Status        0
dtype: int64
```

```
In [27]: s=df.LoanAmount.mean()
s=round(s,1)
s
```

```
Out[27]: 146.4
```

```
In [28]: df.LoanAmount.fillna(s,inplace=True)
```

```
In [29]: df
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
0	LP001002	1.0	No	0.0	Graduate	0.0	5849	
1	LP001003	1.0	Yes	1.0	Graduate	0.0	4583	
2	LP001005	1.0	Yes	0.0	Graduate	1.0	3000	
3	LP001006	1.0	Yes	0.0	Not Graduate	0.0	2583	
4	LP001008	1.0	No	0.0	Graduate	0.0	6000	
...	...	...	...	...	...	...	...	...
609	LP002978	0.0	No	0.0	Graduate	0.0	2900	
610	LP002979	1.0	Yes	3.0	Graduate	0.0	4106	
611	LP002983	1.0	Yes	1.0	Graduate	0.0	8072	
612	LP002984	1.0	Yes	2.0	Graduate	0.0	7583	
613	LP002990	0.0	No	0.0	Graduate	1.0	4583	

614 rows × 13 columns

```
In [30]: df.isnull().sum()
```

```
Out[30]: 
Loan_ID          0
Gender           0
Married          3
Dependents       0
Education         0
Self_Employed    0
ApplicantIncome   0
CoapplicantIncome 0
LoanAmount        0
Loan_Amount_Term  14
Credit_History    0
Property_Area     0
Loan_Status        0
dtype: int64
```

```
In [31]: a=df.Loan_Amount_Term.mean()
a=round(a,1)
a
```

```
Out[31]: 342.0
```

```
In [32]: df.Loan_Amount_Term.fillna(a,inplace=True)
```

```
In [33]: df
```

```
Out[33]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
0	LP001002	1.0	No	0.0	Graduate	0.0	5849	
1	LP001003	1.0	Yes	1.0	Graduate	0.0	4583	
2	LP001005	1.0	Yes	0.0	Graduate	1.0	3000	
3	LP001006	1.0	Yes	0.0	Not Graduate	0.0	2583	
4	LP001008	1.0	No	0.0	Graduate	0.0	6000	
...	...	...	...	...	...	...	...	...
609	LP002978	0.0	No	0.0	Graduate	0.0	2900	
610	LP002979	1.0	Yes	3.0	Graduate	0.0	4106	
611	LP002983	1.0	Yes	1.0	Graduate	0.0	8072	
612	LP002984	1.0	Yes	2.0	Graduate	0.0	7583	
613	LP002990	0.0	No	0.0	Graduate	1.0	4583	

614 rows × 13 columns

```
In [34]: df.isnull().sum()
```

```
Out[34]: 
Loan_ID          0
Gender           0
Married          3
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 [35]: df["Married"] = df["Married"].map({"Yes":1, "No":0})
```

```
In [36]: df
```

```
Out[36]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
0	LP001002	1.0	0.0	0.0	Graduate	0.0	5849	
1	LP001003	1.0	1.0	1.0	Graduate	0.0	4583	
2	LP001005	1.0	1.0	0.0	Graduate	1.0	3000	
3	LP001006	1.0	1.0	0.0	Not Graduate	0.0	2583	
4	LP001008	1.0	0.0	0.0	Graduate	0.0	6000	
...	...	...	...	...	...	...	...	...
609	LP002978	0.0	0.0	0.0	Graduate	0.0	2900	
610	LP002979	1.0	1.0	3.0	Graduate	0.0	4106	
611	LP002983	1.0	1.0	1.0	Graduate	0.0	8072	
612	LP002984	1.0	1.0	2.0	Graduate	0.0	7583	
613	LP002990	0.0	0.0	0.0	Graduate	1.0	4583	

614 rows × 13 columns

```
In [37]: df["Education"] = df["Education"].map({"Graduate":1, "Not Graduate":0})
```

```
In [38]: df
```

Out[38]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
0	LP001002	1.0	0.0	0.0	1	0.0	5849	
1	LP001003	1.0	1.0	1.0	1	0.0	4583	
2	LP001005	1.0	1.0	0.0	1	1.0	3000	
3	LP001006	1.0	1.0	0.0	0	0.0	2583	
4	LP001008	1.0	0.0	0.0	1	0.0	6000	
...	...	...	...	...	...	...	...	...
609	LP002978	0.0	0.0	0.0	1	0.0	2900	
610	LP002979	1.0	1.0	3.0	1	0.0	4106	
611	LP002983	1.0	1.0	1.0	1	0.0	8072	
612	LP002984	1.0	1.0	2.0	1	0.0	7583	
613	LP002990	0.0	0.0	0.0	1	1.0	4583	

614 rows × 13 columns

In [39]: `df["Property_Area"] = df["Property_Area"].map({"Urban":1, "Rural":2, "Semiurban":0})`

