DATABASE MANAGEMENT SYSTEMS

MTE PROJECT

Submitted To-

Prof. SANJAY KUMAR

Submitted By

AYUSH SHARMA 2K19/SE/027

AYUSH KUMAR 2K19/SE/026

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our teacher **PROF. SANJAY KUMAR** who gave us the golden opportunity to do this wonderful project on the topic **Real Estate Price Prediction System using machine learning**, which also helped us in doing a lot of Research and we came to know about so many new things. We would also like to thank our university, Delhi Technological University for giving us this oppurtunity to explore and research in the field of software engineering.

Thanking you,
AYUSH KUMAR/ AYUSH SHARMA

CONTENTS

1.Title
2. Acknowledgement
3. Problem statement
4.Introduction
5. Proposed Approach
6. Data set used
7. Algorithm Used
8. Code
9. Result and Analysis
10. Conclusion
11. References

Problem Statement

- Suppose you are selling your house and you want to know what a good market price would be. One way to do this is to first collect information on recent houses sold and make a model of housing prices.
- A data file is provided containing a training set of housing prices in a city, based on a few parameters.
- The first parameter is the size of the house (in square feet), and the second parameter is the number of bedrooms. Based on these parameters, we are also given the price of the houses in the training data set.
- We will be creating an algorithm, using linear regression which will be trained on the given training data set, and then it would be able to predict the approximate suitable price of a house , given the parameters like size and number of bedrooms in the house.

INTRODUCTION

The program made by us will meet the following objectives:-

- Create a software to predict the suitable price of a house or a number of houses to be sold.
- Given : A training data set consisting of area of houses , number of bedrooms and the price of houses.
- Two data sets will be used in this project : one will be the training data set, and the other one will be the testing data set of the houses whose prices we want to be predicted.
- The output of the program will be the approximate expected prices of the houses.

PROPOSED APPROACH

- We will be using the multivariate linear regression algorithm to train our machine learning model. The input variables, x_1 and x_2 are the size of the house and number of bedrooms in the house respectively, and the output variable is the price of the house(y).
- The linear regression algorithm assumes a linear relationship between the input variables (x) and the output variables(y). This linear relationship is called the hypothesis function. We have to find out the best fitting linear function which suits our training data set.

- To do this, we use the batch gradient descent algorithm which gives us the best fitting parameters for our linear hypothesis function.

After obtaining the relationship between input parameters(x) and output parameters(y), we then plug in the values of the testing data (the houses whose prices we want) into the input variables, and the algorithm outputs the suitable price for these houses !

Software and Hardware Requirements

Reccomended hardware: Any PC system with minimum 2GB RAM

Reccomended Software: Any operating system with MATLAB/Octave Engine installed

DATA SET USED

The data set on which we trained our model consists of area of the house, number of bedrooms, and the price of that house for 47 houses in an area.Here are the first 10 entries of our used data set:-

Area of houses (sq. ft.)	No of bedrooms	Price of house(INR)
2104	3	399900
1600	3	329900
2400	3	369000
1416	2	232000
3000	4	539900
1985	4	299900
1534	3	314900
1427	3	198999
1380	3	212000
1494	3	242500

