Module IV

SMART MANUFACTURING AND INDUSTRY 4.0

MACHINE LEARNING LAB MANUAL

NAME:

REGNO.:

TABLE OF CONTENTS

Ex.No.	Name of the Experiment	DATE

Identification:

Design a Simple linear regression model using Computer dataset using MATLAB in the following three methods

1.Pseudo-inverse Method

2.SVD method

3.MATLAB Inbuilt function.

Software Required:

MATLAB 2024A.

Crucial points and Mathematics of Linear Regression Model:

- Produces or predicts the continuous Output
- Inputs are called Features/Predatory Variables
- Outputs are called Labels/Target Variables
- Single Feature-Simple Linear Regression
- Contrast to Classification which predicts the Discrete Output.

Given the model for simple linear regression:

$$h_\theta(X) = \theta_0 + \theta_1 X$$

The goal is to find the parameters θ_0 and θ_1 that minimize the cost function:

The cost function (also known as the Mean Squared Error, MSE) measures the difference between the predicted values ($h_\theta(X)$) and the actual values y:

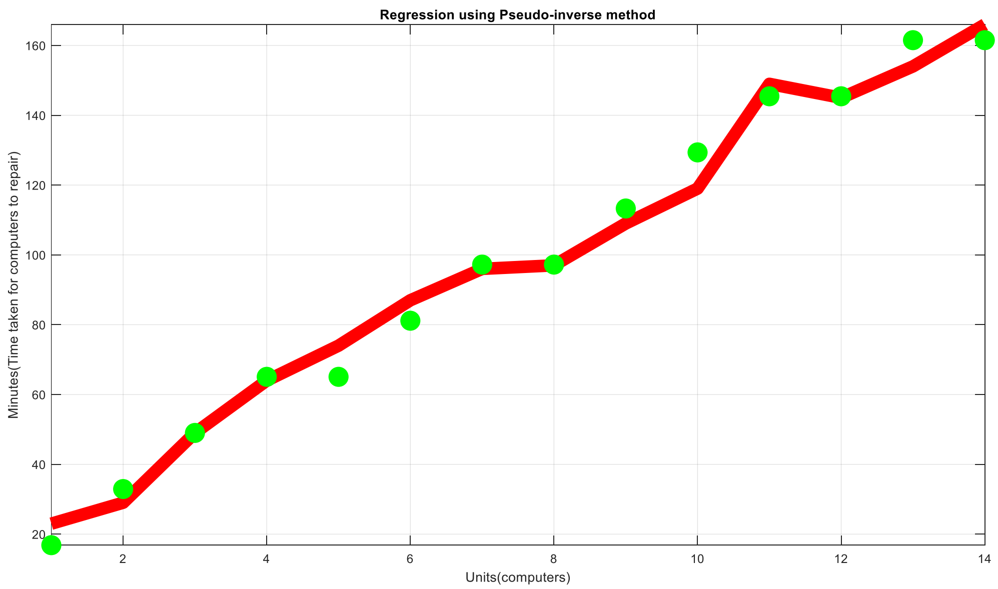
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(X_i) - y_i\right)^2$$

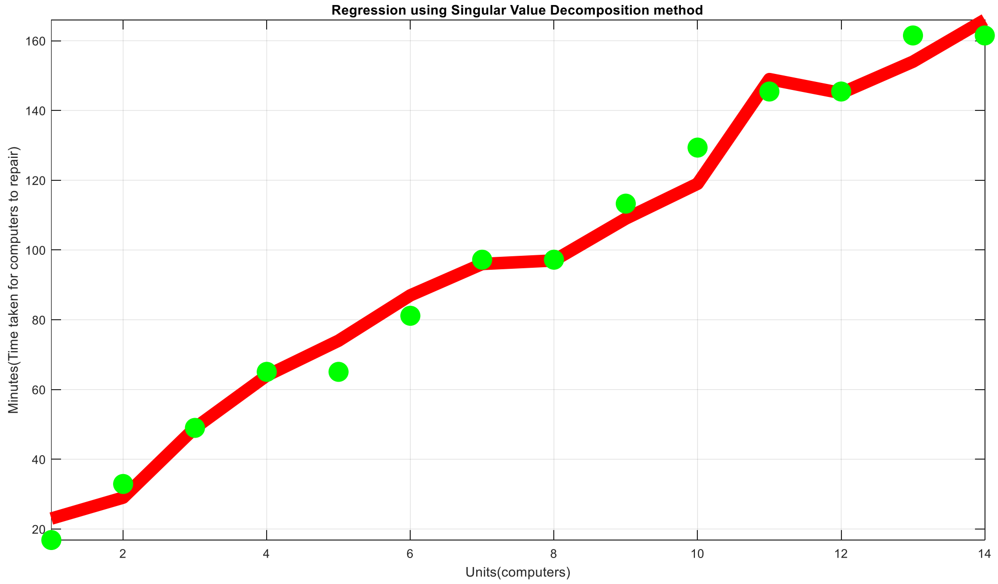
MATLAB CODE:

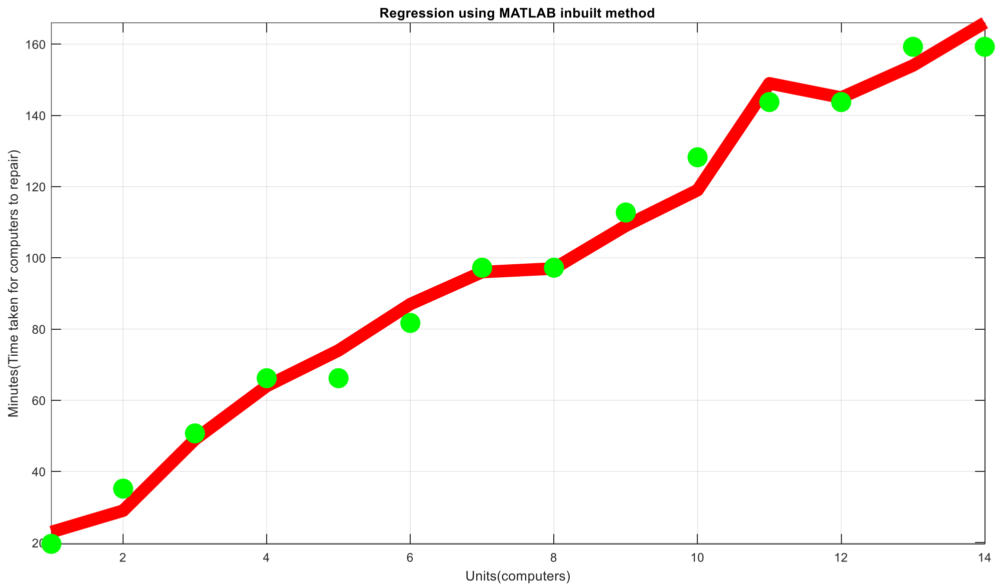
```
close all
Y=Minutes;
X=Units;
m1=X\Y; % Pseduo-inverse method
b0=mean(Y)-m1*mean(X); % calculate the Y-intercept
plot(Y,'r','LineWidth',10),hold on,plot(X*m1+b0,'go','Linewidth',10);
grid on
axis tight

% using SVD
[u,s,v]=svd(X,'econ');
m2=v*inv(s)*u'*Y;
b1=mean(Y)-m2*mean(X); % calculate the y-intercept
figure,plot(Y,'r','LineWidth',10),hold on,plot(X*m2+b1,'go','Linewidth',10);
grid on
axis tight
%using inbuilt matlab fn
c=fitlm(X,Y,"linear");
```

```
y1=4.1617+X*15.509;
figure,plot(Y,'r','LineWidth',10),hold on,plot(y1,'go','Linewidth',10);
%prediction
c1=predict(c,[11;12;13;14;16;18]);
c1=floor(c1); % Minutes data(Output)
c2=[11;12;13;14;16;18]; % Units data(Input)
% Finding weigths for new dataset
m3=c2\c1;
% using SVD finding new weigths
[u1,s1,v1]=svd(c2,'econ');
m4=v1*inv(s1)*u1'*c1;
%Matlab in-built function
c3=fitlm(c2,c1,'linear');
```

Responses:




Regression using Singular Value Decomposition method
Minutes(Time taken for computers to repair)
Units(computers)
20
40
60
80
100
120
140
160
2
4
6
8
10
12
14


Regression using MATLAB inbuilt method
Minutes(Time taken for computers to repair)
Units(computers)
20
40
60
80
100
120
140
160
2
4
6
8
10
12
14

Results:

Ex.No.	Name of the Experiment	DATE

Identification:

Design a Multiple linear regression model using Hald dataset using MATLAB in the following three methods

1.Pseudo-inverse Method

2.SVD method

3.MATLAB Inbuilt function.

