```python
import pandas as pd
df=pd.read_csv("loan_data.csv")
df.head(30)
```

```
     Loan_ID  Gender Married Dependents  ... Loan_Amount_Term 
Credit_History  Property_Area  Loan_Status
0   LP001002    Male      No          0  ...            360.0 
1.0         Urban           Y
1   LP001003    Male     Yes          1  ...            360.0 
1.0         Rural           N
2   LP001005    Male     Yes          0  ...            360.0 
1.0         Urban           Y
3   LP001006    Male     Yes          0  ...            360.0 
1.0         Urban           Y
4   LP001008    Male      No          0  ...            360.0 
1.0         Urban           Y
5   LP001011    Male     Yes          2  ...            360.0 
1.0         Urban           Y
6   LP001013    Male     Yes          0  ...            360.0 
1.0         Urban           Y
7   LP001014    Male     Yes         3+  ...            360.0 
0.0     Semiurban           N
8   LP001018    Male     Yes          2  ...            360.0 
1.0         Urban           Y
9   LP001020    Male     Yes          1  ...            360.0 
1.0     Semiurban           N
10  LP001024    Male     Yes          2  ...            360.0 
1.0         Urban           Y
11  LP001027    Male     Yes          2  ...            360.0 
1.0         Urban           Y
12  LP001028    Male     Yes          2  ...            360.0 
1.0         Urban           Y
13  LP001029    Male      No          0  ...            360.0 
1.0         Rural           N
14  LP001030    Male     Yes          2  ...            120.0 
1.0         Urban           Y
15  LP001032    Male      No          0  ...            360.0 
1.0         Urban           Y
16  LP001034    Male      No          1  ...            240.0 
NaN         Urban           Y
17  LP001036  Female      No          0  ...            360.0 
0.0         Urban           N
18  LP001038    Male     Yes          0  ...            360.0 
1.0         Rural           N
19  LP001041    Male     Yes          0  ...              NaN 
1.0         Urban           Y
20  LP001043    Male     Yes          0  ...            360.0 
0.0         Urban           N
21  LP001046    Male     Yes          1  ...            360.0 
1.0         Urban           Y
```

```
22  LP001047    Male     Yes            0  ...            360.0
0.0      Semiurban            N
23  LP001050     NaN     Yes            2  ...            360.0
0.0          Rural            N
24  LP001052    Male     Yes            1  ...            360.0
NaN      Semiurban            N
25  LP001066    Male     Yes            0  ...            360.0
1.0      Semiurban            Y
26  LP001068    Male     Yes            0  ...            360.0
1.0      Semiurban            Y
27  LP001073    Male     Yes            2  ...            360.0
1.0          Urban            Y
28  LP001086    Male      No            0  ...            360.0
1.0          Urban            N
29  LP001087  Female      No            2  ...            360.0
1.0      Semiurban            Y

[30 rows x 13 columns]
```

```
df.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
```

```
df.isna().sum()
```

```
Loan_ID               0
Gender               13
Married               3
Dependents           15
Education             0
Self_Employed        32
```

```
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           22
Loan_Amount_Term     14
Credit_History       50
Property_Area         0
Loan_Status           0
dtype: int64
```

```python
df.dropna(inplace=True)
```

```python
df.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
Loan_Status          0
dtype: int64
```

```python
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
d=["Loan_ID","Gender","Married","Dependents","Education","Self_Employed","Loan_Status","Property_Area"]
for f in d:
    df[f]=le.fit_transform(df[f])
```

```python
df.corr()
```

```
                    Loan_ID    Gender   Married  ...  Credit_History
Property_Area  Loan_Status
Loan_ID            1.000000 -0.023210  0.005776  ...       -0.018872
-0.197603     0.040306
Gender            -0.023210  1.000000  0.349424  ...        0.022447
-0.000204     0.064504
Married            0.005776  0.349424  1.000000  ...        0.029095
0.038653     0.112321
```

```
Dependents          0.077974  0.217510  0.386367  ...        -0.026651
0.001191     0.035428
Education           0.028438  0.059245  0.001652  ...        -0.056656
-0.055005    -0.068437
Self_Employed       0.049772 -0.002761  0.015674  ...        -0.023568
-0.050797    -0.034715
ApplicantIncome     0.038843  0.032644  0.036717  ...        -0.056152
-0.053160    -0.043152
CoapplicantIncome -0.011608  0.156171  0.102950  ...        -0.008692
0.006540    -0.049020
LoanAmount          0.049712  0.098975  0.183442  ...        -0.040773
-0.109685    -0.071753
Loan_Amount_Term  -0.004265 -0.088704 -0.107504  ...         0.032937
-0.058656    -0.007798
Credit_History    -0.018872  0.022447  0.029095  ...         1.000000
-0.003013     0.529390
Property_Area     -0.197603 -0.000204  0.038653  ...        -0.003013
1.000000     0.031361
Loan_Status         0.040306  0.064504  0.112321  ...         0.529390
0.031361     1.000000

[13 rows x 13 columns]
```

```
x=df[["Credit_History"]]
y=df.Loan_Status
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