This is the whole data set plotted on a graph:-

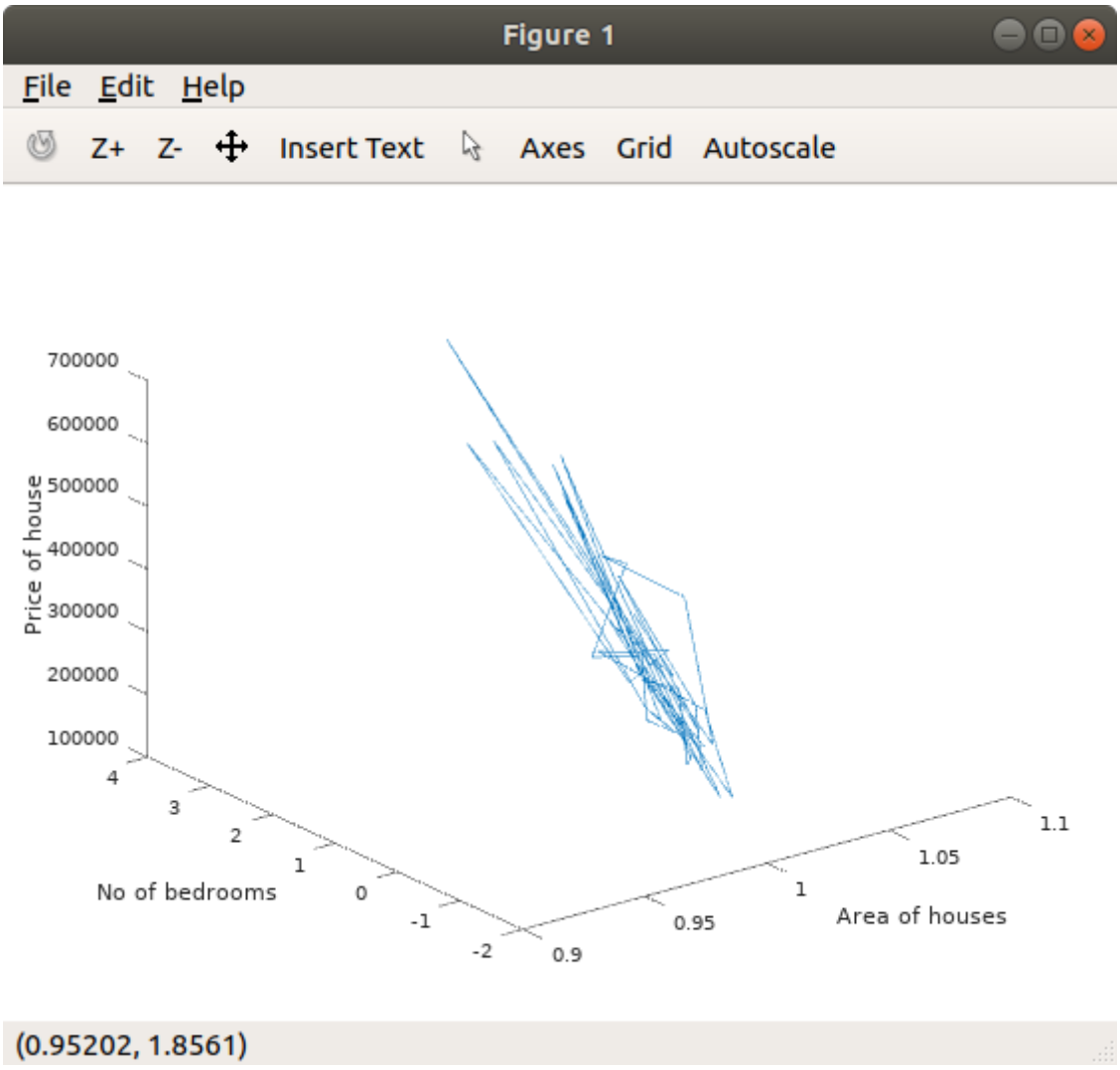


Fig1- Plot of the complete data set

ALGORITHM USED

- To solve this problem we have created a multivariate linear regression model. In this model, one single output variable is controlled by 2 or more input variables(in our case 2- the price of the house and number of bedrooms in the house to predict the output variable- the price of the house).