Software Required:

MATLAB 2024A.

Crucial points and Mathematics of Multiple Linear Regression Model:

- Produces or predicts the continuous Output
- Inputs are called Features/Predatory Variables
- Outputs are called Labels/Target Variables
- It has Multiple features, unlike the simple linear regression.

Given the mathematical model

$$Y = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_3 + \theta_4 X_4 + \cdots$$

and the Cost functions are the measure of Y-predicted output and $\hat{Y}$- Actual Ouput is mentioned as,

$$J(\theta_0, \theta_1, \theta_2, ..) = \frac{1}{2*m} \sum (Y - \hat{Y})$$

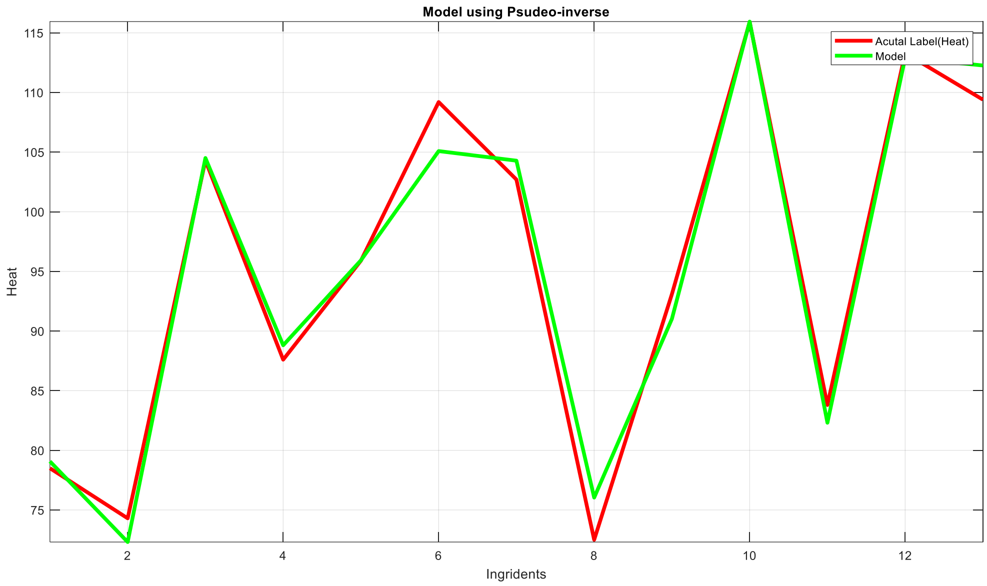
MATLAB CODE:

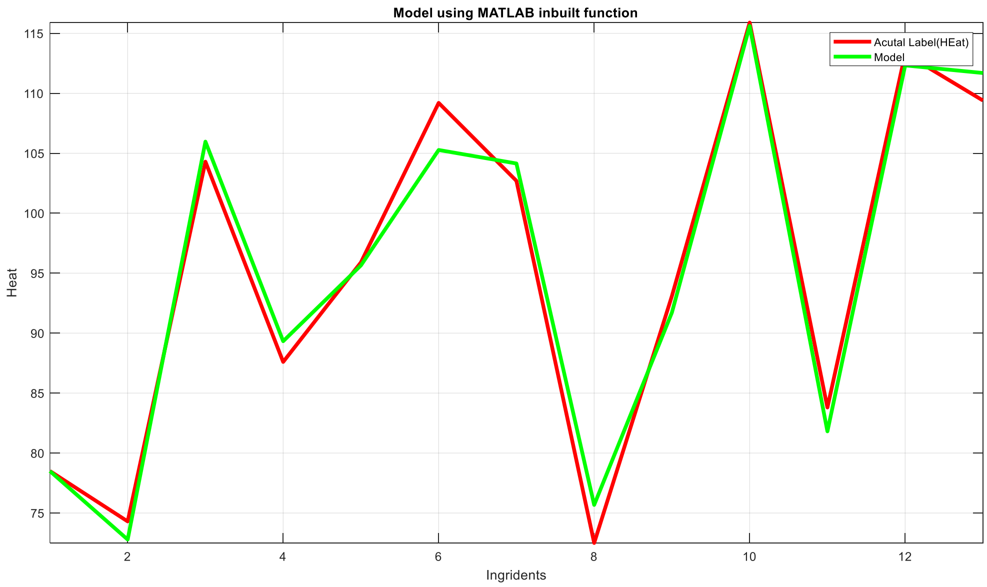
```
% multiple linear regression
% type load hald in command window
close all
clear all
load hald
x=ingredients;
y=heat;
%y=m*x----> Calulate the slope mamually
% First method is Pseudo-inverse
m1=x\y;
x1 = [ones(length(x),1) x];
% find the bias
b0=mean(y)-m1(1,1)*mean(x(:,1))-m1(2,1)*mean(x(:,2))-m1(3,1)*mean(x(:,3))-
m1(4,1)*mean(x(:,4));
y2=b0+m1(1,1)*x1(:,2)+m1(2,1)*x1(:,3)+m1(3,1)*x1(:,4)+m1(4,1)*x1(:,5);
plot(y,'r','LineWidth',3),hold on,plot(y2,'g',"LineWidth",3);
axis tight
legend(["Acutal Label(Heat)" "Model"])
title("Model using Psudeo-inverse")
xlabel("Ingridents")
ylabel("Heat")
% calculate the slope using matlab inbuilt function
c=fitlm(x,y);
y1=62.405+1.5511*x(:,1)+0.51017*x(:,2)+0.10191*x(:,3)-0.14406*x(:,4);
figure,plot(y,'r','LineWidth',3),hold on,plot(y1,'g',"LineWidth",3);
legend(["Acutal Label(HEat)" "Model"])
```

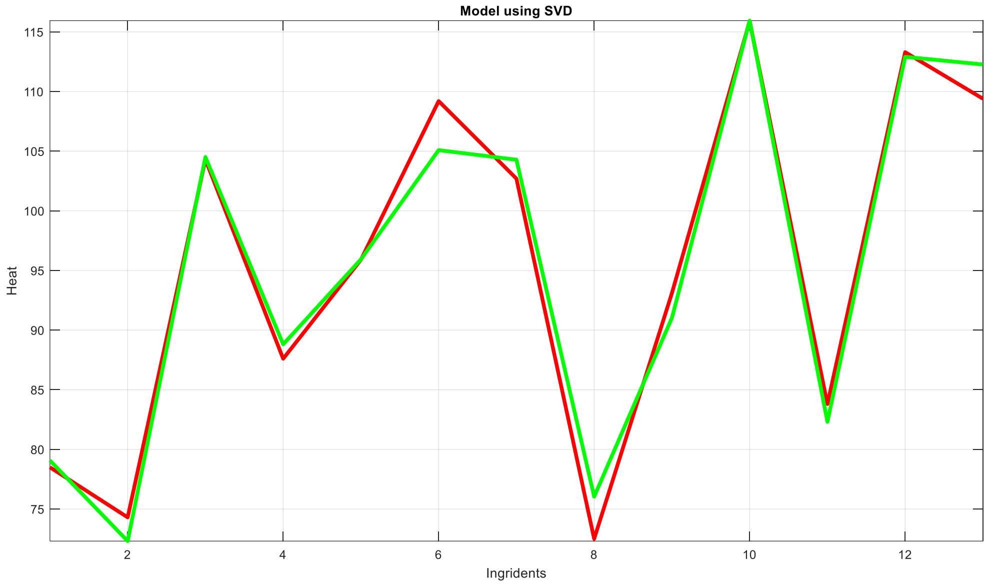
```
title("Model using MATLAB inbuilt function")
axis tight
xlabel("Ingridents")
ylabel("Heat")
% calculate the slope using SVD --> Singular value decomposition
[u,s,v]=svd(x,'econ');
m2=v*inv(s)*u'*y;
b1=mean(y)-m2(1,1)*mean(x(:,1))-m2(2,1)*mean(x(:,2))-m2(3,1)*mean(x(:,3))-
m2(4,1)*mean(x(:,4));
y3=b1+m2(1,1)*x1(:,2)+m2(2,1)*x1(:,3)+m2(3,1)*x1(:,4)+m2(4,1)*x1(:,5);
figure,plot(y,'r','LineWidth',3),hold on,plot(y2,'g',"LineWidth",3);
title("Model using SVD")
axis tight
xlabel("Ingridents")
ylabel("Heat")
% errors between different models
m_In_error=norm(y-y1,inf); % MATLAB inbuilt function error
p_err=norm(y-y2,inf);
svd_err=norm(y-y3,inf);
svd_err1=norm(y-y3,2);
svd_err2=norm(y-y3,1);
c1=[21 31 34 1;56 33 89 2;51 41 19 3;108 21 39 53];
p1=predict(c,c1');
c2=[[21 31 34 1;56 33 89 2;51 41 19 3;108 21 39 53] p1];
m3=c1\p1;
y4=x1([1:4],1)+c1(:,1)*m3(1,1)+c1(:,2)*m3(2,1)+c1(:,3)*m3(3,1)+c1(:,4)*m3(4,1
);
figure,plot(p1,'r','LineWidth',3),hold on,plot(y4,'g',"LineWidth",3);
```

```
legend(["Actual output" "Estimated output"])
pred_error=norm(p1-y4,2);
```