```
from sklearn.linear_model import LinearRegression
le=LinearRegression()
le.fit(x_train,y_train)
```

```
LinearRegression()
```

```
ypr=le.predict(x_test)
```

```
ypr
```

```
array([0.79447853, 0.79447853, 0.0862069 , 0.79447853, 0.79447853,
       0.79447853, 0.79447853, 0.79447853, 0.0862069 , 0.79447853,
       0.79447853, 0.79447853, 0.79447853, 0.79447853, 0.79447853,
       0.79447853, 0.79447853, 0.79447853, 0.79447853, 0.79447853,
       0.0862069 , 0.79447853, 0.79447853, 0.79447853, 0.79447853,
       0.0862069 , 0.79447853, 0.0862069 , 0.79447853, 0.79447853,
       0.79447853, 0.79447853, 0.79447853, 0.79447853, 0.0862069 ,
       0.79447853, 0.0862069 , 0.79447853, 0.79447853, 0.79447853,
       0.79447853, 0.79447853, 0.79447853, 0.79447853, 0.79447853,
       0.79447853, 0.79447853, 0.79447853, 0.79447853, 0.79447853,
       0.79447853, 0.79447853, 0.79447853, 0.79447853, 0.0862069 ,
       0.79447853, 0.79447853, 0.79447853, 0.79447853, 0.79447853,
       0.79447853, 0.79447853, 0.79447853, 0.0862069 , 0.0862069 ,
       0.79447853, 0.79447853, 0.79447853, 0.79447853, 0.79447853,
       0.79447853, 0.79447853, 0.79447853, 0.79447853, 0.79447853,
       0.79447853, 0.79447853, 0.79447853, 0.79447853, 0.79447853,
       0.79447853, 0.79447853, 0.79447853, 0.79447853, 0.79447853,
       0.79447853, 0.0862069 , 0.79447853, 0.79447853, 0.79447853,
       0.79447853, 0.79447853, 0.0862069 , 0.79447853, 0.79447853,
       0.79447853])
```

```python
from sklearn.metrics import r2_score
sk=r2_score(y_test,ypr)
```

```python
sk
```

```
0.1986389256759058
```

multi lin

```python
x=df[["Credit_History","Married","Gender","Property_Area"]]
y=df.Loan_Status
```

```python
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

```python
from sklearn.linear_model import LinearRegression
le=LinearRegression()
le.fit(x_train,y_train)
```

```
LinearRegression()
```

```python
ypr=le.predict(x_test)
```

```python
from sklearn.metrics import r2_score
sk=r2_score(y_test,ypr)
sk
```