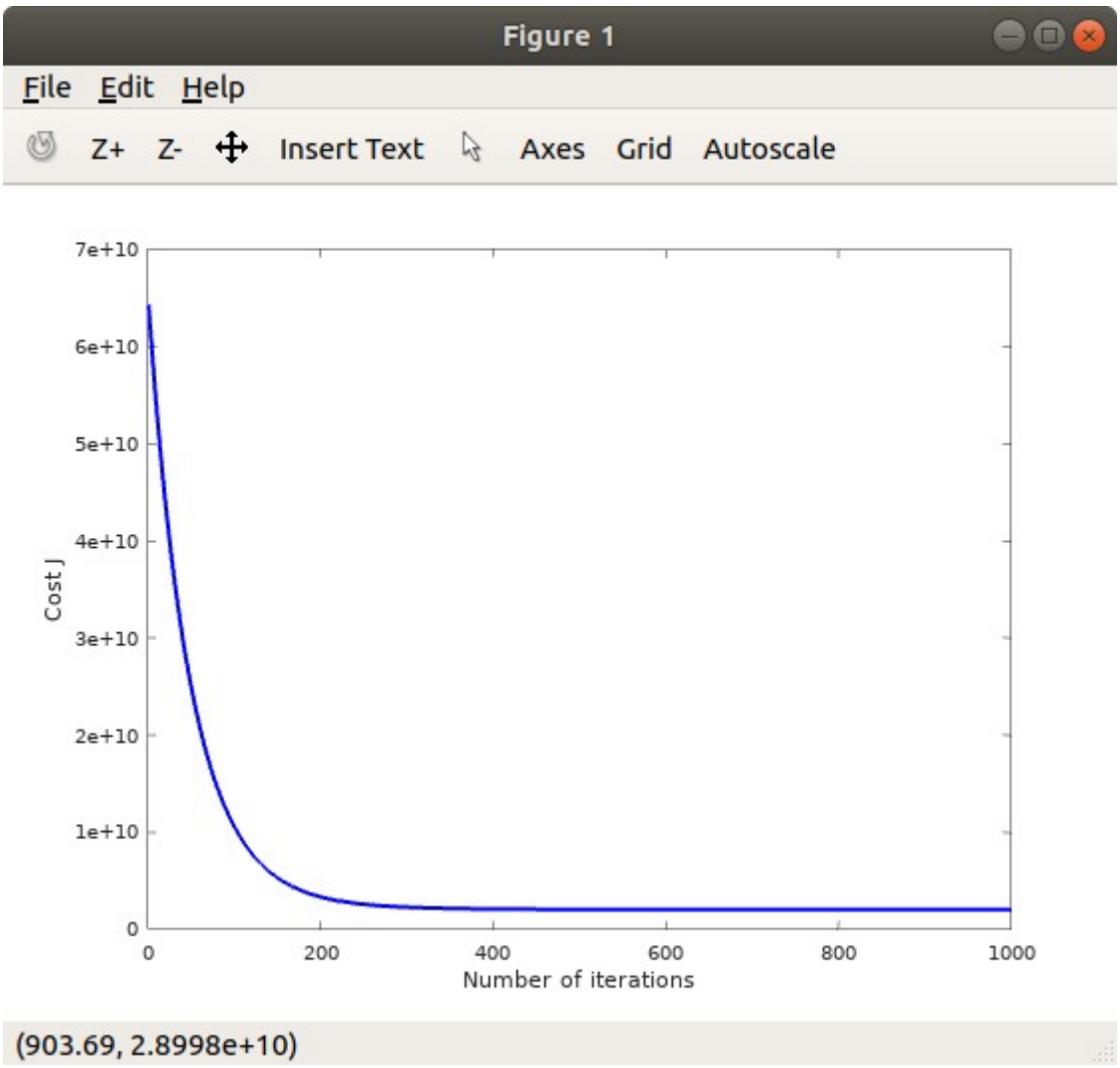
- Within this algorithm, we have to formulate a hypothesis function which is simply a linear polynomial in two variables, which can take two variables as input and predict the output variable. Our task is to find the parameters of this linear polynomial. The more accurate our parameters will be, the lesser will be the cost function of our hypothesis function.

- Thus, to find out accurate parameters of our hypothesis function, we try to minimise the cost function by using the **gradient descent algorithm**.

- Essentially, what this algorithm does is it iteratively updates the values of the parameters of the cost function, until convergence, ie,the cost function cannot be minimised to a greater extent.

- The rate at which the gradient descent algorithm updates the values of the parameters is determined by the learning rate. It is important to carefully choose a learning rate for our algorithm,because if the laerning rate is too high the algorithm may fail to converge, it may even diverge. On the other hand if the learning rate is too slow, the algorithm will take very large number of iterations to converge.

