```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score

#
(a)-----------------------------------------------------------------------
---------
path = 'loan_old.csv'
data_old = pd.read_csv(path)
print('data_old')
print(data_old)
```

```
data_old
      Loan_ID  Gender Married Dependents     Education  Income  \
0    LP001002    Male      No          0      Graduate    5849
1    LP001003    Male     Yes          1      Graduate    4583
2    LP001005    Male     Yes          0      Graduate    3000
3    LP001006    Male     Yes          0  Not Graduate    2583
4    LP001008    Male      No          0      Graduate    6000
..        ...     ...     ...        ...           ...     ...
609  LP002978  Female      No          0      Graduate    2900
610  LP002979    Male     Yes         3+      Graduate    4106
611  LP002983    Male     Yes          1      Graduate    8072
612  LP002984    Male     Yes          2      Graduate    7583
613  LP002990  Female      No          0      Graduate    4583

     Coapplicant_Income  Loan_Tenor  Credit_History Property_Area  \
0                   0.0       144.0             1.0         Urban
1                1508.0       144.0             1.0         Rural
2                   0.0       144.0             1.0         Urban
3                2358.0       144.0             1.0         Urban
4                   0.0       144.0             1.0         Urban
..                  ...         ...             ...           ...
609                 0.0       144.0             1.0         Rural
610                 0.0        72.0             1.0         Rural
611               240.0       144.0             1.0         Urban
612                 0.0       144.0             1.0         Urban
613                 0.0       144.0             0.0     Semiurban

     Max_Loan_Amount Loan_Status
0                NaN           Y
1             236.99           N
2              81.20           Y
3             179.03           Y
4             232.40           Y
```

```
..                ...         ...
609             76.16           Y
610             33.47           Y
611            348.92           Y
612            312.18           Y
613            160.98           N

[614 rows x 12 columns]

path1 = 'loan_new.csv'
data_new = pd.read_csv(path1)
print('data_new')
print(data_new)

data_new
      Loan_ID Gender Married Dependents     Education  Income  \
0    LP001015   Male     Yes          0      Graduate    5720   
1    LP001022   Male     Yes          1      Graduate    3076   
2    LP001031   Male     Yes          2      Graduate    5000   
3    LP001035   Male     Yes          2      Graduate    2340   
4    LP001051   Male      No          0  Not Graduate    3276   
..        ...    ...     ...        ...           ...     ...   
362  LP002971   Male     Yes         3+  Not Graduate    4009   
363  LP002975   Male     Yes          0      Graduate    4158   
364  LP002980   Male      No          0      Graduate    3250   
365  LP002986   Male     Yes          0      Graduate    5000   
366  LP002989   Male      No          0      Graduate    9200   

     Coapplicant_Income  Loan_Tenor  Credit_History Property_Area
0                     0       144.0             1.0         Urban
1                  1500       144.0             1.0         Urban
2                  1800       144.0             1.0         Urban
3                  2546       144.0             NaN         Urban
4                     0       144.0             1.0         Urban
..                  ...         ...             ...           ...
362                1777       144.0             1.0         Urban
363                 709       144.0             1.0         Urban
364                1993       144.0             NaN     Semiurban
365                2393       144.0             1.0         Rural
366                   0        72.0             1.0         Rural

[367 rows x 10 columns]

# (b)
(i)-------------------------------------------------------------------
---------
#  check whether there are missing values for loan_old
missing_values_old = data_old.isnull().sum()
print("Missing values in each column in loan_old:")
print(missing_values_old)
```

```
Missing values in each column in loan_old:
Loan_ID                 0
Gender                 13
Married                 3
Dependents             15
Education               0
Income                  0
Coapplicant_Income      0
Loan_Tenor             15
Credit_History         50
Property_Area           0
Max_Loan_Amount        25
Loan_Status             0
dtype: int64
```

```
# (b)
(ii)----------------------------------------------------------------------
----------
def check_type(column):
    if column.dtype == 'object':
        return 'categorical'
    elif pd.api.types.is_numeric_dtype(column):
        return 'numerical'

# Apply check_type to each column in loan_old
types_old = data_old.apply(check_type)
categorical_columns_old = types_old[types_old ==
'categorical'].index.tolist()
numerical_columns_old = types_old[types_old ==
'numerical'].index.tolist()

print("Categorical columns for loan_old:", categorical_columns_old)
print("Numerical columns for loan_old:", numerical_columns_old)

# Apply check_type to each column in loan_new
types_new = data_new.apply(check_type)
categorical_columns_new = types_new[types_new ==
'categorical'].index.tolist()
numerical_columns_new = types_new[types_new ==
'numerical'].index.tolist()
```

```
Categorical columns for loan_old: ['Loan_ID', 'Gender', 'Married',
'Dependents', 'Education', 'Property_Area', 'Loan_Status']
Numerical columns for loan_old: ['Income', 'Coapplicant_Income',
'Loan_Tenor', 'Credit_History', 'Max_Loan_Amount']
```