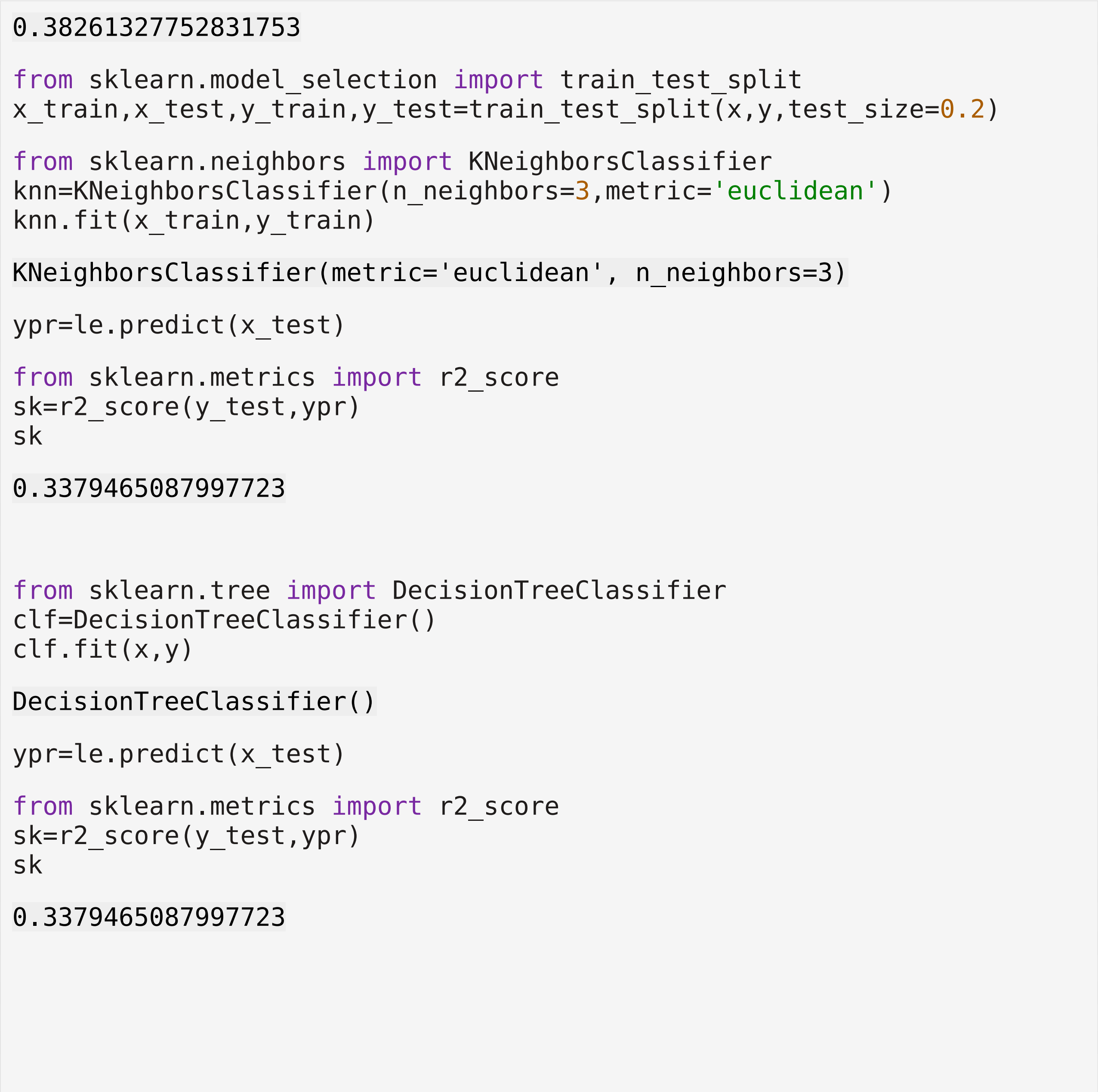
```
0.38261327752831753
```

```python
from sklearn.svm import LinearSVC #support vector classification
import warnings
warnings.simplefilter("ignore")
classifier=LinearSVC()
classifier.fit(x_train,y_train)
```

```
LinearSVC()
```

```python
ypr=le.predict(x_test)
```

```python
from sklearn.metrics import r2_score
sk=r2_score(y_test,ypr)
sk
```



```
0.38261327752831753
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=3,metric='euclidean')
knn.fit(x_train,y_train)
```

```
KNeighborsClassifier(metric='euclidean', n_neighbors=3)
```

```
ypr=le.predict(x_test)
```

```
from sklearn.metrics import r2_score
sk=r2_score(y_test,ypr)
sk
```

```
0.3379465087997723
```

```
from sklearn.tree import DecisionTreeClassifier
clf=DecisionTreeClassifier()
clf.fit(x,y)
```

```
DecisionTreeClassifier()
```

```
ypr=le.predict(x_test)
```

```
from sklearn.metrics import r2_score
sk=r2_score(y_test,ypr)
sk
```

```
0.3379465087997723
```