- Below is a graph which shows the proper convergence of a gradient descen algorithm, after choosing an aproriate learning rate:-



- We also need to do feature scaling on the data set. Feature scaling is done to make the mean of our data set roughly zero, and all the input variables have an almost similar range of values. This is needed to ensure a smooth working of the gradient descent algorithm.

CODE

We have writen the program for the above algorithm in the Octave programming language. The codes written by us can be seen below:-

main.m

```
clear ; close all; clc


fprintf('Loading data ...\n');


%% Load the data set into the system
data = load('dataset.txt');
X = data(:, 1:2);
y = data(:, 3);
m = length(y);


% Print the first 10 data points
fprintf('First 10 examples from the dataset: \n');
fprintf(' x = [%.0f %.0f], y = %.0f \n', [X(1:10,:) y(1:10,:)]');


fprintf('Program paused. Press enter to continue.\n');
pause;


% Feature scaling
fprintf('Normalizing Features ...\n');


[X mu sigma] = featureNormalize(X)



X = [ones(m, 1) X];



%Running gradient descent
fprintf('Running gradient descent ...\n');


% learning rate
```

```
alpha = 0.01;
num_iters = 1000;

theta = zeros(3, 1);
[theta, J_history] = gradientDescentMulti(X, y, theta, alpha, num_iters);



prices_x=X(:,1);
brs_x=X(:,2);



%Plot the data set
figure;

plot3(prices_x,brs_x,y);

xlabel('Area of houses');
ylabel('No of bedrooms');
zlabel('Price of house');



% Plot the convergence graph
figure;
plot(1:numel(J_history), J_history, '-b', 'LineWidth', 2);
%plot(1:50, J1(1:50), 'b');
hold on;
%plot(1:50, J2(1:50), 'r');
%plot(1:50, J3(1:50), 'k');
xlabel('Number of iterations');
ylabel('Cost J');



plot(prices_x,brs_x,y);
```

```
fprintf('Theta computed from gradient descent: \n');
fprintf(' %f \n', theta);
fprintf('\n');

area_house= (1650-mu(1))/sigma(1);
br_house= (3-mu(2))/sigma(2);
price = theta' * [1;area_house;br_house];

fprintf(['Predicted price of a 1650 sq-ft, 3 br house ' ...
        '(using gradient descent):\n $%f\n'], price);
```

featureNormalize.m

```
function [X_norm, mu, sigma] = featureNormalize(X)
%For feature scaling
mu = mean(X);
sigma = std(X);

t = ones(length(X), 1);
X_norm = (X - (t * mu)) ./ (t * sigma);

end
```