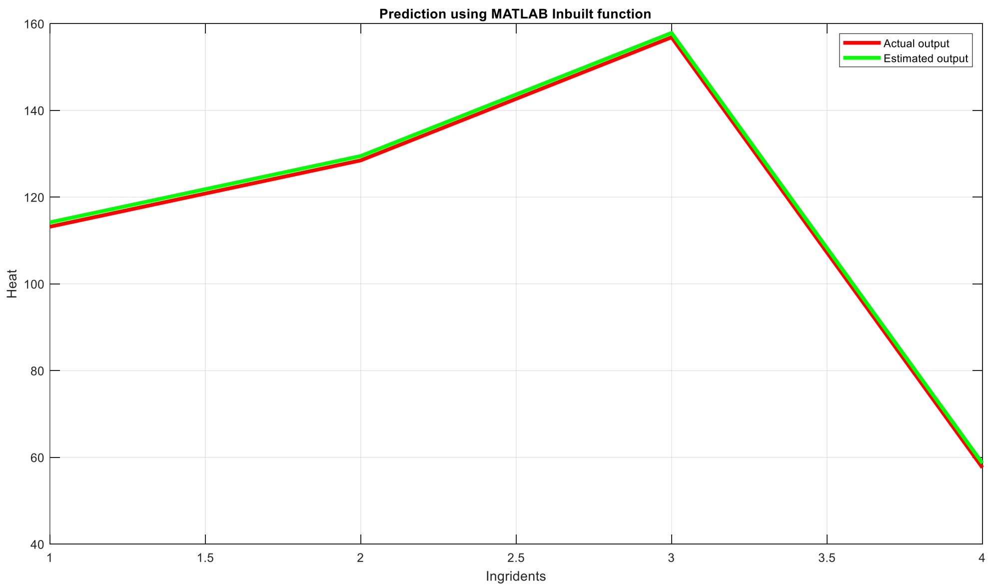
RESPONSES:







Model using SVD
Heat
Ingridents
115
110
105
100
95
90
85
80
75
2
4
6
8
10
12



Prediction using MATLAB Inbuilt function
Actual output
Estimated output
Heat
Ingridents
160
140
120
100
80
60
40
1
1.5
2
2.5
3
3.5
4

RESULTS:

Ex.No.	Name of the Experiment	DATE

IDENTIFICATION:

Design a Simple Logistic Regression model using Coronary Heart dataset using MATLAB.

Software Required:

MATLAB 2024A.