In [40]: `df`

Out[40]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
0	LP001002	1.0	0.0	0.0	1	0.0	5849	
1	LP001003	1.0	1.0	1.0	1	0.0	4583	
2	LP001005	1.0	1.0	0.0	1	1.0	3000	
3	LP001006	1.0	1.0	0.0	0	0.0	2583	
4	LP001008	1.0	0.0	0.0	1	0.0	6000	
...	...	...	...	...	...	...	...	...
609	LP002978	0.0	0.0	0.0	1	0.0	2900	
610	LP002979	1.0	1.0	3.0	1	0.0	4106	
611	LP002983	1.0	1.0	1.0	1	0.0	8072	
612	LP002984	1.0	1.0	2.0	1	0.0	7583	
613	LP002990	0.0	0.0	0.0	1	1.0	4583	

614 rows × 13 columns

In [41]: `df.isnull().sum()`

```
Out[41]: Loan_ID      0
Gender        0
Married       3
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 [42]: df["Married"].value_counts()
```

```
Out[42]: Married
1.0    398
0.0    213
Name: count, dtype: int64
```

```
In [43]: df.Married.fillna(1,inplace=True)
```

```
In [44]: df
```

```
Out[44]:   Loan_ID  Gender  Married  Dependents  Education  Self_Employed  ApplicantIncome  Coapp
0 LP001002  1.0      0.0      0.0          1           0.0            5849
1 LP001003  1.0      1.0      1.0          1           0.0            4583
2 LP001005  1.0      1.0      0.0          1           1.0            3000
3 LP001006  1.0      1.0      0.0          0           0.0            2583
4 LP001008  1.0      0.0      0.0          1           0.0            6000
...
609 LP002978 0.0      0.0      0.0          1           0.0            2900
610 LP002979 1.0      1.0      3.0          1           0.0            4106
611 LP002983 1.0      1.0      1.0          1           0.0            8072
612 LP002984 1.0      1.0      2.0          1           0.0            7583
613 LP002990 0.0      0.0      0.0          1           1.0            4583
```

614 rows × 13 columns

```
In [45]: df.isnull().sum()
```

```
Out[45]: 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 [46]: df.drop("Loan_ID", axis=1, inplace=True)  
df
```

```
Out[46]:   Gender  Married  Dependents  Education  Self_Employed  ApplicantIncome  CoapplicantIncome  
0      1.0      0.0        0.0         1          0.0            5849                  0  
1      1.0      1.0        1.0         1          0.0            4583             1508  
2      1.0      1.0        0.0         1          1.0            3000                  0  
3      1.0      1.0        0.0         0          0.0            2583             2358  
4      1.0      0.0        0.0         1          0.0            6000                  0  
...    ...      ...        ...        ...          ...          ...            ...  
609    0.0      0.0        0.0         1          0.0            2900                  0  
610    1.0      1.0        3.0         1          0.0            4106                  0  
611    1.0      1.0        1.0         1          0.0            8072             240  
612    1.0      1.0        2.0         1          0.0            7583                  0  
613    0.0      0.0        0.0         1          1.0            4583                  0
```

614 rows × 12 columns

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

Out[47]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome
0	1.0	0.0	0.0	1	0.0	5849	0
1	1.0	1.0	1.0	1	0.0	4583	1508
2	1.0	1.0	0.0	1	1.0	3000	0
3	1.0	1.0	0.0	0	0.0	2583	2358
4	1.0	0.0	0.0	1	0.0	6000	0
...	...	...	...	...	...	...	...
609	0.0	0.0	0.0	1	0.0	2900	0
610	1.0	1.0	3.0	1	0.0	4106	0
611	1.0	1.0	1.0	1	0.0	8072	240
612	1.0	1.0	2.0	1	0.0	7583	0
613	0.0	0.0	0.0	1	1.0	4583	0

614 rows × 11 columns

In [48]: `y=df.iloc[:, -1]`  
`y`

Out[48]:

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

Name: Loan\_Status, Length: 614, dtype: int64

In [49]: `from sklearn.model_selection import train_test_split`  
`x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=51)`

In [50]: `print(x_train.shape)`  
`print(x_test.shape)`  
`print(y_train.shape)`  
`print(y_test.shape)`

(429, 11)  
(185, 11)  
(429,)  
(185,)

In [51]: `from sklearn.tree import DecisionTreeClassifier`  
`model=DecisionTreeClassifier(criterion="entropy")`  
`model.fit(x_train,y_train)`

Out[51]:

▼      **DecisionTreeClassifier**

```
DecisionTreeClassifier(criterion='entropy')
```

```
In [52]: model.score(x_test,y_test)
```

```
Out[52]: 0.7081081081081081
```

```
In [53]: from sklearn.tree import DecisionTreeClassifier
model=DecisionTreeClassifier(criterion="gini")
model.fit(x_train,y_train)
```

```
Out[53]: ▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
In [54]: model.score(x_test,y_test)
```

```
Out[54]: 0.7243243243243244
```

```
In [55]: model.predict(x_test)
```

```
Out[55]: array([0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1,
0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1,
1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,
1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,
1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,
1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0,
1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1,
1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1,
1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1,
```

```
In [56]: x_test.iloc[0:1,:]
```

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncom
<b>377</b>	1.0	1.0	0.0	1	0.0	4310	0

```
In [57]: k=model.predict(x_test.iloc[0:1,:])
```

```
In [58]: if k==[1]:
    print("loan sanctioned")
else:
    print("loan rejected")
```

```
loan rejected
```

```
In [ ]:
```