computeCost.m

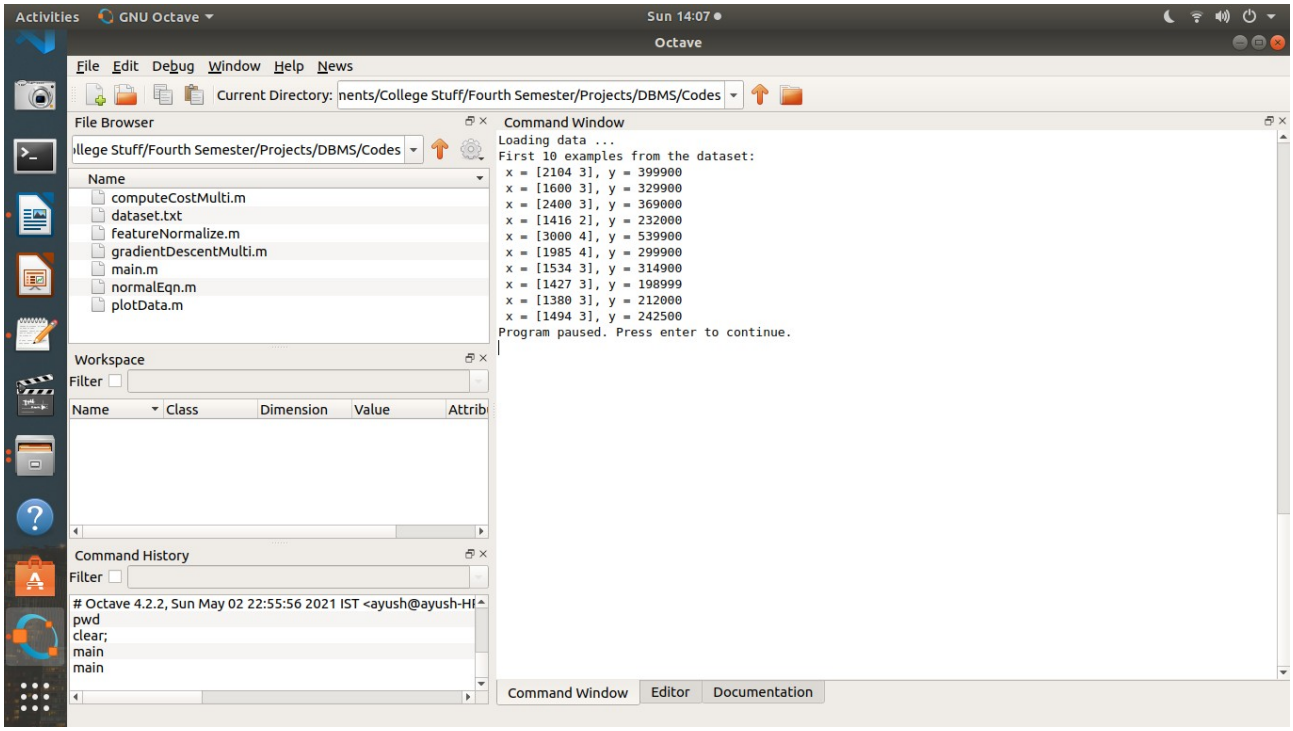
```
function J = computeCostMulti(X, y, theta)
% Cost Function

m = length(y);

J = (1/(2*m)) * (X * theta - y)' * (X * theta - y);

end
```

