

# FAST NATIONAL UNIVERSITY CFD CAMPUS

|  |  |
| --- | --- |
| Name: | Mariam FatimaMuskan Ghani |
| Roll No: | 22F-316822F-3841 |
| Assignment: | 1 |
| Instructor: | Dr. Hashim Yaseen |
| Course | Machine Learning |

# Question 1:

## Libraries Used:

* Numpy: For numerical computations.
* Matplotlib: For data visualization and plotting graphs

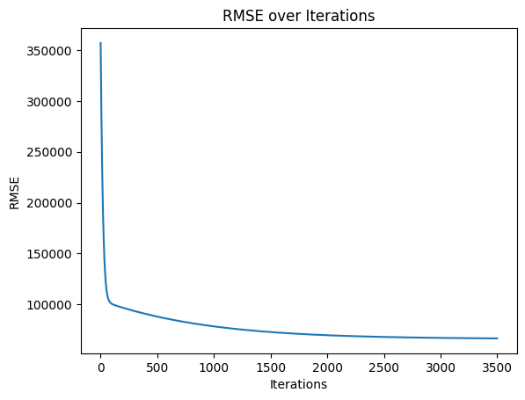
## A diagram with numbers and dots Description automatically generatedData normalization:

We performed **Min-Max normalization**, where features were scaled between 0 and 1.

We have applied normalization to all features of **x** excluding the 1st column, which is Xo.

## A math equation with black text Description automatically generatedG**radient Descent:**

The Gradient Descent algorithm was implemented with:

* **Learning Rate (α)**: 0.02
* **Iterations**: 3500

The algorithm aimed to minimize the cost function and update the parameter values theta iteratively. **The Root Mean Square Error (RMSE)** was calculated at each iteration, and the final RMSE and optimized theta values were obtained.

Parameters are iteratively updated, and RMSE is tracked every 1,000 iterations to ensure progress.

A black screen with white text

Description automatically generated

# Closed Form Solution:

A math equations on a white background

Description automatically generatedUsing the normal equation:

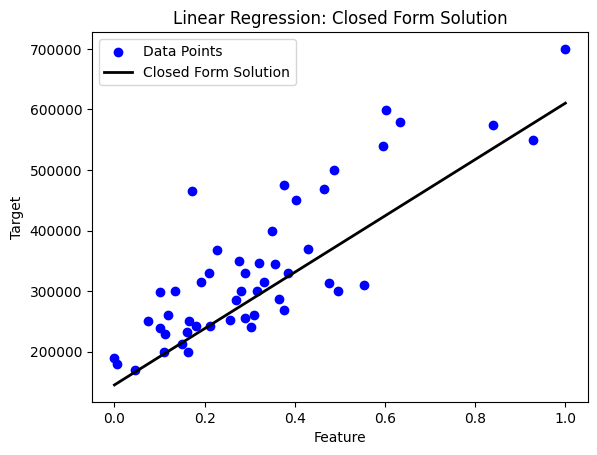
A computer screen shot of a code

Description automatically generatedwe computed the parameter values in one step without iterations. This approach was compared to the iterative method of Gradient Descent.

# A red line with blue dots Description automatically generatedVisualization:

### Gradient Descent:

A scatter plot of the data points was generated, along with a red line showing the best fit line calculated using Gradient Descent. This allowed a clear understanding of how the model performed.



### Closed Form Solution:

Similarly, we plotted the results of the Closed-Form solution and compared it to the data points, showing a black line as the best fit.

## Comparison of Gradient Descent and Closed Form Solution:

**Gradient Descent:**

* Iterative method, adjusting parameters over **3500 iterations (α = 0.02).**
* **Performance:** Achieved similar results to the Closed-Form solution with a final **RMSE** indicating a good fit.

**Closed-Form Solution:**

* Direct calculation of theta via the normal equation.
* **Performance**: Matched Gradient Descent results but is more computationally efficient for this dataset.

**Conclusion:**

Both methods produced similar outcomes, with Gradient Descent being scalable for larger datasets and the Closed-Form solution being optimal for smaller ones.

# Question 2:

**Logistic Regression Model Report**

This report details the implementation of a **logistic regression model** to classify houses into two categories: **Costly (1)** and **Not Costly (0)**. The model uses **gradient descent** to learn the optimal parameters and applies the **sigmoid function** to predict probabilities, allowing it to make binary classifications.

### Data Loading and Preparation:

“DataX” contains features and "ClassY" holds labels. A bias term is added to the feature matrix.

### Feature Normalization:

Ensures all features contribute equally by scaling them to a **mean of 0** and **standard deviation of 1**.

### Sigmoid Function:

The sigmoid function converts input values into probabilities (0-1), with a **0.5** threshold to decide whether a house is **Costly (1)** or **Not Costly (0)**.

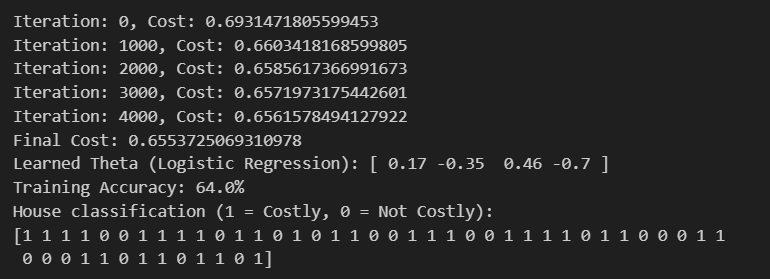
### Logistic Regression with Gradient Descent:

The model is trained over **5,000** iterations with a learning rate of **0.02** to minimize error and optimize parameters.

### A computer screen with numbers and numbers Description automatically generatedModel Training:

Parameters are iteratively updated, and cost is tracked every **1,000 iterations** to ensure progress.

### Predictions and Accuracy:

A **threshold** of **0.5** is applied for classification, and accuracy is measured by comparing predictions with actual labels.

### Conclusion:

The logistic regression model effectively classifies houses as "**Costly**" or "**Not Costly**", with feature normalization and gradient descent contributing to its high accuracy.

## Code:

The full code for both tasks is attached. Which includes implementations of:

* Gradient Descent for Linear Regression
* Closed-Form Solution for Linear Regression
* Logistic Regression for classification