Crucial points and Mathematics of Simple Logistic Regression

- It is the probability based ML technique
- Outcomes are binary values
- Activation function is Non-linear

$$sigmoid(z) = \frac{1}{1 + e^{-z}}$$

$$where, \quad z = \beta_0 + \beta_1.x_1 + \beta_2.x_2 + \beta_3.x_3 + ... + \beta_n.x_n$$

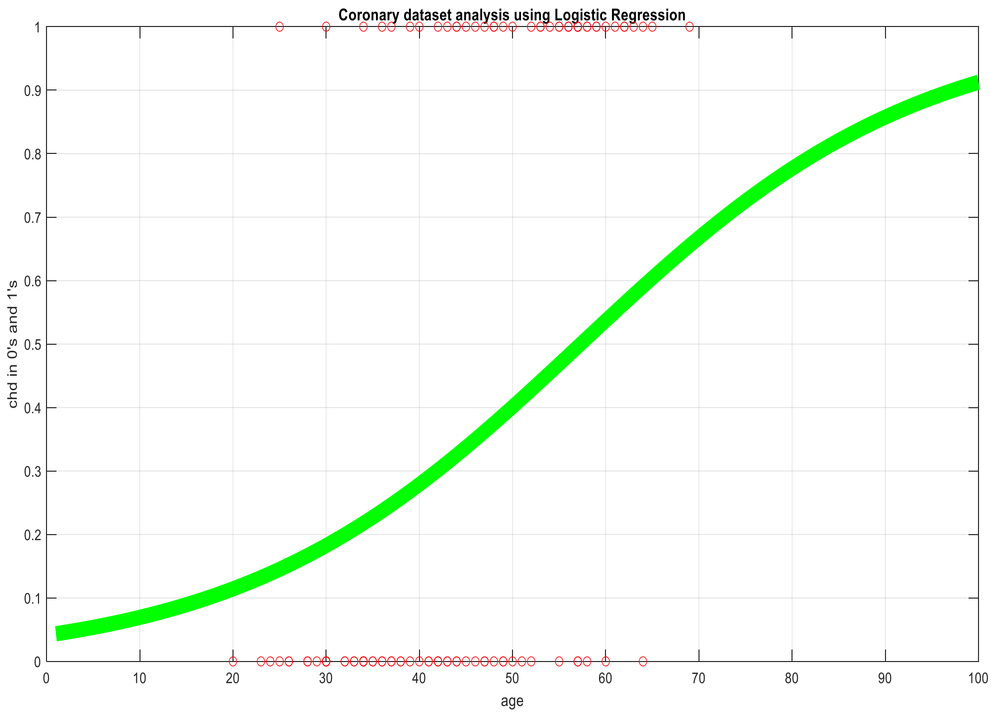
MATLAB CODE:

```
close all
X=age; % Features
Y=chd; % outputs
```

```
scatter(X,Y,'r')
figure,plot(X,Y,'ro');
hold on;

%Matlab inbuilt function
c4=fitglm(X,Y,'Distribution','binomial','Link','logit');

x_pred=linspace(min(X),max(X),100);
c5=predict(c4,x_pred');
plot(c5,'g',"LineWidth",10);
hold off
grid on
title("Coronary dataset analysis using Logistic Regression")
xlabel("age")
ylabel("chd in 0's and 1's")

aa=[];
for i=1:length(Y)
   if c5(i,1)<=0.5
      aa(i,1)=0;
   else
      aa(i,1)=1;
   end
end
% concatenate my predicted input and output
cc=[x_pred aa'];
```

```
bb=[];
for i=1:length(Y)
   if c5(i,1)<=0.5
      bb(i,1)=0;
   else
      bb(i,1)=1;
   end
end
```

RESPONSES:



RESULT:

Ex.No.	Name of the Experiment	DATE

IDENTIFICATION:

Design a Multiple Logistic Regression model using Coronary Heart dataset using MATLAB.

Software Required:

MATLAB 2024A.

