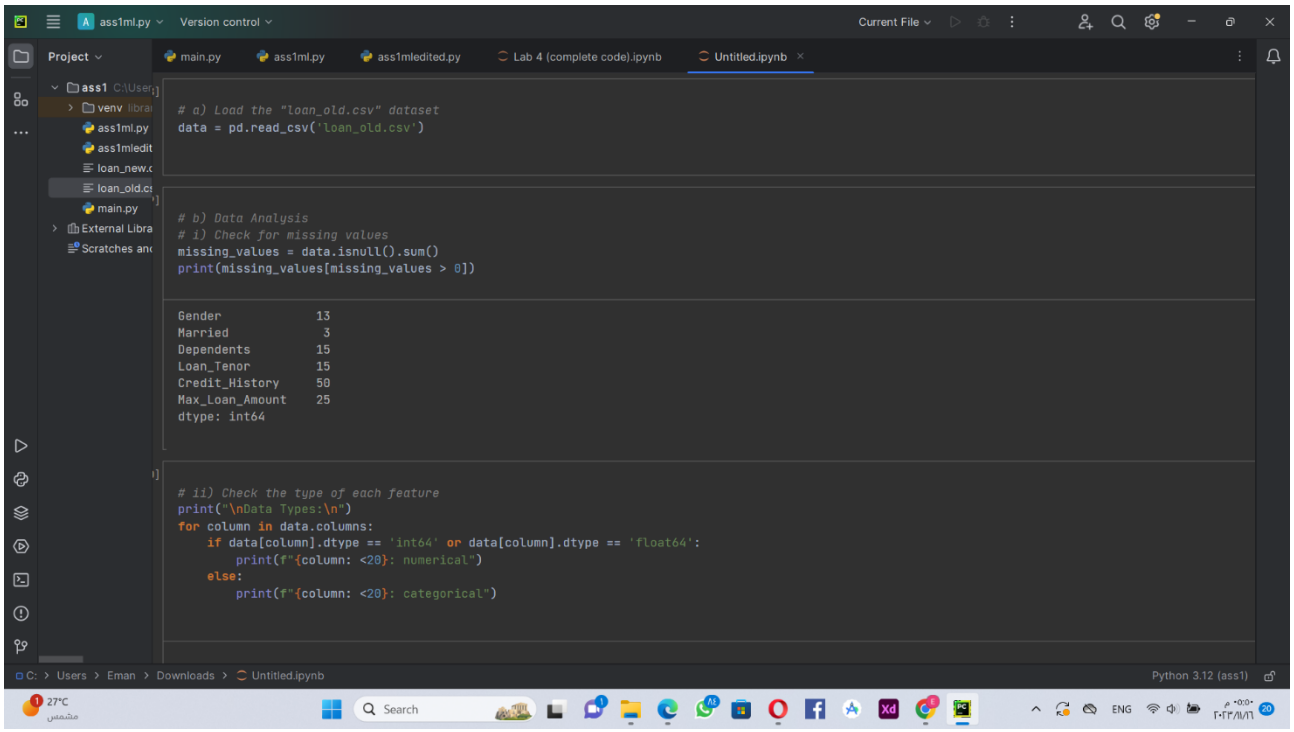
Machine Learning

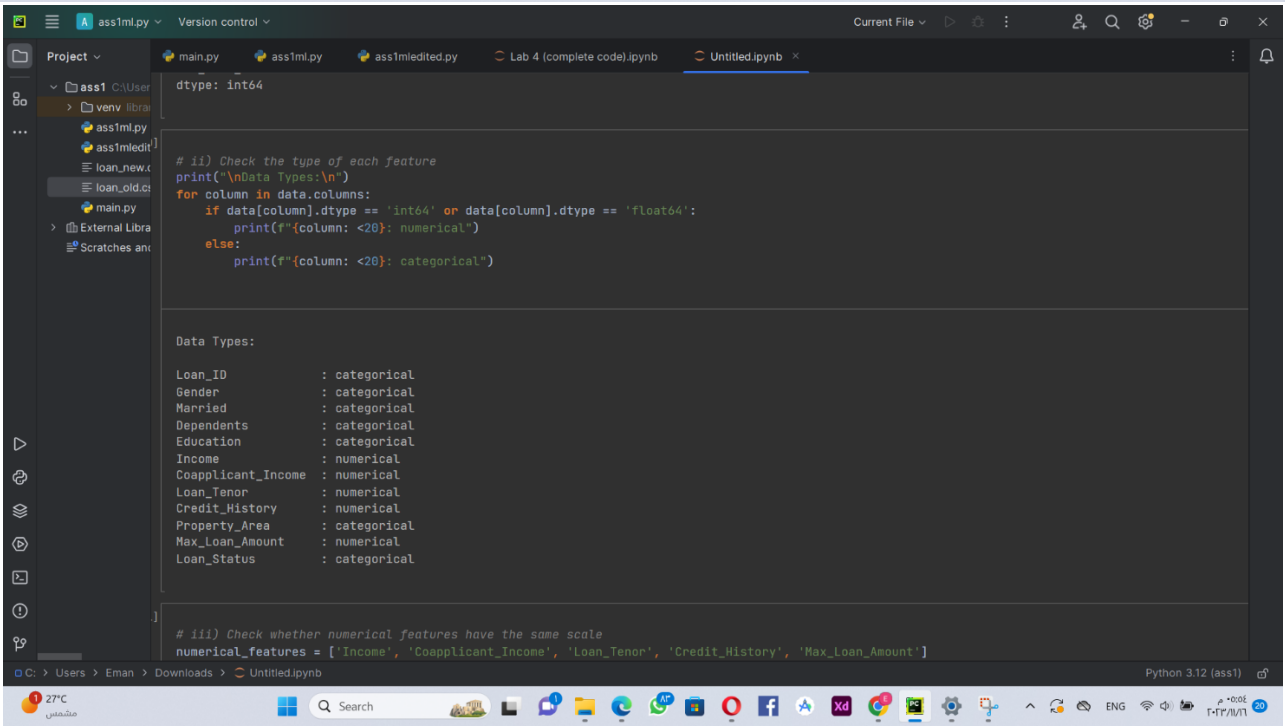
Assignment 1

Name	ID
Eman Fathy Abo Alhassan	20200105
Sara Ahmed Sayed	20200214
Fatma Mahmoud Ramadan	20201134
Basmala Magdy Mohamed	20201045
Esraa Osama Mohamed	20201014

a-load the dataset

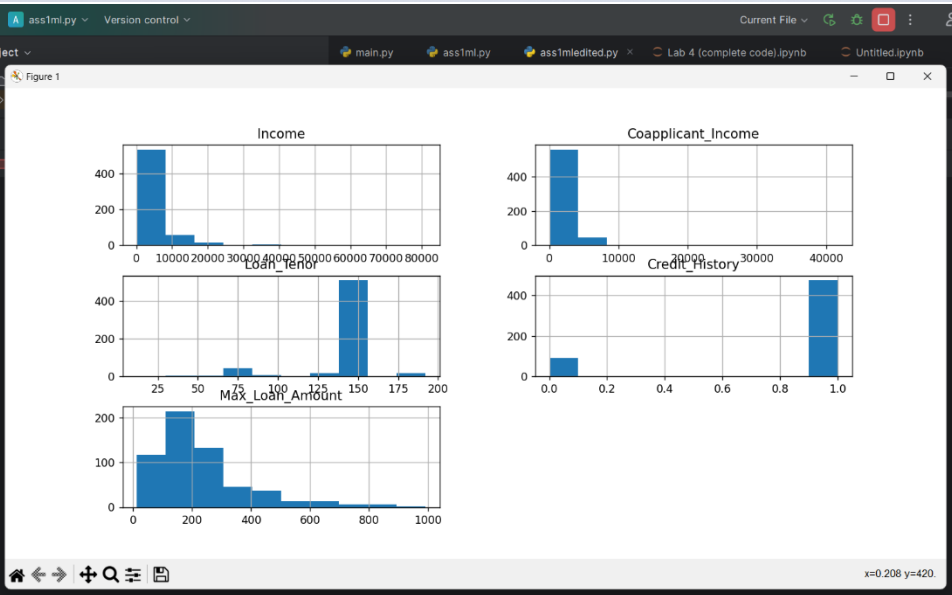
.b-Data analysis

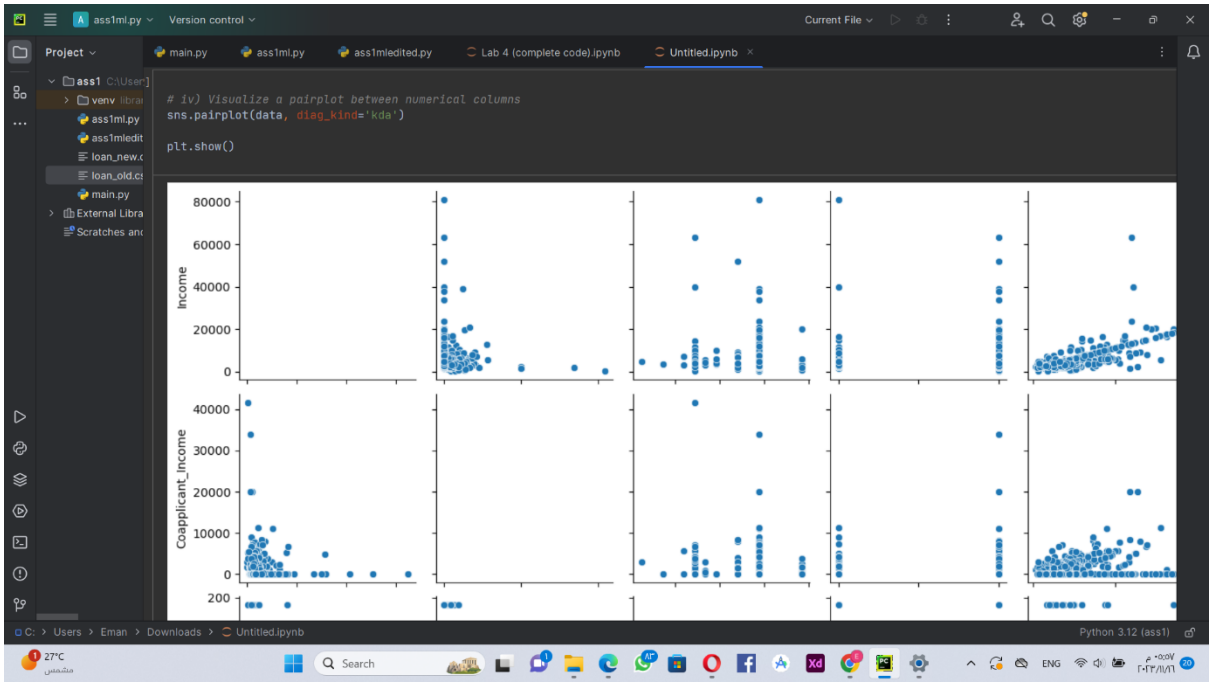




```
# iii) Check whether numerical features have the same scale
numerical_features = ['Income', 'Coapplicant_Income', 'Loan_Tenor', 'Credit_History', 'Max_Loan_Amount']

# Calculate the ranges
ranges = data[numerical_features].max() - data[numerical_features].min()

# Calculate the standard deviations
std_devs = data[numerical_features].std()

for feature in numerical_features:
    print(f"Range of {feature}: {ranges[feature]}")
    print(f"Standard deviation of {feature}: {std_devs[feature]}")
    print()

# histograms for each feature
data[numerical_features].hist(bins=10, figsize=(12, 6))
plt.show()
```

```
Range of Income: 80850.0
Standard deviation of Income: 6109.041673387174

Range of Coapplicant_Income: 41667.0
Standard deviation of Coapplicant_Income: 2926.2483692241917

Range of Loan_Tenor: 180.0
Standard deviation of Loan_Tenor: 23.366293783636753

Range of Credit_History: 1.0
Standard deviation of Credit_History: 0.3648783192364049

Range of Max_Loan_Amount: 977.66
Standard deviation of Max_Loan_Amount: 161.97696676527076
```




ass1ml.py
Version control
Current File
Project
main.py
ass1ml.py
ass1mledited.py
Lab 4 (complete code).ipynb
Untitled.ipynb
# iv) Visualize a pairplot between numerical columns
sns.pairplot(data, diag_kind='kda')
plt.show()
Income
Coapplicant_Income
Python 3.12 (ass1)

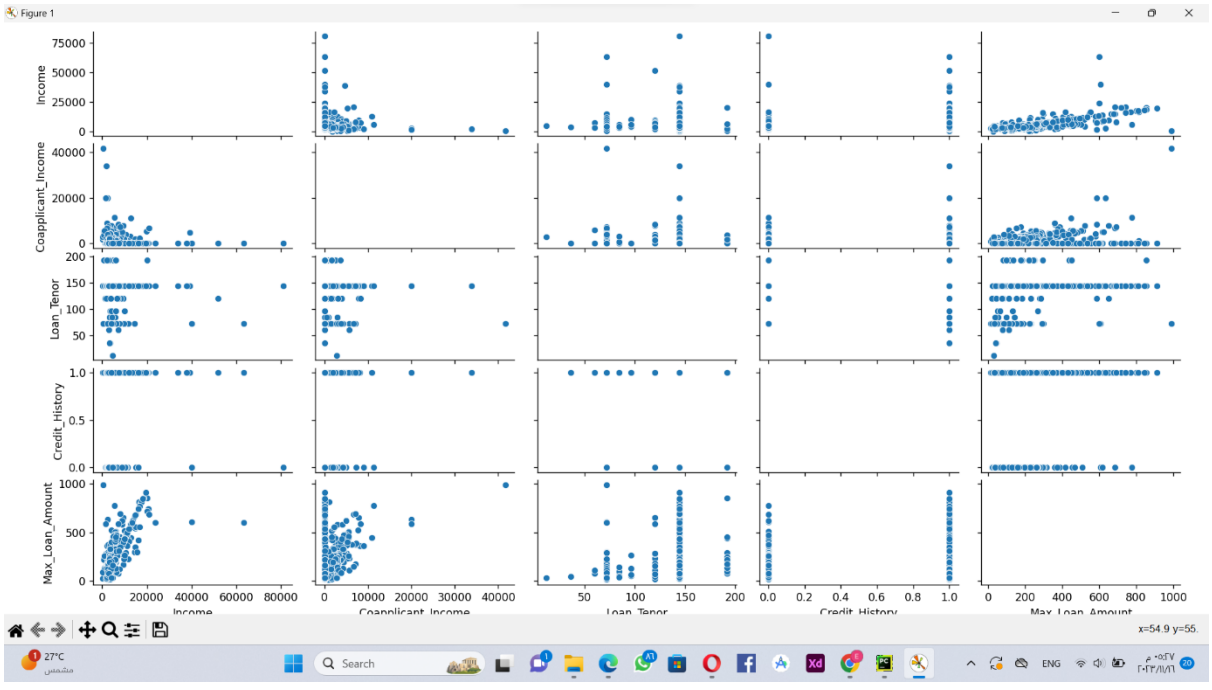
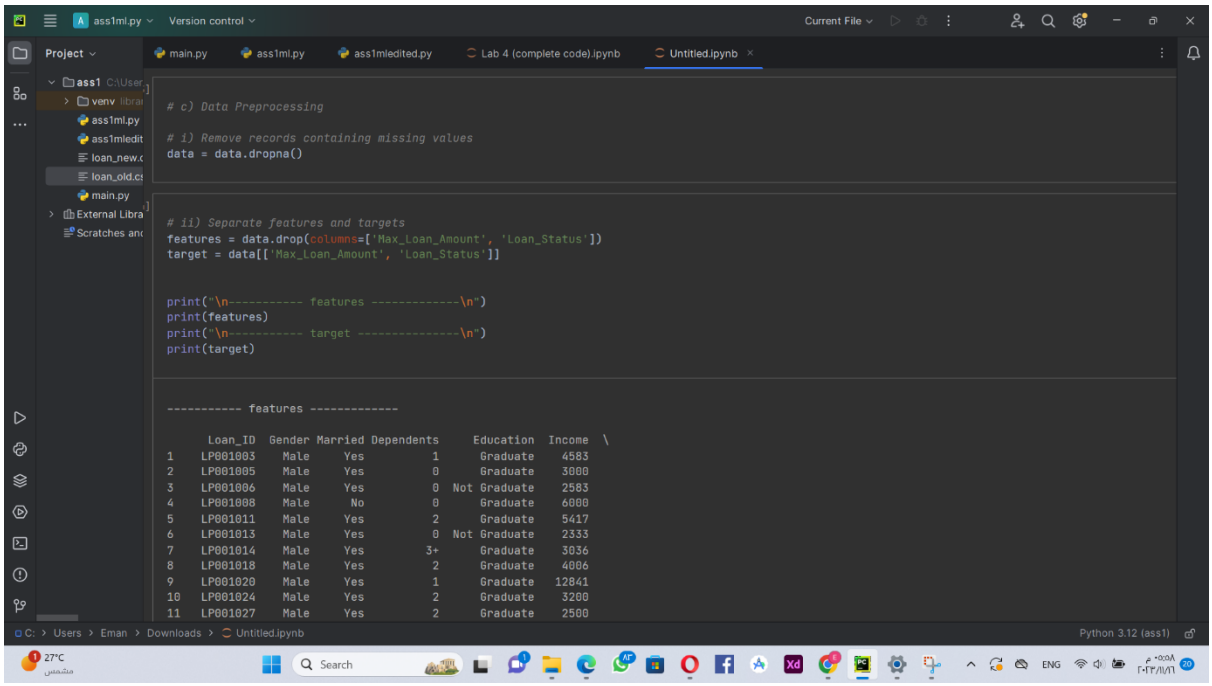
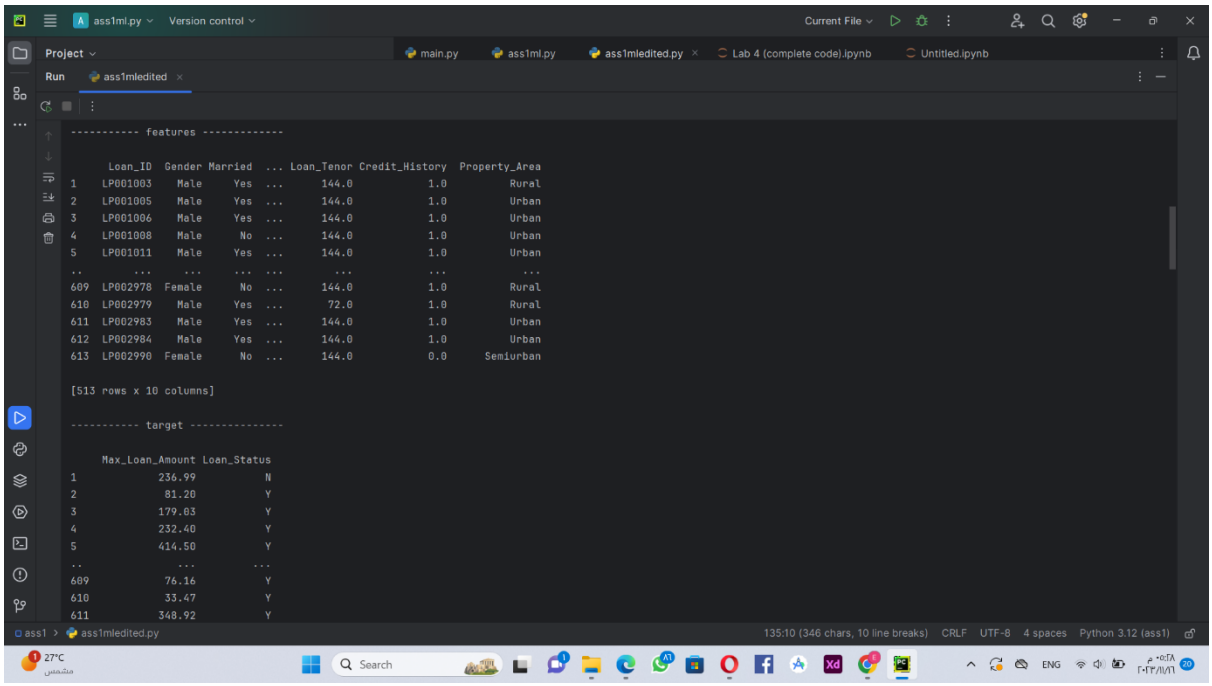
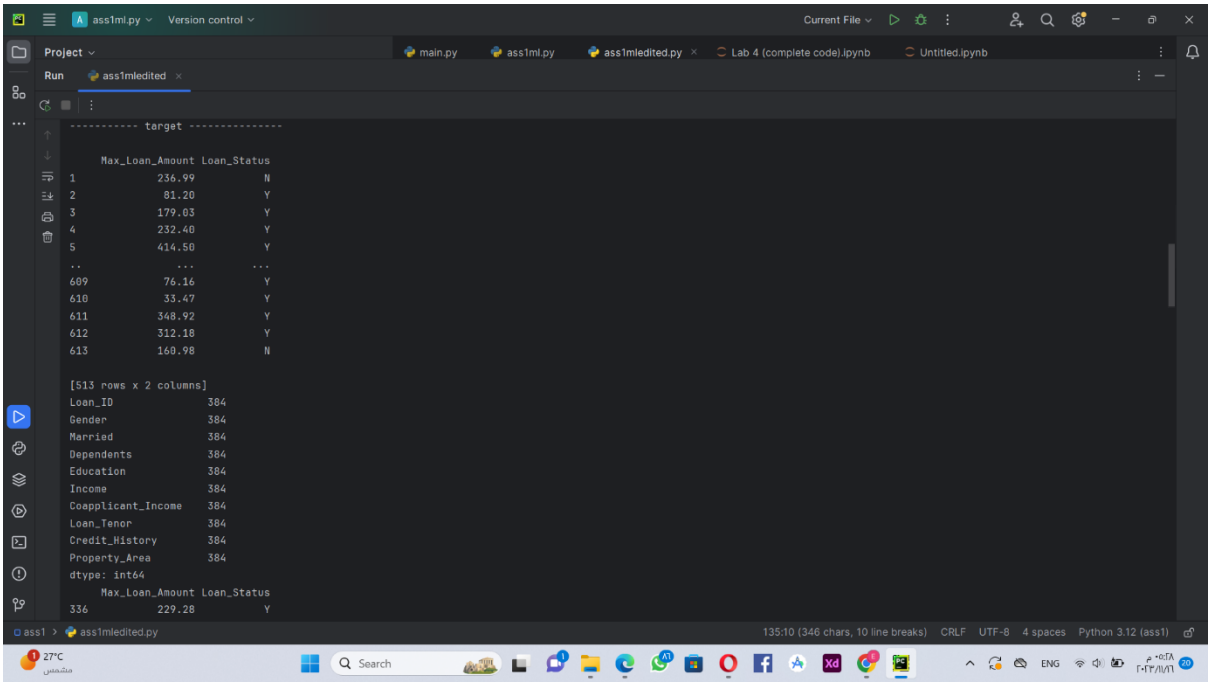


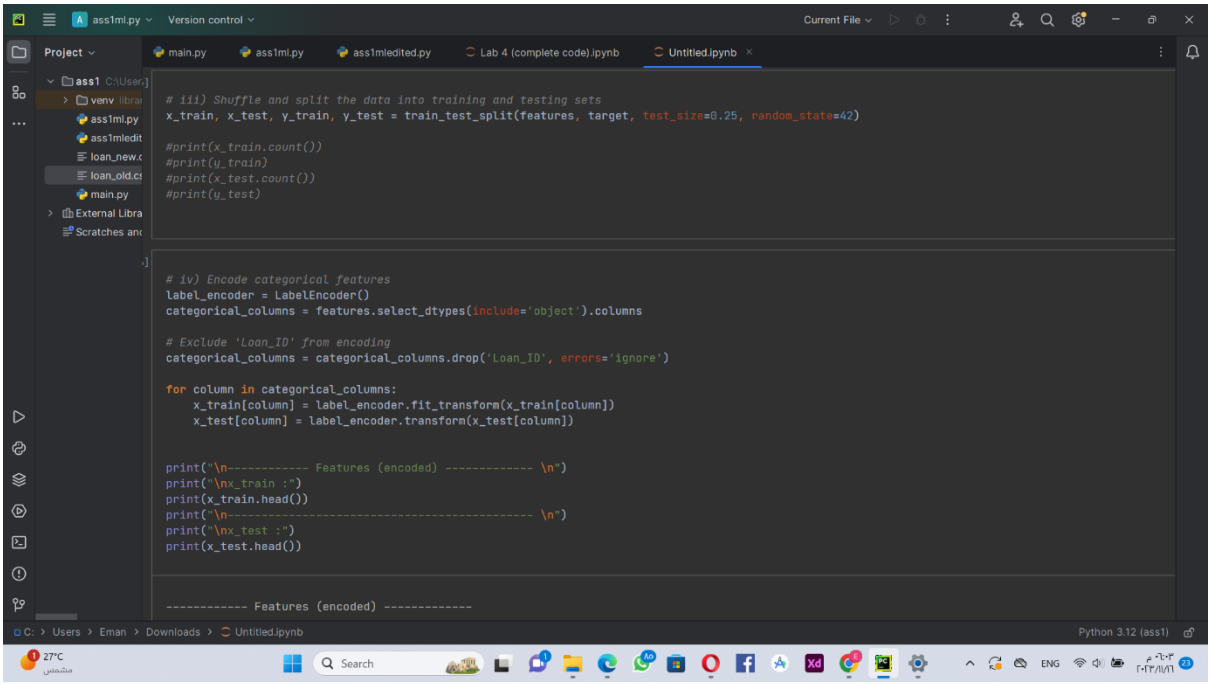
Figure 1
Income
Coapplicant_Income
Loan_Tenor
Credit_History
Max_Loan_Amount
x=54.9 y=55.

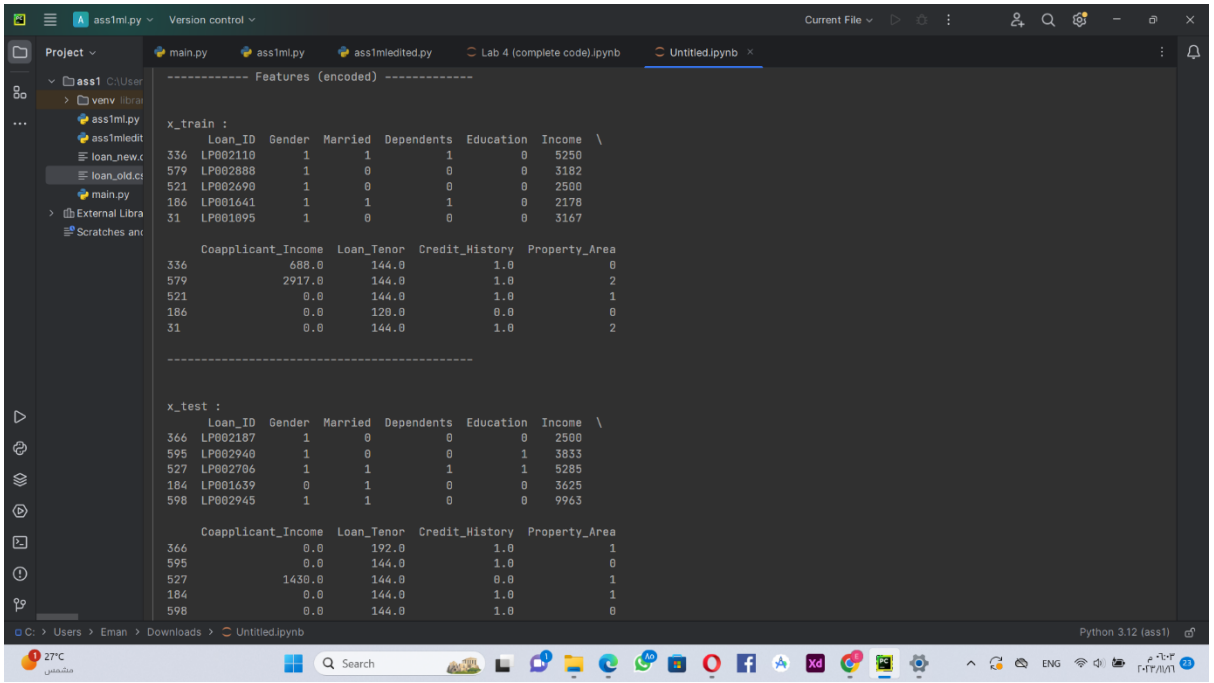
c-data preprocessing

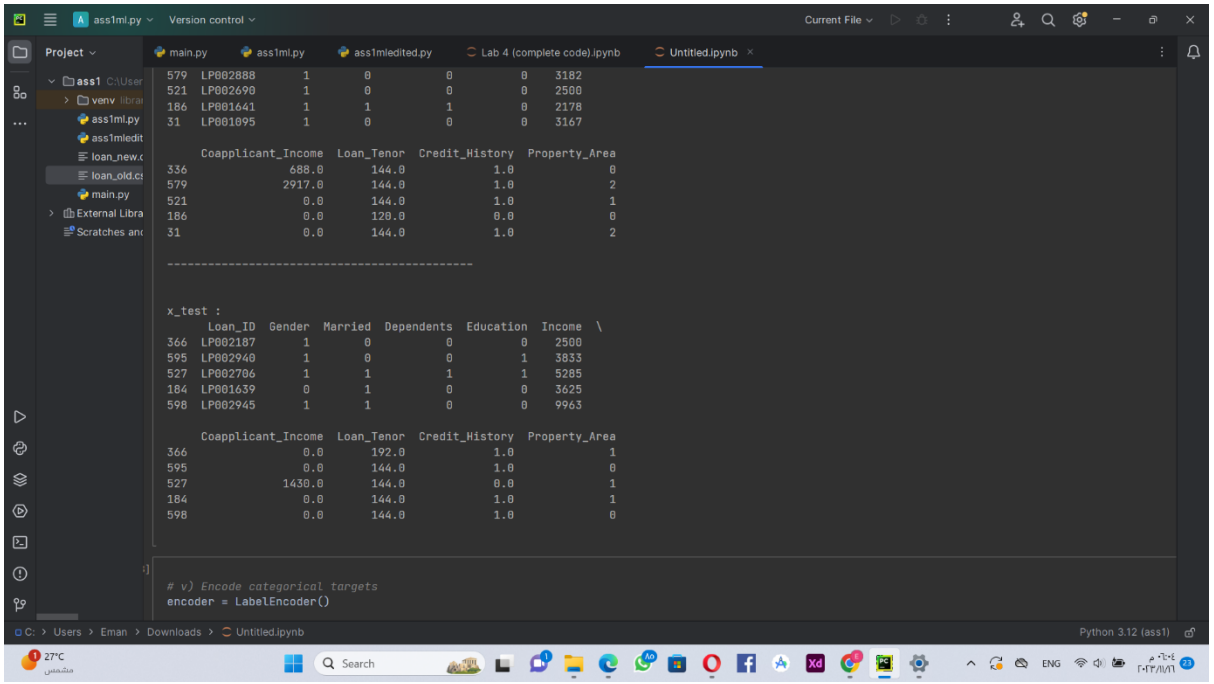


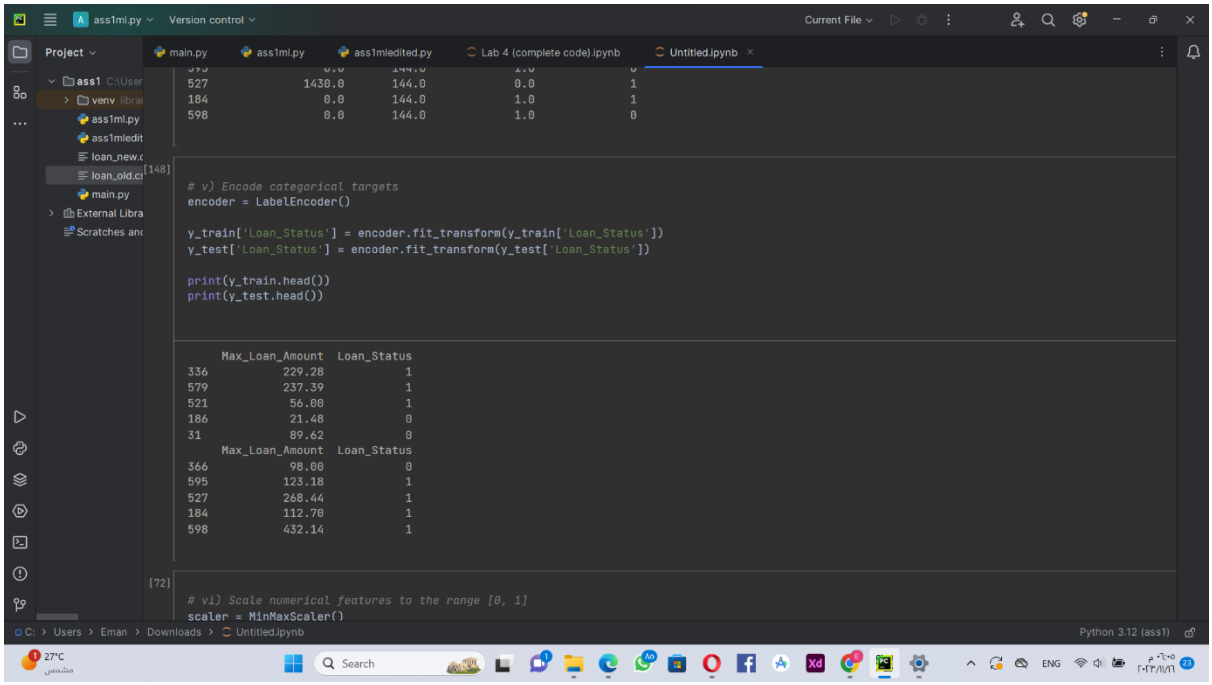


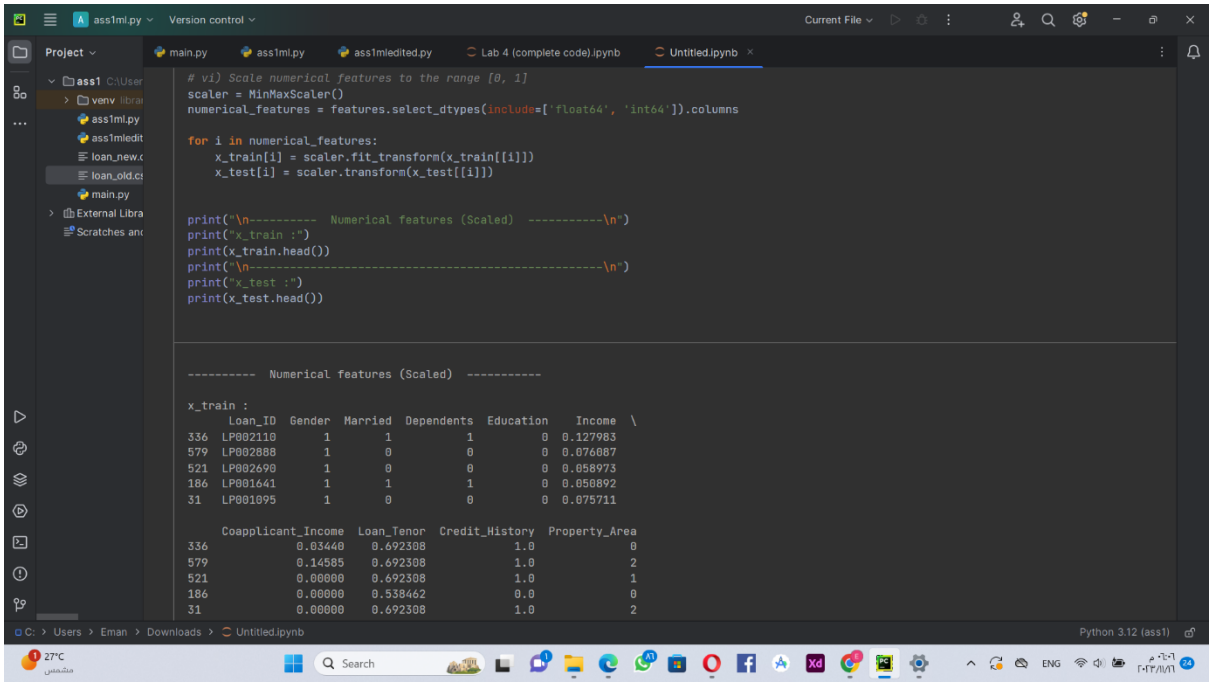

ass1ml.py
Version control
Current File
Project
main.py
ass1ml.py
ass1mledited.py
Lab 4 (complete code).ipynb
Untitled.ipynb
Run
ass1mledited
---------- target --------------
Max_Loan_Amount Loan_Status
1 236.99 N
2 81.20 Y
3 179.03 Y
4 232.40 Y
5 414.50 Y
.. ..
609 76.16 Y
610 33.47 Y
611 348.92 Y
612 312.18 Y
613 160.98 N
[513 rows x 2 columns]
Loan_ID 384
Gender 384
Married 384
Dependents 384
Education 384
Income 384
Coapplicant_Income 384
Loan_Tenor 384
Credit_History 384
Property_Area 384
dtype: int64
Max_Loan_Amount Loan_Status
336 229.28 Y
ass1
ass1mledited.py
135:10 (346 chars, 10 line breaks)
CRLF
UTF-8
4 spaces
Python 3.12 (ass1)
27°C
Search
ENG

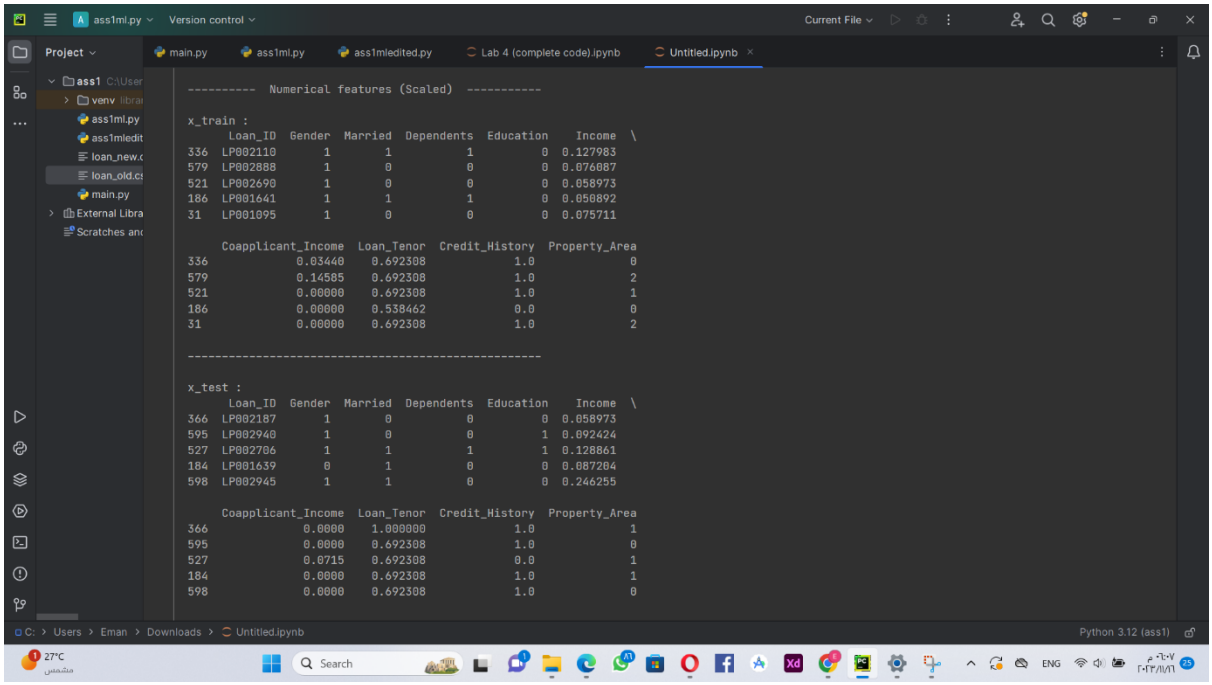

ass1ml.py
Version control
Current File
Project
main.py
ass1ml.py
ass1mledited.py
Lab 4 (complete code).ipynb
Untitled.ipynb
ass1
venv
ass1ml.py
ass1mledit
loan_new.c
loan_old.cs
main.py
External Libra
Scratches and
iii) Shuffle and split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(features, target, test_size=0.25, random_state=42)
#print(x_train.count())
#print(y_train)
#print(x_test.count())
#print(y_test)
# iv) Encode categorical features
label_encoder = LabelEncoder()
categorical_columns = features.select_dtypes(include='object').columns
# Exclude 'Loan_ID' from encoding
categorical_columns = categorical_columns.drop('Loan_ID', errors='ignore')
for column in categorical_columns:
x_train[column] = label_encoder.fit_transform(x_train[column])
x_test[column] = label_encoder.transform(x_test[column])
print("\n------------ Features (encoded) ------------ \n")
print("\nx_train :")
print(x_train.head())
print("\n--- \n")
print("\nx_test :")
print(x_test.head())
------------ Features (encoded) ------------
C: Users Eman Downloads Untitled.ipynb
Python 3.12 (ass1)
27°C
Search
ENG


