Green University of Bangladesh
Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: Spring, Year: 2025, B.Sc. in CSE (Day)

Lab Report:01

Code: CSE-412
Section: 222 D3
Course Title: Algorithom

Lab Experiment Name: LinearRegression

Student Details

Name	ID
Md.Jabed Mollah	222002167

Lab Date : 04/07/2025
Submission Date : 10/07/2025

Course Teacher's Name: **Md. Sabbir Hosen Mamu**

Lab Report Status

Marks: ……………………………………

Comments:...

Signature:..................
...
Date:..........................

Lab 01:

1. TITLE

Diabetes Prediction using Linear Regression

2. OBJECTIVES

1. **Handle Missing Values**: Replace zeros in key features (Glucose, BloodPressure, etc.) with median values to avoid bias.
2. **Feature Engineering**: Create new meaningful features (e.g., Glucose_BMI, Age_Insulin) to capture interactions between variables.
3. **Feature Scaling**: Normalize features using StandardScaler to ensure all features contribute equally to the model.

3.Ml Code:

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score, confusion_matrix,
precision_score, recall_score, f1_score
from sklearn.model_selection import train_test_split


df = pd.read_csv("https://raw.githubusercontent.com/mdjabedmollah/ml-
learning/refs/heads/main/diabetes.csv")


print("Original dataset info:")
print(df.info())
print("\nFirst 5 rows:")
print(df.head())



key_features = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
'BMI']
for feature in key_features:
    median_val = df[feature].median()
    df[feature] = df[feature].replace(0, median_val)

max_glucose = df['Glucose'].max()
df.loc[0, 'Glucose'] = max_glucose

min_age = df['Age'].min()
min_glucose = df['Glucose'].min()
df.loc[df['Age'] == min_age, 'Glucose'] = min_glucose

print("\nAfter preprocessing:")
print(df.describe())

X = df.drop('Outcome', axis=1)
y = df['Outcome']

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)
predictions = model.predict(X_test)
predictions_rounded = np.round(predictions).astype(int)

predictions_rounded = np.clip(predictions_rounded, 0, 1)

accuracy = accuracy_score(y_test, predictions_rounded)
conf_matrix = confusion_matrix(y_test, predictions_rounded)
precision = precision_score(y_test, predictions_rounded)
recall = recall_score(y_test, predictions_rounded)
f1 = f1_score(y_test, predictions_rounded)

print("\nModel Evaluation:")
print(f"Accuracy: {accuracy:.4f}")
print(f"Confusion Matrix:\n{conf_matrix}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1-Score: {f1:.4f}")

results = pd.DataFrame({'Actual': y_test, 'Predicted':
predictions_rounded})
print("\nSample predictions:")
print(results.head(10))
```

4.OUTPUT:

```
Original dataset info:
<class ’pandas.core.frame.DataFrame’>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
None

First 5 rows:
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0            6      148             72             35        0  33.6
1            1       85             66             29        0  26.6
2            8      183             64              0        0  23.3
3            1       89             66             23       94  28.1
4            0      137             40             35      168  43.1

   DiabetesPedigreeFunction  Age  Outcome
0                     0.627   50        1
1                     0.351   31        0
2                     0.672   32        1
3                     0.167   21        0
4                     2.288   33        1

After preprocessing:
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000   
mean      3.845052  116.294271      72.386719      27.334635   94.652344   
std       3.369578   36.797403      12.096642       9.229014  105.547598   
min       0.000000   44.000000      24.000000       7.000000   14.000000   
25%       1.000000   95.000000      64.000000      23.000000   30.500000   
50%       3.000000  115.000000      72.000000      23.000000   31.250000   
75%       6.000000  140.000000      80.000000      32.000000  127.250000   
max      17.000000  199.000000     122.000000      99.000000  846.000000   

              BMI  DiabetesPedigreeFunction         Age     Outcome  
count  768.000000                768.000000  768.000000  768.000000  
mean    32.450911                  0.471876   33.240885    0.348958  
std      6.875366                  0.331329   11.760232    0.476951  
min     18.200000                  0.078000   21.000000    0.000000  
25%     27.500000                  0.243750   24.000000    0.000000  
50%     32.000000                  0.372500   29.000000    0.000000  
75%     36.600000                  0.626250   41.000000    1.000000  
max     67.100000                  2.420000   81.000000    1.000000  

Model Evaluation:
Accuracy: 0.7662
Confusion Matrix:
[[83 16]
 [20 35]]
Precision: 0.6863
Recall: 0.6364
F1-Score: 0.6604

Sample predictions:
     Actual  Predicted
668       0          0
324       0          0
624       0          0
690       0          0
473       0          0
204       0          0
97        0          0
336       0          0
568       0          1
148       0          1
```

5. DISCUSSION

Doctors: Can use this tool to flag high-risk patients for further tests.
Patients: Early warnings may encourage lifestyle changes (diet/exercise) to prevent diabetes.
Hospitals: Reduces costs by focusing resources on those who need it most.

8.Reference:

https://github.com/mdjabedmollah/ml-learning/blob/main/lab1.ipynb Date and Time: 10-07-2025 05:53