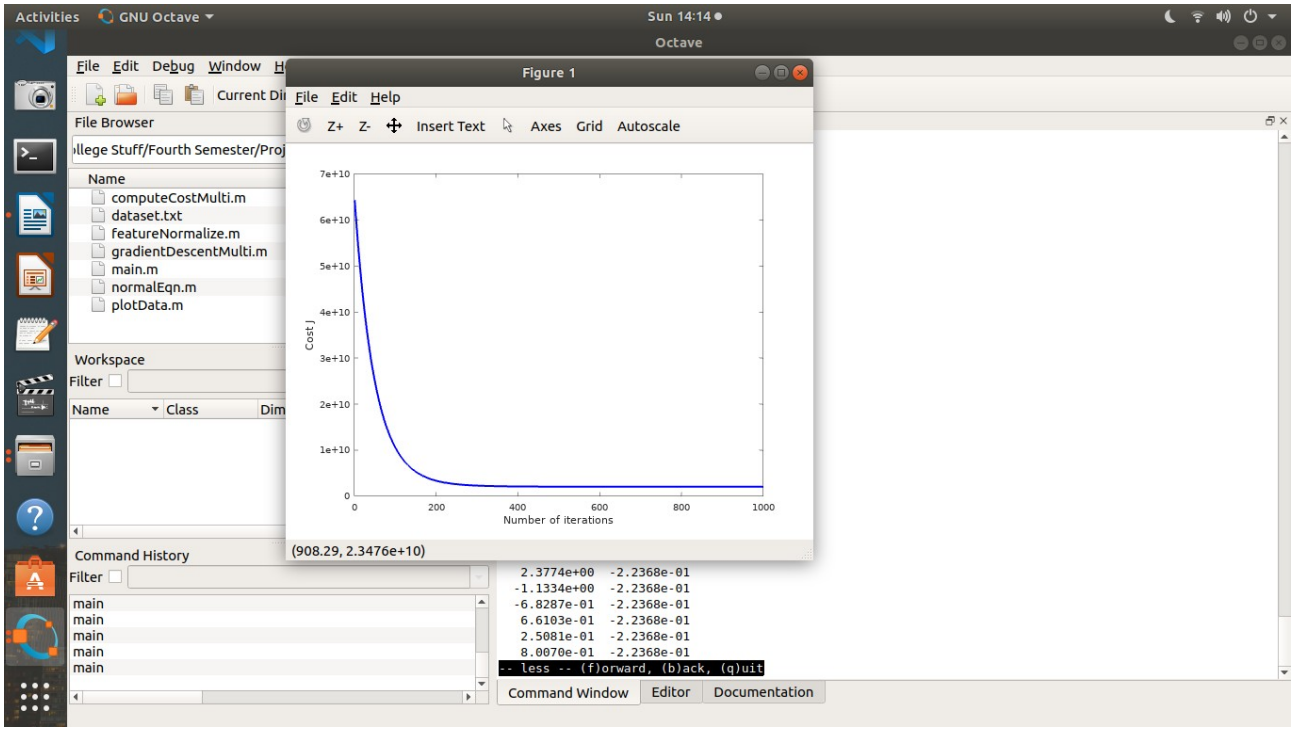
gradientDescent.m

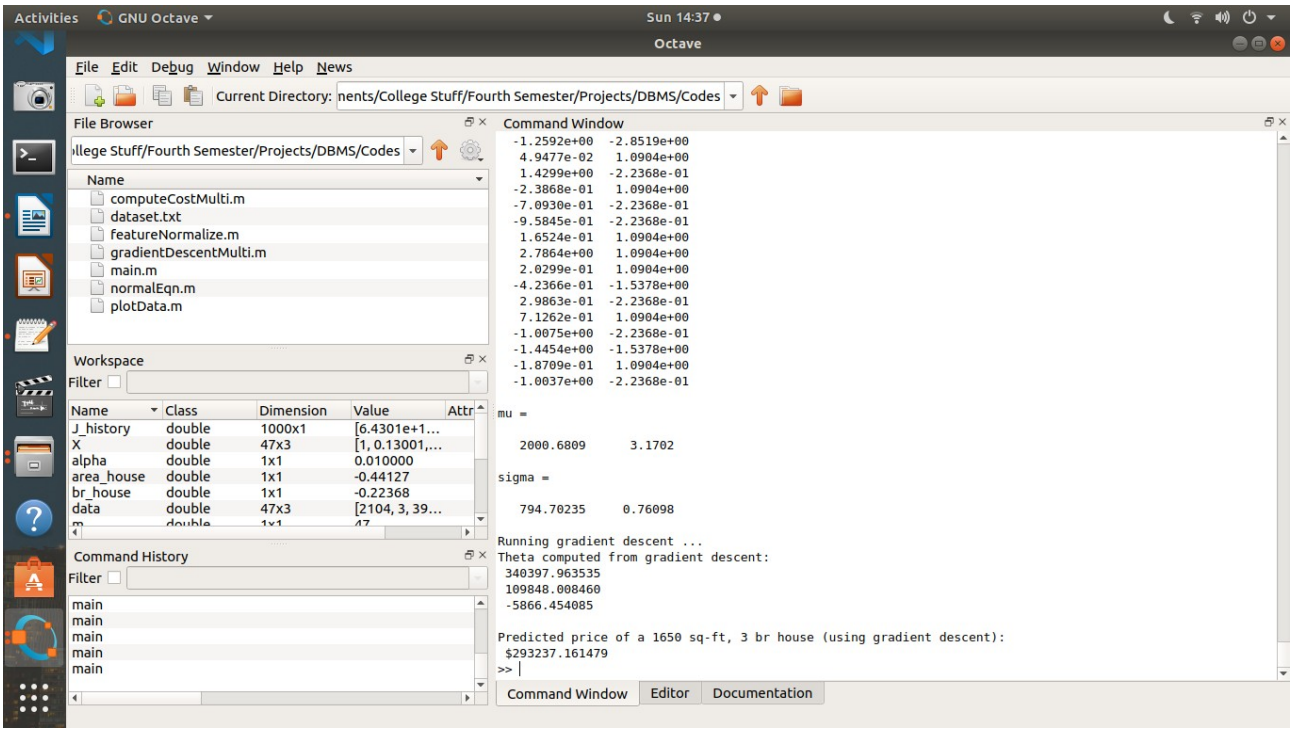
```
function [theta, J_history] = gradientDescentMulti(X, y, theta, alpha, num_iters)

m = length(y);
J_history = zeros(num_iters, 1);

for iter = 1:num_iters
    theta = theta - alpha * (1/m) * (((X*theta) - y)' * X)';
    J_history(iter) = computeCostMulti(X, y, theta);
end

end
```