ass1ml.py
Version control
Current File
Project
main.py
ass1ml.py
ass1mledited.py
Lab 4 (complete code).ipynb
Untitled.ipynb
---------- Features (encoded) ----------
x_train :
Loan_ID Gender Married Dependents Education Income \
336 LP002110 1 1 1 0 5250
579 LP002888 1 0 0 0 3182
521 LP002690 1 0 0 0 2500
186 LP001641 1 1 1 0 2178
31 LP001095 1 0 0 0 3167
Coapplicant_Income Loan_Tenor Credit_History Property_Area
336 688.0 144.0 1.0 0
579 2917.0 144.0 1.0 2
521 0.0 144.0 1.0 1
186 0.0 120.0 0.0 0
31 0.0 144.0 1.0 2
x_test :
Loan_ID Gender Married Dependents Education Income \
366 LP002187 1 0 0 0 2500
595 LP002940 1 0 0 1 3833
527 LP002706 1 1 1 1 5285
184 LP001639 0 1 0 0 3625
598 LP002945 1 1 0 0 9963
Coapplicant_Income Loan_Tenor Credit_History Property_Area
366 0.0 192.0 1.0 1
595 0.0 144.0 1.0 0
527 1430.0 144.0 0.0 1
184 0.0 144.0 1.0 1
598 0.0 144.0 1.0 0
C: Users Eman Downloads Untitled.ipynb
Python 3.12 (ass1)
Search
ENG


ass1ml.py
Version control
Current File
Project
main.py
ass1ml.py
ass1mledited.py
Lab 4 (complete code).ipynb
Untitled.ipynb
579 LP002888 1 0 0 0 3182
521 LP002690 1 0 0 0 2500
186 LP001641 1 1 1 0 2178
31 LP001095 1 0 0 0 3167
Coapplicant_Income Loan_Tenor Credit_History Property_Area
336 688.0 144.0 1.0 0
579 2917.0 144.0 1.0 2
521 0.0 144.0 1.0 1
186 0.0 120.0 0.0 0
31 0.0 144.0 1.0 2
x_test :
Loan_ID Gender Married Dependents Education Income \
366 LP002187 1 0 0 0 2500
595 LP002940 1 0 0 1 3833
527 LP002706 1 1 1 1 5285
184 LP001639 0 1 0 0 3625
598 LP002945 1 1 0 0 9963
Coapplicant_Income Loan_Tenor Credit_History Property_Area
366 0.0 192.0 1.0 1
595 0.0 144.0 1.0 0
527 1430.0 144.0 0.0 1
184 0.0 144.0 1.0 1
598 0.0 144.0 1.0 0
# v) Encode categorical targets
encoder = LabelEncoder()
C: Users Eman Downloads Untitled.ipynb
Python 3.12 (ass1)
Search
ENG

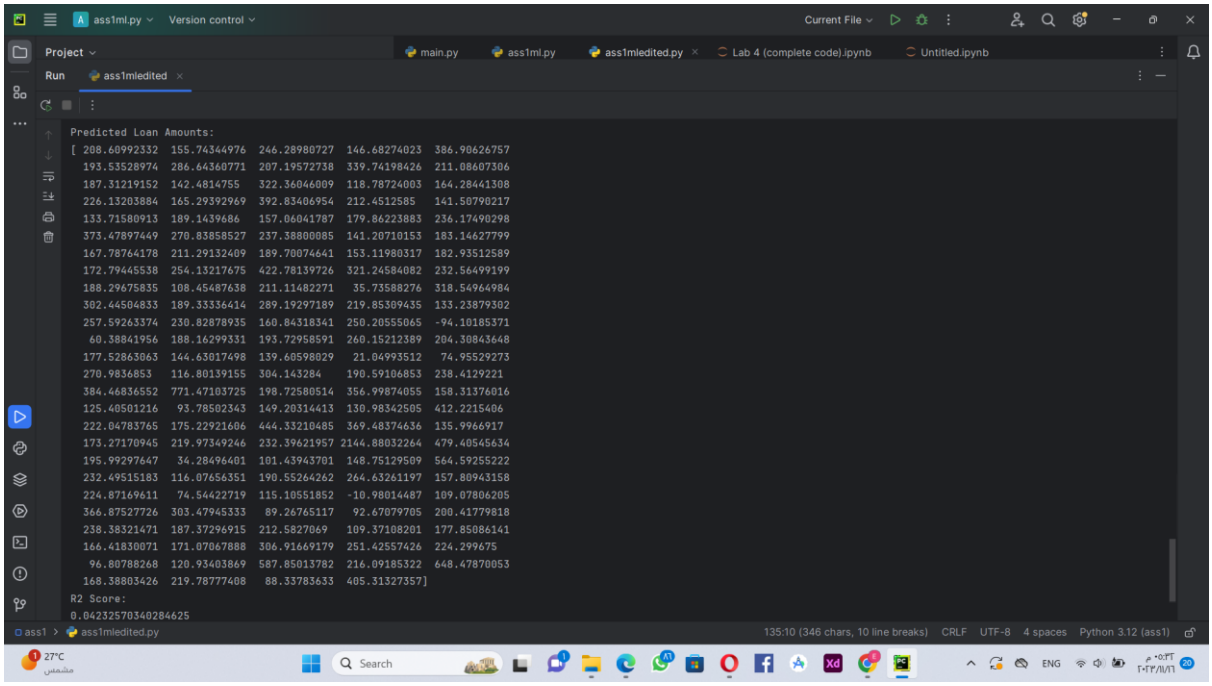

ass1ml.py
Version control
Current File
Project
main.py
ass1ml.py
ass1mledited.py
Lab 4 (complete code).ipynb
Untitled.ipynb
527 1430.0 144.0 0.0 1
184 0.0 144.0 1.0 1
598 0.0 144.0 1.0 0
[148]
# v) Encode categorical targets
encoder = LabelEncoder()
y_train['Loan_Status'] = encoder.fit_transform(y_train['Loan_Status'])
y_test['Loan_Status'] = encoder.fit_transform(y_test['Loan_Status'])
print(y_train.head())
print(y_test.head())
Max_Loan_Amount Loan_Status
336 229.28 1
579 237.39 1
521 56.00 1
186 21.48 0
31 89.62 0
Max_Loan_Amount Loan_Status
366 98.00 0
595 123.18 1
527 268.44 1
184 112.70 1
598 432.14 1
[72]
# vi) Scale numerical features to the range [0, 1]
scaler = MinMaxScaler()
C: Users Eman Downloads Untitled.ipynb
Python 3.12 (ass1)
Search
ENG


ass1ml.py
Version control
Current File
Project
main.py
ass1ml.py
ass1mledited.py
Lab 4 (complete code).ipynb
Untitled.ipynb
# vi) Scale numerical features to the range [0, 1]
scaler = MinMaxScaler()
numerical_features = features.select_dtypes(include=['float64', 'int64']).columns
for i in numerical_features:
x_train[i] = scaler.fit_transform(x_train[[i]])
x_test[i] = scaler.transform(x_test[[i]])
print("\n---------- Numerical features (Scaled) ----------\n")
print("x_train :")
print(x_train.head())
print("\n----------\n")
print("x_test :")
print(x_test.head())
---------- Numerical features (Scaled) ----------
x_train :
Loan_ID Gender Married Dependents Education Income \
336 LP002110 1 1 1 0 0.127983
579 LP002888 1 0 0 0 0.076087
521 LP002690 1 0 0 0 0.058973
186 LP001641 1 1 1 0 0.050892
31 LP001095 1 0 0 0 0.075711
Coapplicant_Income Loan_Tenor Credit_History Property_Area
336 0.03440 0.692308 1.0 0
579 0.14585 0.692308 1.0 2
521 0.00000 0.692308 1.0 1
186 0.00000 0.538462 0.0 0
31 0.00000 0.692308 1.0 2
C: Users Eman Downloads Untitled.ipynb
Python 3.12 (ass1)
Search
ENG