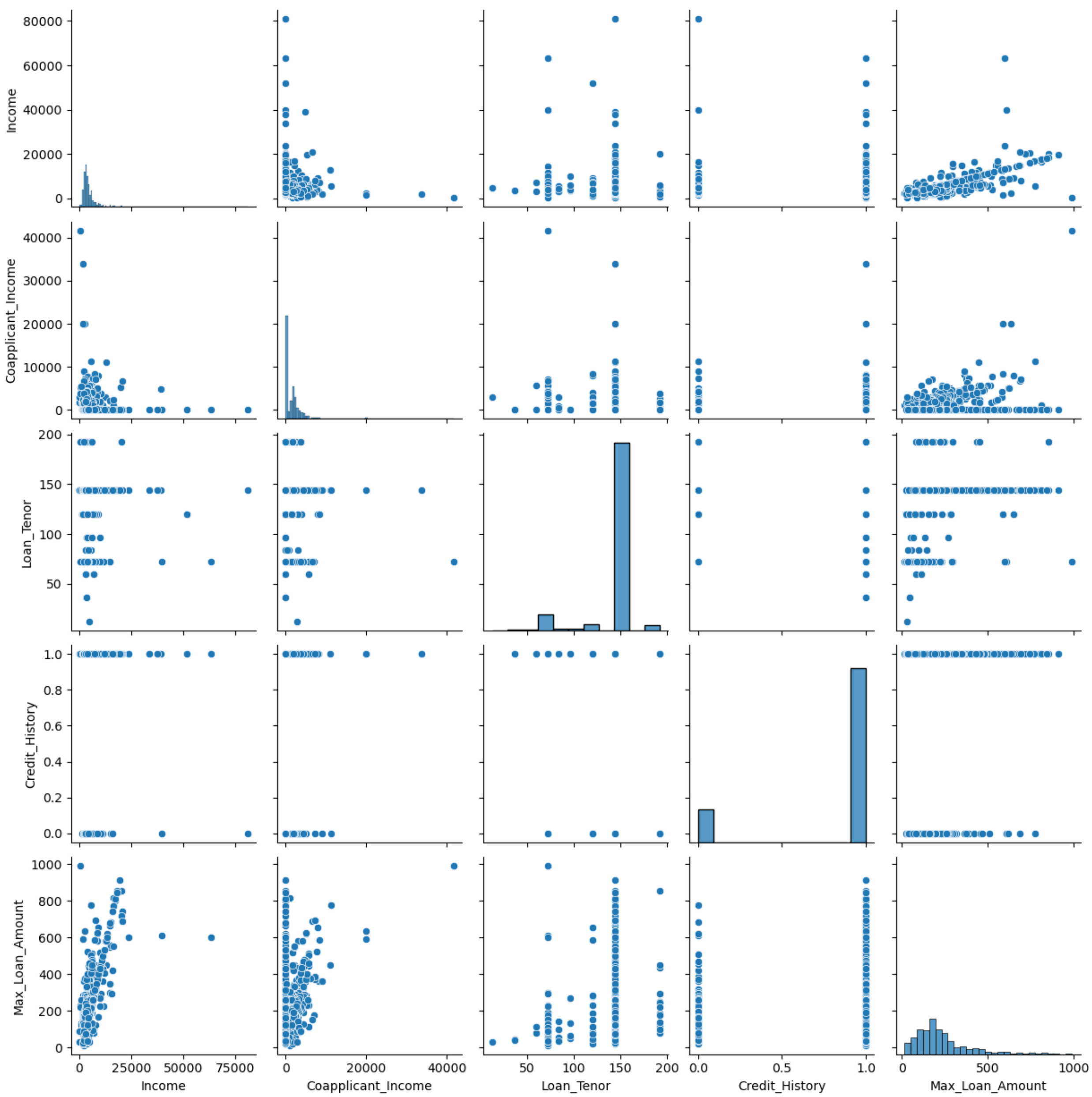
```
# (b)
(iii)---------------------------------------------------------------------
-----------
# check whether numerical features have the same scale
```

```python
for i in range(len(numerical_columns_old)-1):
    column_summary = data_old[numerical_columns_old[i]].describe()
    print("Summary statistics of", numerical_columns_old[i])
    print(column_summary)
    print()
```

```
Summary statistics of Income
count      614.000000
mean      5403.459283
std       6109.041673
min        150.000000
25%       2877.500000
50%       3812.500000
75%       5795.000000
max      81000.000000
Name: Income, dtype: float64

Summary statistics of Coapplicant_Income
count      614.000000
mean      1621.245798
std       2926.248369
min          0.000000
25%          0.000000
50%       1188.500000
75%       2297.250000
max      41667.000000
Name: Coapplicant_Income, dtype: float64

Summary statistics of Loan_Tenor
count    599.000000
mean     137.689482
std       23.366294
min       12.000000
25%      144.000000
50%      144.000000
75%      144.000000
max      192.000000
Name: Loan_Tenor, dtype: float64

Summary statistics of Credit_History
count    564.000000
mean       0.842199
std        0.364878
min        0.000000
25%        1.000000
50%        1.000000
75%        1.000000
max        1.000000
Name: Credit_History, dtype: float64
```

```
# (b)
(iv)----------------------------------------------------------------------
----------
# Visualize a pairplot between numercial columns for loan_old
pairplot_numerical_old = data_old[numerical_columns_old]
sns.pairplot(pairplot_numerical_old)
plt.show()
```

```
/Users/sarah/anaconda3/lib/python3.11/site-packages/seaborn/
axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



```
# (c)
(i)----------------------------------------------------------------------
---------
# Records containing missing values are removed
data_old = data_old.dropna()
print('data_old after removing missing values')
print(data_old)
```

```
data_old after removing missing values
      Loan_ID  Gender Married Dependents     Education  Income  \
1    LP001003    Male     Yes          1      Graduate    4583
2    LP001005    Male     Yes          0      Graduate    3000
3    LP001006    Male     Yes          0  Not Graduate    2583
4    LP001008    Male      No          0      Graduate    6000
5    LP001011    Male     Yes          2      Graduate    5417
..        ...     ...     ...        ...           ...     ...
609  LP002978  Female      No          0      Graduate    2900
610  LP002979    Male     Yes         3+      Graduate    4106
611  LP002983    Male     Yes          1      Graduate    8072
612  LP002984    Male     Yes          2      Graduate    7583
613  LP002990  Female      No          0      Graduate    4583

     Coapplicant_Income  Loan_Tenor  Credit_History Property_Area  \
1                1508.0       144.0             1.0         Rural
2                   0.0       144.0             1.0         Urban
3                2358.0       144.0             1.0         Urban
4                   0.0       144.0             1.0         Urban
5                4196.0       144.0             1.0         Urban
..                  ...         ...             ...           ...
609                 0.0       144.0             1.0         Rural
610                 0.0        72.0             1.0         Rural
611               240.0       144.0             1.0         Urban
612                 0.0       144.0             1.0         Urban
613                 0.0       144.0             0.0     Semiurban

     Max_Loan_Amount Loan_Status
1             236.99           N
2              81.20           Y
3             179.03           Y
4             232.40           Y
5             414.50           Y
..               ...         ...
609            76.16           Y
610            33.47           Y
611           348.92           Y
612           312.18           Y
613           160.98           N

[513 rows x 12 columns]
```

```
data_new = data_new.dropna()
print('data_new after removing missing values')
print(data_new)
data_new_copy = data_new.copy()
```

```
data_new after removing missing values
      Loan_ID Gender Married Dependents     Education  Income  \
0    LP001015   Male     Yes          0      Graduate    5720
1    LP001022   Male     Yes          1      Graduate    3076
2    LP001031   Male     Yes          2      Graduate    5000
4    LP001051   Male      No          0  Not Graduate    3276
5    LP001054   Male     Yes          0  Not Graduate    2165
..        ...    ...     ...        ...           ...     ...
361  LP002969   Male     Yes          1      Graduate    2269
362  LP002971   Male     Yes         3+  Not Graduate    4009
363  LP002975   Male     Yes          0      Graduate    4158
365  LP002986   Male     Yes          0      Graduate    5000
366  LP002989   Male      No          0      Graduate    9200

     Coapplicant_Income  Loan_Tenor  Credit_History Property_Area
0                     0       144.0             1.0         Urban
1                  1500       144.0             1.0         Urban
2                  1800       144.0             1.0         Urban
4                     0       144.0             1.0         Urban
5                  3422       144.0             1.0         Urban
..                  ...         ...             ...           ...
361                2167       144.0             1.0     Semiurban
362                1777       144.0             1.0         Urban
363                 709       144.0             1.0         Urban
365                2393       144.0             1.0         Rural
366                   0        72.0             1.0         Rural

[314 rows x 10 columns]
```

```
# (c)
(ii)----------------------------------------------------------------------
----------
# the features and targets are separated
num_of_cols = data_old.shape[1]
X_old = data_old.iloc[:, 1:num_of_cols - 2]
y_old = data_old.iloc[:, num_of_cols - 2:num_of_cols]
print("features")
print(X_old)
print("---------------------------------------------")
print("targets")
print(y_old)
```

```
features
     Gender Married Dependents     Education  Income
Coapplicant_Income  \
```

```
1      Male     Yes          1      Graduate    4583
1508.0
2      Male     Yes          0      Graduate    3000
0.0
3      Male     Yes          0  Not Graduate    2583
2358.0
4      Male      No          0      Graduate    6000
0.0
5      Male     Yes          2      Graduate    5417
4196.0
..      ...     ...        ...           ...     ...                 .
..
609  Female      No          0      Graduate    2900
0.0
610    Male     Yes         3+      Graduate    4106
0.0
611    Male     Yes          1      Graduate    8072
240.0
612    Male     Yes          2      Graduate    7583
0.0
613  Female      No          0      Graduate    4583
0.0

     Loan_Tenor  Credit_History Property_Area
1         144.0             1.0         Rural
2         144.0             1.0         Urban
3         144.0             1.0         Urban
4         144.0             1.0         Urban
5         144.0             1.0         Urban
..          ...             ...           ...
609       144.0             1.0         Rural
610        72.0             1.0         Rural
611       144.0             1.0         Urban
612       144.0             1.0         Urban
613       144.0             0.0     Semiurban

[513 rows x 9 columns]
---------------------------------------------
targets
     Max_Loan_Amount Loan_Status
1             236.99           N
2              81.20           Y
3             179.03           Y
4             232.40           Y
5             414.50           Y
..               ...         ...
609            76.16           Y
610            33.47           Y
611           348.92           Y
```

```
612           312.18          Y
613           160.98          N

[513 rows x 2 columns]

# (c)
(iii)-----------------------------------------------------------------------
-----------
# the data is shuffled and split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_old, y_old,
shuffle=True, test_size=0.25, random_state=45)
print("X_train")
print(X_train)
print("----------------------------------------------")
print("X_test")
print(X_test)
print("----------------------------------------------")
print("y_train")
print(y_train)
print("----------------------------------------------")
print("y_test")
print(y_test)

X_train
     Gender Married Dependents     Education  Income
Coapplicant_Income  \
413    Male     Yes          0  Not Graduate    2253
2033.0
339  Female      No          0      Graduate    4160
0.0
445    Male     Yes          1      Graduate    3466
1210.0
258    Male     Yes          0      Graduate   14683
2100.0
566    Male      No          0      Graduate    3333
0.0
..      ...     ...        ...           ...     ...                 .
..
115    Male     Yes          1      Graduate   14583
0.0
40     Male      No          0      Graduate    3600
0.0
457    Male     Yes          0      Graduate    3708
2569.0
158    Male      No          0      Graduate    2980
2083.0
498    Male     Yes          1      Graduate    2895
0.0

     Loan_Tenor  Credit_History Property_Area
```

```
413       144.0            1.0         Rural
339       144.0            1.0     Semiurban
445       144.0            1.0         Rural
258       144.0            1.0         Rural
566       144.0            1.0         Urban
..          ...            ...           ...
115        72.0            1.0         Rural
40        144.0            1.0         Urban
457       144.0            1.0         Urban
158       144.0            1.0         Rural
498       144.0            1.0     Semiurban

[384 rows x 9 columns]
---------------------------------------------
X_test
    Gender Married Dependents     Education  Income
Coapplicant_Income  \
385   Male      No          1      Graduate    3667
0.0
154   Male      No          0      Graduate    3254
0.0
447   Male     Yes          0      Graduate    3539
1376.0
543   Male     Yes          1  Not Graduate    2239
2524.0
525   Male     Yes          2      Graduate   17500
0.0
..     ...     ...        ...           ...     ...                  ..
.
485   Male     Yes          1  Not Graduate    1958
2436.0
55    Male     Yes          2      Graduate    2708
1167.0
584   Male     Yes          1      Graduate    2787
1917.0
361   Male     Yes          2      Graduate    5000
3667.0
528   Male      No          1  Not Graduate    2679
1302.0

     Loan_Tenor  Credit_History Property_Area
385        72.0            1.0         Urban
154       144.0            1.0         Urban
447       144.0            1.0         Rural
543       144.0            1.0         Urban
525       144.0            1.0         Rural
..          ...            ...           ...
485       144.0            1.0         Rural
55        144.0            1.0     Semiurban
```

```
584       144.0             0.0        Rural
361       144.0             1.0    Semiurban
528       144.0             1.0    Semiurban

[129 rows x 9 columns]
---------------------------------------------
y_train
     Max_Loan_Amount Loan_Status
413           146.01           Y
339           139.66           Y
445           165.67           Y
258           545.86           N
566            97.98           Y
..               ...         ...
115           297.49           Y
40            111.44           N
457           246.36           N
158           185.18           Y
498            75.91           Y

[384 rows x 2 columns]
---------------------------------------------
y_test
     Max_Loan_Amount Loan_Status
385            22.41           Y
154            94.00           Y
447           177.72           N
543           170.06           Y
525           812.00           Y
..               ...         ...
485           151.46           Y
55            125.30           Y
584           167.08           N
361           366.82           Y
528           130.64           Y

[129 rows x 2 columns]

# (c)
(iv)----------------------------------------------------------------------
----------
# categorical features are encoded
categorical_columns_X = categorical_columns_old[:-1]
label_encoder_X = LabelEncoder()

for i in range(1, len(categorical_columns_X)):
    X_train[categorical_columns_X[i]] =
label_encoder_X.fit_transform(X_train[categorical_columns_X[i]])
    X_test[categorical_columns_X[i]] =
label_encoder_X.transform(X_test[categorical_columns_X[i]])
```

```
    data_new[categorical_columns_X[i]] =
label_encoder_X.transform(data_new[categorical_columns_X[i]])

print("X_train after encoding")
print(X_train)
print("---------------------------------------------")
print("X_test after encoding")
print(X_test)
print("---------------------------------------------")
print("loan_new after encoding")
print(data_new)
```

```
X_train after encoding
     Gender  Married  Dependents  Education  Income
Coapplicant_Income  \
413       1        1           0          1    2253
2033.0
339       0        0           0          0    4160
0.0
445       1        1           1          0    3466
1210.0
258       1        1           0          0   14683
2100.0
566       1        0           0          0    3333
0.0
..      ...      ...         ...        ...     ...                 ..
.
115       1        1           1          0   14583
0.0
40        1        0           0          0    3600
0.0
457       1        1           0          0    3708
2569.0
158       1        0           0          0    2980
2083.0
498       1        1           1          0    2895
0.0

     Loan_Tenor  Credit_History  Property_Area
413       144.0             1.0              0
339       144.0             1.0              1
445       144.0             1.0              0
258       144.0             1.0              0
566       144.0             1.0              2
..          ...             ...            ...
115        72.0             1.0              0
40        144.0             1.0              2
457       144.0             1.0              2
158       144.0             1.0              0
498       144.0             1.0              1
```

```
[384 rows x 9 columns]
---------------------------------------------
X_test after encoding
     Gender  Married  Dependents  Education  Income
Coapplicant_Income  \
385       1        0           1          0    3667
0.0
154       1        0           0          0    3254
0.0
447       1        1           0          0    3539
1376.0
543       1        1           1          1    2239
2524.0
525       1        1           2          0   17500
0.0
..      ...      ...         ...        ...     ...                  ..
.
485       1        1           1          1    1958
2436.0
55        1        1           2          0    2708
1167.0
584       1        1           1          0    2787
1917.0
361       1        1           2          0    5000
3667.0
528       1        0           1          1    2679
1302.0

     Loan_Tenor  Credit_History  Property_Area
385        72.0             1.0              2
154       144.0             1.0              2
447       144.0             1.0              0
543       144.0             1.0              2
525       144.0             1.0              0
..          ...             ...            ...
485       144.0             1.0              0
55        144.0             1.0              1
584       144.0             0.0              0
361       144.0             1.0              1
528       144.0             1.0              1

[129 rows x 9 columns]
---------------------------------------------
loan_new after encoding
      Loan_ID  Gender  Married  Dependents  Education  Income  \
0    LP001015       1        1           0          0    5720
1    LP001022       1        1           1          0    3076
2    LP001031       1        1           2          0    5000
4    LP001051       1        0           0          1    3276
```

```
5    LP001054       1       1           0          1    2165
..        ...     ...     ...         ...        ...     ...
361  LP002969       1       1           1          0    2269
362  LP002971       1       1           3          1    4009
363  LP002975       1       1           0          0    4158
365  LP002986       1       1           0          0    5000
366  LP002989       1       0           0          0    9200

     Coapplicant_Income  Loan_Tenor  Credit_History  Property_Area
0                     0       144.0             1.0              2
1                  1500       144.0             1.0              2
2                  1800       144.0             1.0              2
4                     0       144.0             1.0              2
5                  3422       144.0             1.0              2
..                  ...         ...             ...            ...
361                2167       144.0             1.0              1
362                1777       144.0             1.0              2
363                 709       144.0             1.0              2
365                2393       144.0             1.0              0
366                   0        72.0             1.0              0

[314 rows x 10 columns]
```

```python
# (c)
(v)--------------------------------------------------------------------
---------
# categorical targets are encoded
categorical_columns_y_old = categorical_columns_old[-1:]
label_encoder_y = LabelEncoder()

for i in range(len(categorical_columns_y_old)):
    y_train[categorical_columns_y_old[i]] =
label_encoder_y.fit_transform(y_train[categorical_columns_y_old[i]])
    y_test[categorical_columns_y_old[i]] =
label_encoder_y.transform(y_test[categorical_columns_y_old[i]])

print("y_train after encoding")
print(y_train)
print("--------------------------------------------")
print("y_test after encoding")
print(y_test)
```

```
y_train after encoding
     Max_Loan_Amount  Loan_Status
413           146.01            1
339           139.66            1
445           165.67            1
258           545.86            0
566            97.98            1
..               ...          ...
```

```
115           297.49            1
40            111.44            0
457           246.36            0
158           185.18            1
498            75.91            1

[384 rows x 2 columns]
---------------------------------------------
y_test after encoding
     Max_Loan_Amount  Loan_Status
385            22.41            1
154            94.00            1
447           177.72            0
543           170.06            1
525           812.00            1
..               ...          ...
485           151.46            1
55            125.30            1
584           167.08            0
361           366.82            1
528           130.64            1

[129 rows x 2 columns]

# (c)
(vi)-----------------------------------------------------------------------
----------
# numerical features are standardized
numerical_columns_X = numerical_columns_old[:-2]

for feature in numerical_columns_X:
    mean_value = X_train[feature].mean()
    std_value = X_train[feature].std()
    X_train[feature] = (X_train[feature] - mean_value) / std_value
    X_test[feature] = (X_test[feature] - mean_value) / std_value
    data_new[feature] = (data_new[feature] - mean_value) / std_value

print("X_train after standardization")
print(X_train)
print("---------------------------------------------")
print("X_test after standardization")
print(X_test)
print("---------------------------------------------")
print("loan_new after standardization")
print(data_new)

X_train after standardization
     Gender  Married  Dependents  Education    Income
Coapplicant_Income  \
413       1        1           0          1 -0.611908
```

```
0.297003
339       0        0           0           0 -0.208621            -
0.734998
445       1        1           1           0 -0.355386            -
0.120772
258       1        1           0           0  2.016752
0.331014
566       1        0           0           0 -0.383513            -
0.734998
..      ...      ...         ...         ...       ...
...
115       1        1           1           0  1.995605            -
0.734998
40        1        0           0           0 -0.327048            -
0.734998
457       1        1           0           0 -0.304209
0.569090
158       1        0           0           0 -0.458164
0.322384
498       1        1           1           0 -0.476140            -
0.734998

     Loan_Tenor  Credit_History  Property_Area
413    0.282617             1.0              0
339    0.282617             1.0              1
445    0.282617             1.0              0
258    0.282617             1.0              0
566    0.282617             1.0              2
..          ...             ...            ...
115   -2.863034             1.0              0
40     0.282617             1.0              2
457    0.282617             1.0              2
158    0.282617             1.0              0
498    0.282617             1.0              1

[384 rows x 9 columns]
---------------------------------------------
X_test after standardization
     Gender  Married  Dependents  Education    Income
Coapplicant_Income  \
385       1        0           1           0 -0.312879            -
0.734998
154       1        0           0           0 -0.400219            -
0.734998
447       1        1           0           0 -0.339948            -
0.036506
543       1        1           1           1 -0.614869
0.546247
525       1        1           2           0  2.612483            -
```

```
0.734998
..      ...      ...         ...        ...       ...
...
485       1        1           1          1 -0.674294
0.501576
55        1        1           2          0 -0.515686           -
0.142600
584       1        1           1          0 -0.498979
0.238119
361       1        1           2          0 -0.030980
1.126462
528       1        0           1          1 -0.521819           -
0.074071

     Loan_Tenor  Credit_History  Property_Area
385   -2.863034             1.0              2
154    0.282617             1.0              2
447    0.282617             1.0              0
543    0.282617             1.0              2
525    0.282617             1.0              0
..          ...             ...            ...
485    0.282617             1.0              0
55     0.282617             1.0              1
584    0.282617             0.0              0
361    0.282617             1.0              1
528    0.282617             1.0              1

[129 rows x 9 columns]
---------------------------------------------
loan_new after standardization
      Loan_ID  Gender  Married  Dependents  Education    Income  \
0    LP001015       1        1           0          0  0.121283
1    LP001022       1        1           1          0 -0.437862
2    LP001031       1        1           2          0 -0.030980
4    LP001051       1        0           0          1 -0.395567
5    LP001054       1        1           0          1 -0.630518
..        ...     ...      ...         ...        ...       ...
361  LP002969       1        1           1          0 -0.608524
362  LP002971       1        1           3          1 -0.240554
363  LP002975       1        1           0          0 -0.209044
365  LP002986       1        1           0          0 -0.030980
366  LP002989       1        0           0          0  0.857223

     Coapplicant_Income  Loan_Tenor  Credit_History  Property_Area
0             -0.734998    0.282617             1.0              2
1              0.026439    0.282617             1.0              2
2              0.178727    0.282617             1.0              2
4             -0.734998    0.282617             1.0              2
5              1.002094    0.282617             1.0              2
..                  ...         ...             ...            ...
```

```
361            0.365025    0.282617            1.0              1
362            0.167051    0.282617            1.0              2
363           -0.375092    0.282617            1.0              2
365            0.479748    0.282617            1.0              0
366           -0.734998   -2.863034            1.0              0

[314 rows x 10 columns]

#
----------------------------------------------------------------------
------
# Convert to numpy array
X_train = X_train.to_numpy().reshape((-1, num_of_cols - 3))
X_test = X_test.to_numpy().reshape((-1, num_of_cols - 3))

# y_train for linear
y_train_lin = y_train.iloc[:, :1]
y_test_lin = y_test.iloc[:, :1]
y_train_lin = y_train_lin.to_numpy()
y_test_lin = y_test_lin.to_numpy()

# y_train for logistic
y_train_log = y_train.iloc[:, 1:]
y_test_log = y_test.iloc[:, 1:]
y_train_log = y_train_log.to_numpy()
y_test_log = y_test_log.to_numpy()

#
(d)-------------------------------------------------------------------
---------
# Fit a linear regression model to the data to predict the loan amount
print('Linear Regression')
model = LinearRegression()
model.fit(X_train, y_train_lin)
print('Coefficients: \n', model.coef_, " ", model.intercept_)
y_pred = model.predict(X_test)

Linear Regression
Coefficients:
 [[ 12.72936668   3.74565423   6.23014953 -15.77484775 118.57504181
   58.4506875   52.09701305   6.12000006  -8.22183511]]
[217.88119749]

#
(e)-------------------------------------------------------------------
---------
# Evaluate the linear regression model using sklearn's R^2 score
r2 = r2_score(y_test_lin, y_pred)
print(f"R² Score: {r2}")
print()
```

```
print("*"*70)
print()
```

```
R² Score: 0.824933001935585

**********************************************************************
```

```
#
(f)----------------------------------------------------------------------
---------
# Fit a logistic regression model to the data to predict the loan
status
print('Logistic Regression')

# add 1s
X_train = np.hstack((np.ones((len(X_train), 1)), X_train))
X_test = np.hstack((np.ones((len(X_test), 1)), X_test))
theta = np.zeros((X_train.shape[1], 1))

def sigmoid(z):
    return 1 / (1 + np.exp(-z))

def cost(theta, X, y):
    m = len(y)
    h = sigmoid(np.dot(X, theta))
    J = (-1 / m) * (np.dot(y.T, np.log(h)) + np.dot((1 - y).T,
np.log(1 - h)))
    return J

def gradient(theta, X, y):
    m = len(y)
    h = sigmoid(np.dot(X, theta))
    grad = (1 / m) * np.dot(X.T, (h - y.reshape(-1, 1)))
    return grad

def gradient_descent(X, y, theta, alpha, num_iters):
    costs = []
    for _ in range(num_iters):
        theta = theta - alpha * gradient(theta, X, y)
        cost_val = cost(theta, X, y)
        costs.append(cost_val)

    return theta, costs

alpha = 0.01
num_iters = 3000

theta, cost_history = gradient_descent(X_train, y_train_log, theta,
alpha, num_iters)
print('theta',theta)
```

```
y_pred_num = sigmoid(np.dot(X_test, theta))
y_pred_log = np.round(y_pred_num)
```

```
Logistic Regression
theta [[-0.3469397 ]
 [-0.13837104]
 [ 0.23788795]
 [-0.02059603]
 [-0.57503303]
 [ 0.07508358]
 [ 0.06602735]
 [ 0.03337178]
 [ 1.54551126]
 [ 0.00321776]]
```

```
#
(g)-----------------------------------------------------------------------
---------
# Write a function (from scratch) to calculate the accuracy of the
model
def calculate_accuracy(y_test_log, y_pred_log):
    correct_predictions = (y_pred_log == y_test_log).sum()
    total_samples = len(y_test_log)
    accuracy = correct_predictions / total_samples
    return accuracy

print('Accuracy:',calculate_accuracy(y_test_log,y_pred_log))
```

```
Accuracy: 0.813953488372093
```

```
#
(j)-----------------------------------------------------------------------
---------
#  predict the loan amounts
X_data_new = data_new.iloc[:, 1:]
X_data_new = X_data_new.to_numpy().reshape((-1, num_of_cols - 3))
loan_amounts_pred = model.predict(X_data_new)
print("loan_amounts_pred:")
print(loan_amounts_pred)
print()
print("*"*70)
print()
loan_amounts_pred = loan_amounts_pred.flatten().tolist()
# add Max Loan Amount column to excel sheet
loan_amounts_column = 'Max_Loan_Amount'
data_new_copy['Max_Loan_Amount'] = loan_amounts_pred
```

```
loan_amounts_pred:
[[ 210.17606732]
 [ 194.61205705]
```

```
 [ 257.98955094]
 [ 129.37005697]
 [ 206.79061113]
 [ 104.76297616]
 [ 171.07060209]
 [ 311.80930931]
 [ 186.8358474 ]
 [ 124.73101481]
 [ 180.96599837]
 [ 386.55300728]
 [ 175.49287914]
 [ 211.33153973]
 [ 280.64575693]
 [ 198.47200298]
 [ 532.96797586]
 [  42.89616886]
 [ 152.60755818]
 [ -58.46343733]
 [ 128.90280715]
 [ 332.11084369]
 [ 785.42438334]
 [ 366.07749892]
 [  43.5668982 ]
 [ 104.85195434]
 [ 260.25683129]
 [ 203.10245846]
 [ 216.87725935]
 [ 181.17291571]
 [ 144.08717591]
 [ 219.17560366]
 [ 206.43045874]
 [ 209.26399019]
 [ 212.59197267]
 [ 232.16505406]
 [ 141.64347056]
 [ 163.12586095]
 [ 311.35610376]
 [ 142.19514044]
 [ 170.95633992]
 [ 285.90431833]
 [ 184.27908943]
 [ 260.25727307]
 [  48.20661331]
 [ 183.69380874]
 [ 125.83497937]
 [ 161.31982795]
 [ 119.1499875 ]
 [ 246.97521783]
 [  48.1334242 ]
```

```
 [ 170.81101792]
 [ 251.33837607]
 [ 195.52610108]
 [ 157.78155077]
 [ 182.24830675]
 [ 254.42140763]
 [ 170.92563295]
 [ 155.2302217 ]
 [ 267.19561081]
 [ 290.17269465]
 [ 183.13183739]
 [  36.83082856]
 [ 243.11270789]
 [ 284.61069116]
 [ 254.76315919]
 [ 216.94025855]
 [ 238.82235799]
 [ 287.03458452]
 [ 253.36596564]
 [ 208.0551167 ]
 [1904.16255872]
 [ 286.71763966]
 [ 295.05187842]
 [  16.70293699]
 [ 285.36181418]
 [ 213.16729494]
 [ 148.10080008]
 [ 208.81118344]
 [ 193.54446845]
 [ 440.8520823 ]
 [ 264.8026755 ]
 [ 239.99943985]
 [ 259.87121156]
 [ 237.89997227]
 [ 281.16831699]
 [ 259.16020992]
 [ 329.14505436]
 [ 191.07783756]
 [ 222.08874323]
 [ 170.79314822]
 [ -16.62904997]
 [ 186.10533191]
 [ 215.04357049]
 [ 187.80530005]
 [ 204.44123933]
 [ 177.39864781]
 [ 135.95124045]
 [ 248.72923888]
 [ 328.73288609]
```

```
 [  94.93485961]
 [ 167.433558  ]
 [ 211.82610074]
 [ 141.35844328]
 [ 254.52653934]
 [ 215.72341486]
 [ 326.31959293]
 [ 357.59567267]
 [ 188.24732172]
 [ 219.86961782]
 [ 267.27147476]
 [ -23.82452694]
 [ 188.00208025]
 [ 164.47230385]
 [ 218.25540086]
 [ 147.2717984 ]
 [  26.73400622]
 [ 173.51509627]
 [ 208.48743422]
 [ 226.70632048]
 [ 191.85371098]
 [ 159.06618919]
 [ 249.56948686]
 [  68.27758222]
 [ 320.14780514]
 [ 137.59189208]
 [ 275.7806559 ]
 [ 221.37793925]
 [ 210.94407573]
 [ 267.85676762]
 [ 179.72914291]
 [ 204.81943272]
 [ 182.86275713]
 [ 217.38937999]
 [ 114.31936299]
 [ 311.6576334 ]
 [ 250.71137481]
 [ 298.50322418]
 [ 265.38498009]
 [ 309.78477378]
 [ 139.42305785]
 [ 202.9950143 ]
 [ 134.61609502]
 [ 106.1953812 ]
 [ 142.01974819]
 [ 236.99091635]
 [ 205.77208934]
 [ 166.05687225]
 [ 165.9609745 ]
```

```
 [ 192.28629611]
 [ 214.20716526]
 [  53.06527295]
 [ 176.25149836]
 [ 355.72079059]
 [ 225.01061716]
 [ 224.23510167]
 [ 264.76858276]
 [ 271.02385255]
 [ 188.48298569]
 [ 293.21134969]
 [ 220.14783245]
 [ 317.29484115]
 [ 395.4787679 ]
 [ 410.79846738]
 [  48.77438941]
 [ 165.38115892]
 [ 255.0653242 ]
 [ 205.81531493]
 [ 411.76796184]
 [ 209.26399019]
 [ 195.33791698]
 [ 165.03615968]
 [ 157.54470657]
 [ 173.61539598]
 [ 410.7490678 ]
 [ 157.33089696]
 [ 211.88814701]
 [ 141.24175274]
 [ 212.20192417]
 [ 282.715016  ]
 [ 190.41092521]
 [ 155.9116597 ]
 [ 109.49616216]
 [ 278.26015542]
 [ 228.66284741]
 [ 222.57317246]
 [ 132.61116461]
 [ -44.81477554]
 [ 344.30172973]
 [ 270.76128189]
 [ 230.39776983]
 [ 219.50315415]
 [ 245.29984905]
 [ 145.24858052]
 [ 194.82249619]
 [ 114.08505252]
 [ 183.21904317]
 [ 288.59343664]
```

```
 [ 221.66665303]
 [ 193.43068016]
 [ 875.56235834]
 [  -3.98381419]
 [ 255.29572979]
 [ 161.15303906]
 [ 195.12799357]
 [ 264.00700044]
 [ 649.37890788]
 [ 167.59976261]
 [ 280.90281697]
 [ 143.61317021]
 [ 211.3247077 ]
 [ 187.54775966]
 [ 191.29270543]
 [  95.54879411]
 [ 268.84914409]
 [ 158.063174  ]
 [ -11.09209045]
 [ 282.82734444]
 [ 165.72460218]
 [ 210.16936626]
 [ 200.52853303]
 [ 144.98684717]
 [ 166.57214078]
 [ 277.54921718]
 [ 269.11335883]
 [ 225.5996577 ]
 [ 195.73015867]
 [ 561.84574214]
 [ 203.97974119]
 [ 285.30603358]
 [ 259.88227712]
 [ 258.04687828]
 [ 167.40618457]
 [ 170.44320704]
 [ 266.48024723]
 [ 699.25541048]
 [ 236.46442712]
 [ 140.13437549]
 [ 207.60717937]
 [ 229.73527211]
 [  39.27257756]
 [ 189.88579609]
 [ 186.56577401]
 [ 208.56508695]
 [ 289.34819933]
 [ 663.37398084]
 [ 292.38620472]
```

```
 [ 257.86951854]
 [ 187.94947214]
 [ 400.61363102]
 [ 226.25911141]
 [ 242.65050693]
 [ 145.68834767]
 [ 163.42967104]
 [ 178.23938489]
 [ 238.33940834]
 [ 159.1541079 ]
 [ 152.42786301]
 [ 198.95859085]
 [ 261.04132523]
 [ 224.83024086]
 [ 160.9307969 ]
 [ 283.71514498]
 [ 176.63893227]
 [ 260.55876053]
 [ 307.39897618]
 [ 196.24959623]
 [ 295.03619946]
 [ 221.02538601]
 [  90.10146562]
 [ 121.50769645]
 [ 177.8562974 ]
 [ 173.52204371]
 [ 221.54450679]
 [ 202.04586552]
 [ 131.81803398]
 [ 116.06407851]
 [ 627.01778424]
 [ 216.91536284]
 [ 148.95307508]
 [ 226.37954151]
 [ 318.21996332]
 [ 221.32872188]
 [ 328.64353628]
 [ 215.88384084]
 [ 180.31679722]
 [ 170.86696284]
 [ 135.63871224]
 [ 170.44331824]
 [ 138.88843936]
 [ 238.46692684]
 [ 120.19950111]
 [ 305.88501231]
 [  10.53947906]
 [ 190.82879519]
 [ 251.41046159]
```

```
 [ 294.41492789]
 [ 220.4311824 ]
 [ 119.13908831]
 [ 190.08279552]
 [ 100.03160868]
 [ 312.97872447]
 [ 239.60905773]
 [ 323.37919207]
 [  46.73229558]
 [ 312.07538905]
 [ 225.47789263]
 [ 125.68600354]
 [ 250.6543031 ]
 [ 202.38820799]
 [ 222.91219882]
 [ 192.04425875]
 [ 279.56783667]
 [ 146.25920456]]

**********************************************************************
```

```
#  predict the status
X_data_new = np.hstack((np.ones((len(X_data_new), 1)), X_data_new))
status_pred_num = sigmoid(np.dot(X_data_new, theta))
status_pred_log = np.round(status_pred_num)
status_pred_log = status_pred_log.flatten().tolist()
status_pred_log = [int(x) for x in status_pred_log]
decoded_status_pred_log =
label_encoder_y.inverse_transform(status_pred_log)
print("decoded_status_pred_log:")
print(decoded_status_pred_log)
print()
print("*"*70)
print()
# add Loan Status column to excel sheet
data_new_copy['Loan_Status'] = decoded_status_pred_log
```

```
decoded_status_pred_log:
['Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y'
'Y'
 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'Y' 'Y' 'Y'
'Y'
 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'Y' 'N' 'Y' 'Y' 'Y'
'Y'
 'N' 'Y' 'Y' 'N' 'N' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N'
'Y'
 'N' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y'
'N'
 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'N' 'Y' 'Y' 'Y' 'N'
```

```
'N'
 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'N' 'Y' 'Y'
'Y'
 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'Y'
'N'
 'N' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'N' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y'
'Y'
 'Y' 'Y' 'Y' 'Y' 'N' 'N' 'Y' 'Y' 'N' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y'
'Y'
 'Y' 'Y' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'Y'
'Y'
 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'N' 'N' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'N' 'Y' 'Y'
'Y'
 'Y' 'N' 'Y' 'Y' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'N' 'Y'
'Y'
 'Y' 'Y' 'N' 'N' 'Y' 'Y' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y'
'Y'
 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'Y'
'Y'
 'Y' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y'
'Y'
 'Y' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'N'
'Y'
 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y' 'Y']

**********************************************************************


data_new_copy.to_csv('loan_new1.csv', index=False)
print("Predicted Date is written to loan_new1.csv")
print("loan_new1 data")
print(data_new_copy)

Predicted Date is written to loan_new1.csv
loan_new1 data
      Loan_ID Gender Married Dependents     Education  Income  \
0    LP001015   Male     Yes          0      Graduate    5720
1    LP001022   Male     Yes          1      Graduate    3076
2    LP001031   Male     Yes          2      Graduate    5000
4    LP001051   Male      No          0  Not Graduate    3276
5    LP001054   Male     Yes          0  Not Graduate    2165
..        ...    ...     ...        ...           ...     ...
361  LP002969   Male     Yes          1      Graduate    2269
362  LP002971   Male     Yes         3+  Not Graduate    4009
363  LP002975   Male     Yes          0      Graduate    4158
365  LP002986   Male     Yes          0      Graduate    5000
366  LP002989   Male      No          0      Graduate    9200

     Coapplicant_Income  Loan_Tenor  Credit_History Property_Area  \
0                     0       144.0             1.0         Urban
```

```
1                     1500         144.0               1.0         Urban
2                     1800         144.0               1.0         Urban
4                        0         144.0               1.0         Urban
5                     3422         144.0               1.0         Urban
..                     ...           ...               ...           ...
361                   2167         144.0               1.0     Semiurban
362                   1777         144.0               1.0         Urban
363                    709         144.0               1.0         Urban
365                   2393         144.0               1.0         Rural
366                      0          72.0               1.0         Rural

     Max_Loan_Amount Loan_Status
0         210.176067           Y
1         194.612057           Y
2         257.989551           Y
4         129.370057           Y
5         206.790611           Y
..               ...         ...
361       202.388208           Y
362       222.912199           Y
363       192.044259           Y
365       279.567837           Y
366       146.259205           Y

[314 rows x 12 columns]
```