RESULT AND ANALYSIS

The output and implementation of our algorithm can be seen here :-




Activities
GNU Octave
Sun 14:14
Octave
File Edit Debug Window
Figure 1
File Edit Help
Z+ Z- Insert Text Axes Grid Autoscale
Cost J
Number of iterations
(908.29, 2.3476e+10)
File Browser
computeCostMulti.m
dataset.txt
featureNormalize.m
gradientDescentMulti.m
main.m
normalEqn.m
plotData.m
Workspace
Filter
Name Class
Command History
main
main
main
main
main
2.3774e+00 -2.2368e-01
-1.1334e+00 -2.2368e-01
-6.8287e-01 -2.2368e-01
6.6103e-01 -2.2368e-01
2.5081e-01 -2.2368e-01
8.0070e-01 -2.2368e-01
-- less -- (f)orward, (b)ack, (q)uit
Command Window
Editor
Documentation


Activities
GNU Octave
Sun 14:37
Octave
File Edit Debug Window Help News
Current Directory: nents/College Stuff/Fourth Semester/Projects/DBMS/Codes
File Browser
llege Stuff/Fourth Semester/Projects/DBMS/Codes
computeCostMulti.m
dataset.txt
featureNormalize.m
gradientDescentMulti.m
main.m
normalEqn.m
plotData.m
Workspace
Filter
Name Class Dimension Value
J_history double 1000x1 [6.4301e+1...
X double 47x3 [1, 0.13001,...
alpha double 1x1 0.010000
area_house double 1x1 -0.44127
br_house double 1x1 -0.22368
data double 47x3 [2104, 3, 39...
Command History
main
main
main
main
main
Command Window
-1.2592e+00 -2.8519e+00
4.9477e-02 1.0904e+00
1.4299e+00 -2.2368e-01
-2.3868e-01 1.0904e+00
-7.0930e-01 -2.2368e-01
-9.5845e-01 -2.2368e-01
1.6524e-01 1.0904e+00
2.7864e+00 1.0904e+00
2.0299e-01 1.0904e+00
-4.2366e-01 -1.5378e+00
2.9863e-01 -2.2368e-01
7.1262e-01 1.0904e+00
-1.0075e+00 -2.2368e-01
-1.4454e+00 -1.5378e+00
-1.8709e-01 1.0904e+00
-1.0037e+00 -2.2368e-01
mu =
2000.6809 3.1702
sigma =
794.70235 0.76098
Running gradient descent ...
Theta computed from gradient descent:
340397.963535
109848.008460
-5866.454085
Predicted price of a 1650 sq-ft, 3 br house (using gradient descent):
$293237.161479
>>
Command Window
Editor
Documentation

CONCLUSION

- The parameters of the hypothesis function found out by our gradient descent algorithm are :- (340397.963, 109848.008, -5866.454)

- Based on these parameters, our program predicted the estimated price of a house which has 3 bedrooms and an area of 1650 sq. ft. The predicted price for this house as per our algorithm would be :- Rs.293237.161

- Hence, in conclusion we have created a real estate price prediction program using machine learning!

REFERENCES-

- https://towardsdatascience.com/linear-regression-using-gradient-descent-97a6c8700931
- https://www.hackerearth.com/practice/machine-learning/linear-regression/multivariate-linear-regression-1/tutorial/
- https://www.geeksforgeeks.org/ml-feature-scaling-part-2/
- https://www.geeksforgeeks.org/ml-linear-regression/