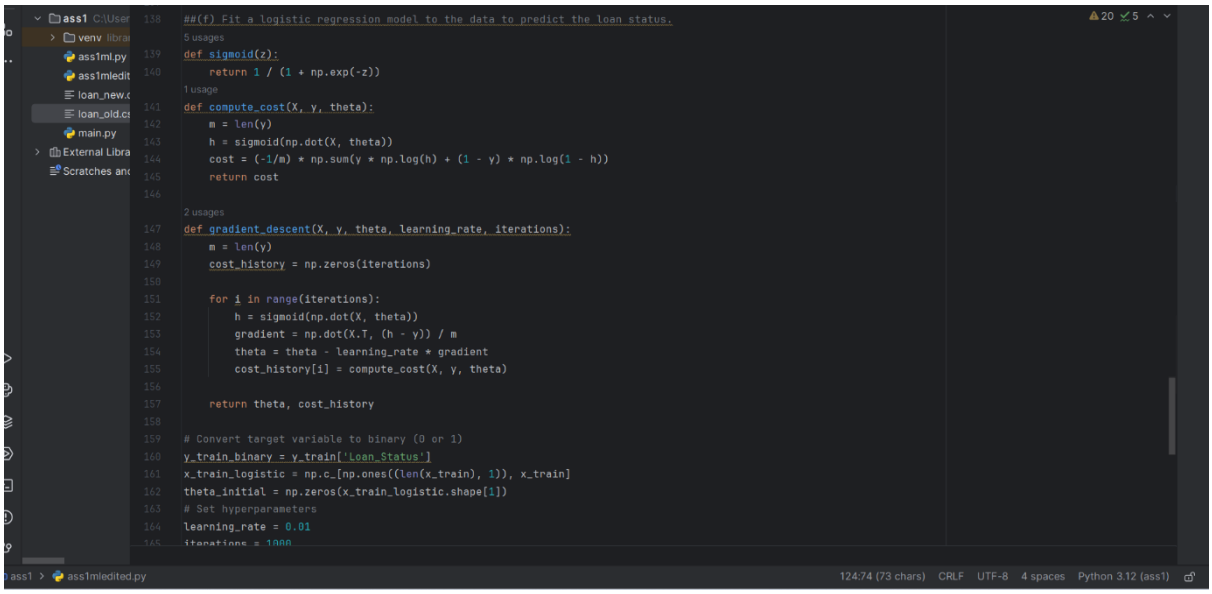


D-Linear Regression

```
x_train = x_train.drop(columns=['Loan_ID'])
x_test = x_test.drop(columns=['Loan_ID'])

# d)Fit a linear regression model to the data to predict the loan amount.
linear_model = LinearRegression()
linear_model.fit(x_train, y_train['Max_Loan_Amount'])
#predict the loan amount.
y_pred = linear_model.predict(x_test)
print("Predicted Loan Amounts:")
print(y_pred)

# e) Evaluate the linear regression model using sklearn's R2 score.
r2 = r2_score(y_test['Max_Loan_Amount'], y_pred)
print("R2 Score: ")
print(r2)
print("\n**************************************************\n")
```



F-Logistic regression



```python
def gradient_descent(X, y, theta, learning_rate, iterations):
    m = len(y)
    cost_history = np.zeros(iterations)

    for i in range(iterations):
        h = sigmoid(np.dot(X, theta))
        gradient = np.dot(X.T, (h - y)) / m
        theta = theta - learning_rate * gradient
        cost_history[i] = compute_cost(X, y, theta)

    return theta, cost_history

# Convert target variable to binary (0 or 1)
y_train_binary = y_train['Loan_Status']
x_train_logistic = np.c_[np.ones((len(x_train), 1)), x_train]
theta_initial = np.zeros(x_train_logistic.shape[1])
# Set hyperparameters
learning_rate = 0.01
iterations = 1000
theta_final, cost_history = gradient_descent(x_train_logistic, y_train_binary, theta_initial, learning_rate, iterations)
x_test_logistic = np.c_[np.ones((len(x_test), 1)), x_test]
probabilities = sigmoid(np.dot(x_test_logistic, theta_final))
```

Accurecy of prediction

```
Accuracy of Logistic Regression (from scratch): 71.31782945736434%
```

Data analysis of new data

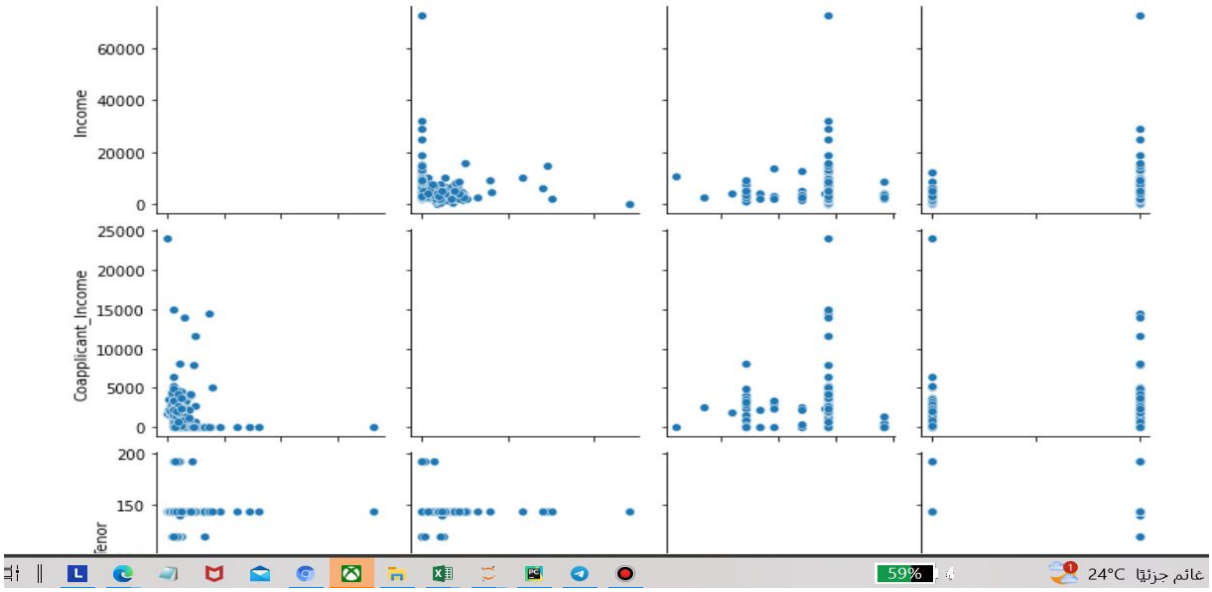
```python
print(missing_values[missing_values > 0])
```

```
Gender            11
Dependents        10
Loan_Tenor         7
Credit_History    29
dtype: int64
```

```
Data Types of new data:

Gender              : categorical
Married             : categorical
Dependents          : categorical
Education           : categorical
Income              : numerical
Coapplicant_Income  : numerical
Loan_Tenor          : numerical
Credit_History      : numerical
Property_Area       : categorical
```

```
New data Ranges & standerd deviations
Range of Income: 72529.0
Standard deviation of Income: 0.06541198928270776

Range of Coapplicant_Income: 24000.0
Standard deviation of Coapplicant_Income: 0.0973469898089432

Range of Loan_Tenor: 180.0
Standard deviation of Loan_Tenor: 0.12790746528146

Range of Credit_History: 1.0
Standard deviation of Credit_History: 0.3807099557532439
```

```
sns.pairplot(data2, diag_kind='kda')

plt.show()
```



Data processing of new data

```
------------ data2 (encoded) -------------

    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

   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

---------------------------------------------

----------  Numerical features (Scaled)  -----------

data2 :
    Loan_ID  Gender  Married  Dependents  Education    Income  \
0  LP001015       1        1           0          0  0.078865
1  LP001022       1        1           1          0  0.042411
2  LP001031       1        1           2          0  0.068938
4  LP001051       1        0           0          1  0.045168
5  LP001054       1        1           0          1  0.029850

   Coapplicant_Income  Loan_Tenor  Credit_History  Property_Area
0            0.000000    0.733333             1.0              2
1            0.062500    0.733333             1.0              2
2            0.075000    0.733333             1.0              2
4            0.000000    0.733333             1.0              2
5            0.142583    0.733333             1.0              2

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

```
------------ data2 (encoded) -------------

    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

   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

---------------------------------------------

----------  Numerical features (Scaled)  -----------

data2 :
    Loan_ID  Gender  Married  Dependents  Education    Income  \
0  LP001015       1        1           0          0  0.078865
1  LP001022       1        1           1          0  0.042411
2  LP001031       1        1           2          0  0.068938
4  LP001051       1        0           0          1  0.045168
5  LP001054       1        1           0          1  0.029850

   Coapplicant_Income  Loan_Tenor  Credit_History  Property_Area
0            0.000000    0.733333             1.0              2
1            0.062500    0.733333             1.0              2
2            0.075000    0.733333             1.0              2
4            0.000000    0.733333             1.0              2
5            0.142583    0.733333             1.0              2

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

Predictions of max loan amount in new data

```
In [118]: print(y_pred2)
[ 1.54232016e+02  1.38618269e+02  1.81319022e+02  9.50603875e+01
  1.72810751e+02  6.52031244e+01  1.06530491e+02  2.15559154e+02
  1.44790400e+02  9.14086390e+01  1.24768755e+02  2.68980101e+02
  1.18693373e+02  1.41101512e+02  2.02712078e+02  1.18486467e+02
  4.33568333e+02  2.09855679e+01  1.16267836e+02 -7.99955058e+01
  9.55093008e+01  2.64182870e+02  6.66728553e+02  2.88911759e+02
  2.12028956e+01  7.34183498e+01  1.79978833e+02  1.44976891e+02
  1.68142082e+02  1.36315204e+02  8.85119974e+01  1.54294403e+02
  1.60186450e+02  1.47619168e+02  1.56498327e+02  1.74149615e+02
  9.52065543e+01  1.30822363e+02  2.33115751e+02  9.56408163e+01
  1.30702433e+02  1.97378441e+02  1.31144456e+02  1.92162628e+02
  1.00528841e+01  1.43717691e+02  7.73029499e+01  1.14709909e+02
  7.08353755e+01  1.71607811e+02 -4.49729383e+00  1.35905763e+02
  1.87565154e+02  1.18172702e+02  1.01733025e+02  1.18144274e+02
  1.97327440e+02  1.25865409e+02  1.03178431e+02  1.95957838e+02
  2.22521800e+02  1.23889865e+02  1.11180947e+01  1.80518826e+02
  2.31423665e+02  1.94388315e+02  1.67104322e+02  1.64569510e+02
  2.16353304e+02  1.89046844e+02  1.27001255e+02  1.46326412e+03
  1.99951082e+02  2.14726992e+02 -9.44590530e+00  2.12739736e+02
  1.62699409e+02  9.63142290e+01  1.44570862e+02  1.29744507e+02
  3.05062046e+02  1.93900777e+02  1.73839794e+02  1.84843527e+02
  1.49278148e+02  2.15362919e+02  1.99995876e+02  2.23446146e+02
  1.26986109e+02  1.59010533e+02  1.15085494e+02 -4.01584168e+01
  1.37046012e+02  1.27310720e+02  1.33463590e+02  1.64354362e+02
  1.20193550e+02  9.07257602e+01  1.76776759e+02  2.56399065e+02
  7.12239763e+01  1.19366240e+02  1.39472775e+02  9.41654656e+01
  1.94953048e+02  1.39137606e+02  2.49254963e+02  2.60556420e+02
  1.27916658e+02  1.51324879e+02  1.82885855e+02 -3.94416206e+01
  1.40826825e+02  1.06042838e+02  1.50966647e+02  9.14918623e+01
  9.53375163e-01  1.25373333e+02  1.30295062e+02  1.62108729e+02
  1.22954061e+02  9.04236815e+01  1.68178820e+02  1.33639068e+01
  2.34528172e+02  9.83737626e+01  1.99351351e+02  1.54032333e+02
  1.48941693e+02  2.07482242e+02  1.34907658e+02  1.37803180e+02
  1.37800892e+02  1.37302471e+02  8.67501013e+01  2.31364913e+02
  1.86929678e+02  2.14626674e+02  2.01705791e+02  2.34000680e+02
  8.48405097e+01  1.42684373e+02  8.33574310e+01  6.90163452e+01
  9.05164091e+01  1.85438715e+02  1.28703889e+02  1.05856594e+02
  1.08031407e+02  1.27873801e+02  1.51087872e+02  2.55133773e+01
  1.32702604e+02  2.60906542e+02  1.70578962e+02  1.42691268e+02
  1.86149666e+02  2.08362108e+02  1.20865348e+02  2.18439903e+02
  1.56739874e+02  2.19104445e+02  3.04112663e+02  3.02199339e+02
  1.05521501e+01  1.13075601e+02  1.81934764e+02  1.32106608e+02
  3.10194364e+02  1.47619168e+02  1.26605113e+02  1.27815425e+02
  9.41007405e+01  1.23733081e+02  3.08187060e+02  1.00258139e+02
  1.60807790e+02  9.54050641e+01  1.60375182e+02  2.08319835e+02
  1.30822873e+02  1.06438148e+02  6.99009241e+01  1.98624165e+02
  1.41190171e+02  1.51778521e+02  1.09105622e+02 -3.68480879e+01
  2.60456319e+02  1.97498877e+02  1.68825558e+02  1.72501794e+02
  1.78308070e+02  8.23889622e+01  1.12976042e+02  7.35131943e+01
  1.38649707e+02  2.13643684e+02  1.59849838e+02  1.22303986e+02
  7.15214113e+02 -1.91631028e+01  1.73367178e+02  1.22160347e+02
  1.40694330e+02  2.00343271e+02  5.32274511e+02  1.15139263e+02
  2.06397496e+02  1.06272242e+02  1.52172016e+02  1.35901116e+02
  1.33840902e+02  6.60951136e+01  1.91034517e+02  1.20383172e+02
 -3.57362692e+01  1.86122441e+02  1.02858823e+02  1.41204806e+02
  1.44363347e+02  8.60615317e+01  1.14407385e+02  2.20939837e+02
  2.06186741e+02  1.57462049e+02  1.44644367e+02  4.16595871e+02
  1.30318069e+02  2.18583337e+02  1.95550500e+02  1.67055868e+02
  1.08352322e+02  1.17461508e+02  1.99644901e+02  5.22025423e+02
  1.52236818e+02  8.62171646e+01  1.29389134e+02  1.69628577e+02
  7.88504069e+00  1.19256598e+02  1.33152168e+02  1.46646556e+02
  2.14674211e+02  5.32705010e+02  2.10800783e+02  1.93424345e+02
  1.30840864e+02  3.08749812e+02  1.70110986e+02  1.72538418e+02
  1.01588442e+02  1.19985669e+02  1.39457962e+02  1.76445485e+02
  1.12188382e+02  1.00473537e+02  1.22794173e+02  1.86126807e+02
  1.52979467e+02  1.14540693e+02  1.89512605e+02  1.15556552e+02
  1.79429610e+02  2.42591972e+02  1.37374533e+02  1.97369045e+02
  1.40164652e+02  5.34409241e+01  8.13164184e+01  1.23290017e+02
  1.40315184e+02  1.66016813e+02  1.38286587e+02  1.09317498e+02
  5.45207111e+01  4.81105180e+02  1.53219703e+02  1.07037351e+02
  1.52928755e+02  2.13028107e+02  1.56693799e+02  2.23051363e+02
  1.70392229e+02  1.28385533e+02  1.26351299e+02  7.33726558e+01
  1.27923317e+02  9.19657656e+01  1.67902918e+02  9.20717158e+01
  2.30991463e+02 -2.69171522e+01  1.30857100e+02  1.74478579e+02
  1.96107381e+02  1.57613492e+02  8.70068016e+01  1.36857015e+02
  6.09623932e+01  2.12548230e+02  1.80302727e+02  2.49043875e+02
  6.85168367e-01  2.13877205e+02  1.64305433e+02  8.18284774e+01
  1.85546317e+02  1.43707685e+02  1.55820311e+02  1.42219867e+02
  2.10168901e+02  8.81270926e+01]
```

Predictions of loan status in new data

```
predictions_logistic = [1 if p >= 0.5 else 0 for p in probabilities]
```

```
: print (predictions_logistic)

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```