Crucial points and Mathematics of Simple Logistic Regression

- It is the Probability based ML technique
- Outcomes are discrete values but not binary values
- Activation function is Non-linear.

$$sigmoid(z) = \frac{1}{1 + e^{-z}}$$

$$where, \quad z = \beta_0 + \beta_1.x_1 + \beta_2.x_2 + \beta_3.x_3 + \ldots + \beta_n.x_n$$

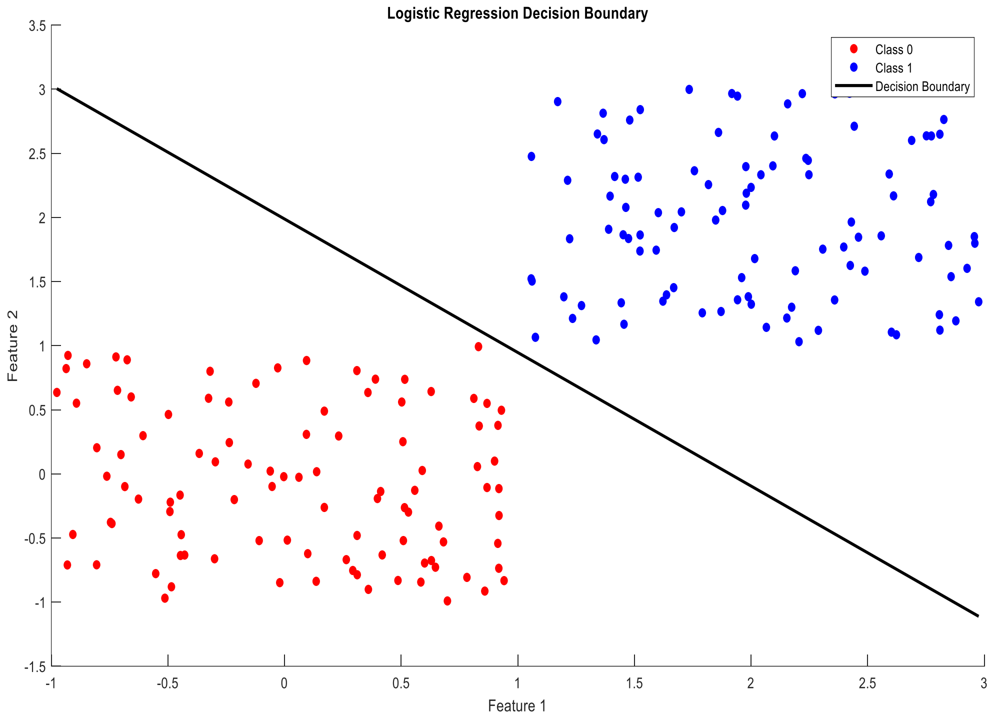
MATLAB CODE:

```
% Generate synthetic data (or load your dataset)
% Logistic Regression in MATLAB


% Generate synthetic data
```

```
rng(0); % For reproducibility
num_points = 100;
X = [rand(num_points, 2) * 2 - 1; rand(num_points, 2) * 2 + 1];
y = [zeros(num_points, 1); ones(num_points, 1)];

% Add bias term (intercept) to X
X = [ones(size(X, 1), 1), X];

% Sigmoid function
sigmoid = @(z) 1 ./ (1 + exp(-z));

% Cost function
costFunction = @(theta) (-1/length(y)) * sum(y .* log(sigmoid(X * theta)) + (1 - y) .* log(1 - sigmoid(X * theta)));

% Gradient descent parameters
alpha = 10;
num_iters = 1000;
theta = zeros(size(X, 2), 1);
m = length(y);

% Gradient descent loop
for iter = 1:num_iters
```

```
    gradient = (1/m) * X' * (sigmoid(X * theta) - y);
    theta = theta - alpha * gradient;
end

% Plot the data
figure;
hold on;

scatter(X(y==0, 2), X(y==0, 3), 'r', 'filled'); % Class 0
scatter(X(y==1, 2), X(y==1, 3), 'b', 'filled'); % Class 1

% Plot decision boundary
x_vals = linspace(min(X(:,2)), max(X(:,2)), 100);
y_vals = -(theta(1) + theta(2) * x_vals) / theta(3);
plot(x_vals, y_vals, 'k-', 'LineWidth', 2);

title('Logistic Regression Decision Boundary');
xlabel('Feature 1');
ylabel('Feature 2');
legend('Class 0', 'Class 1', 'Decision Boundary');
hold off;
```

Responses:



Results: