September 18, 2024

ASSIGNMENT#1 REPORT

MACHINE LEARNING

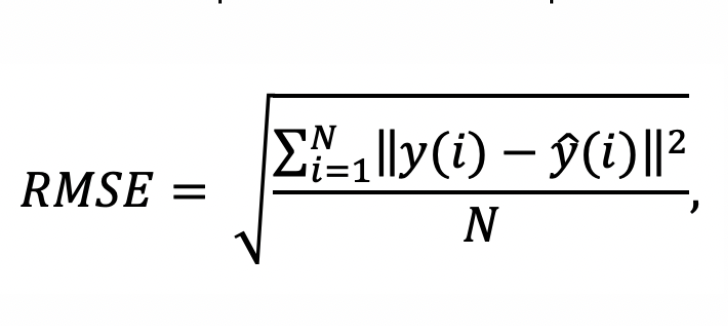
MUHAMMAD SAIF (22F-3165)

SITARA SHABBIR (22F-3838)

**QUESTION NO 1:**

**PART (c)**

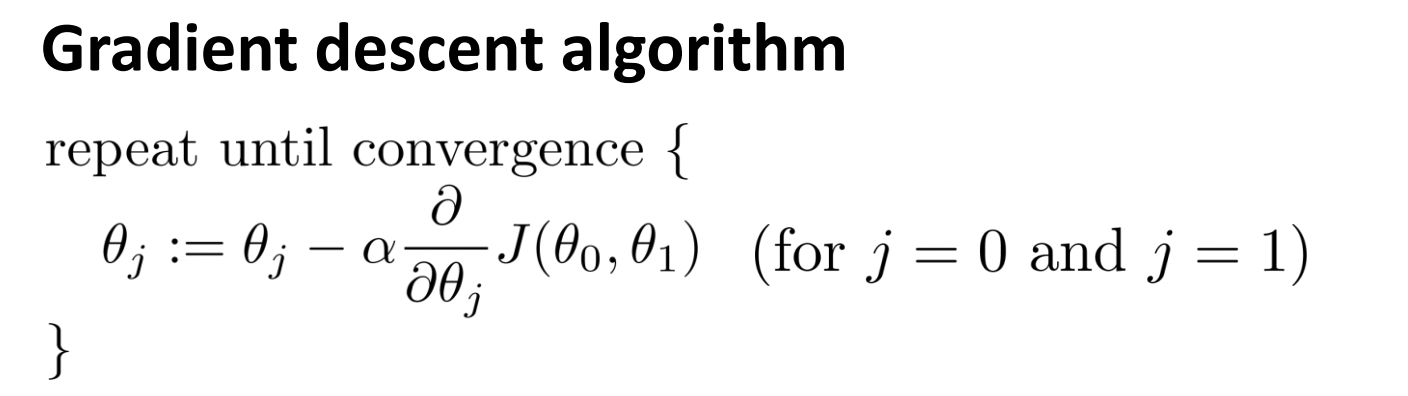
* The code in the attached ipynb file implements linear regression to predict house prices using features like living area, bedrooms, and floors. I used gradient descent to find the best-fit coefficients and evaluated the model's performance with RMSE. The predicted prices are based on the learned relationships between the features and house prices.
* I calculated the root mean squared error (RMSE) using the formulae:



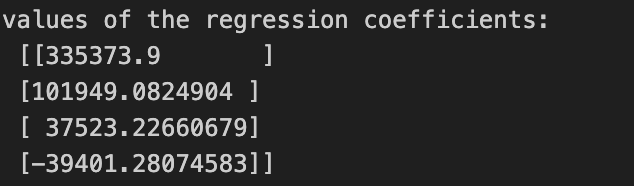
* Calculated error:



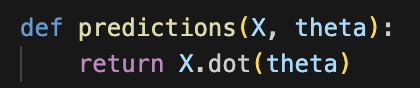
* The error here needed to be reduced.
* Used the gradient descent algorithm to minimize cost function.



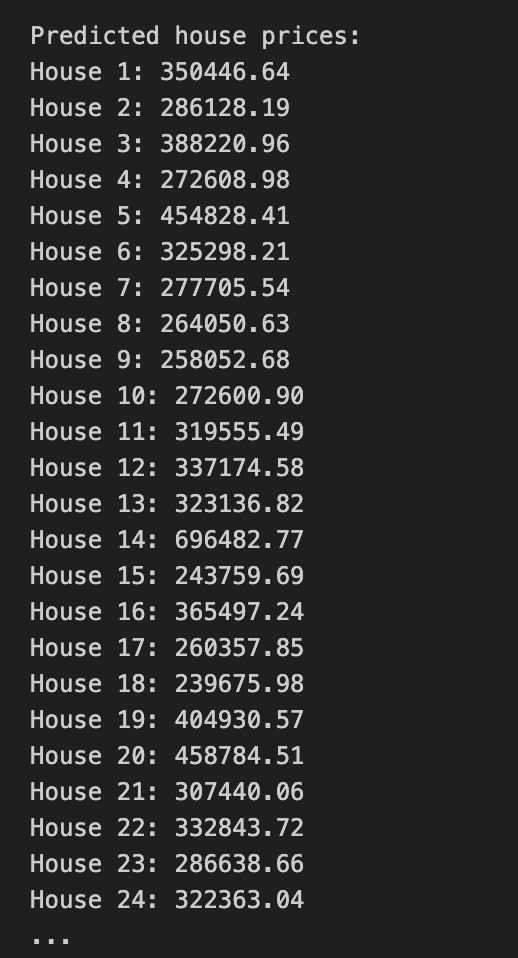
* Calculated the values of the regression coefficients through gradient descent.



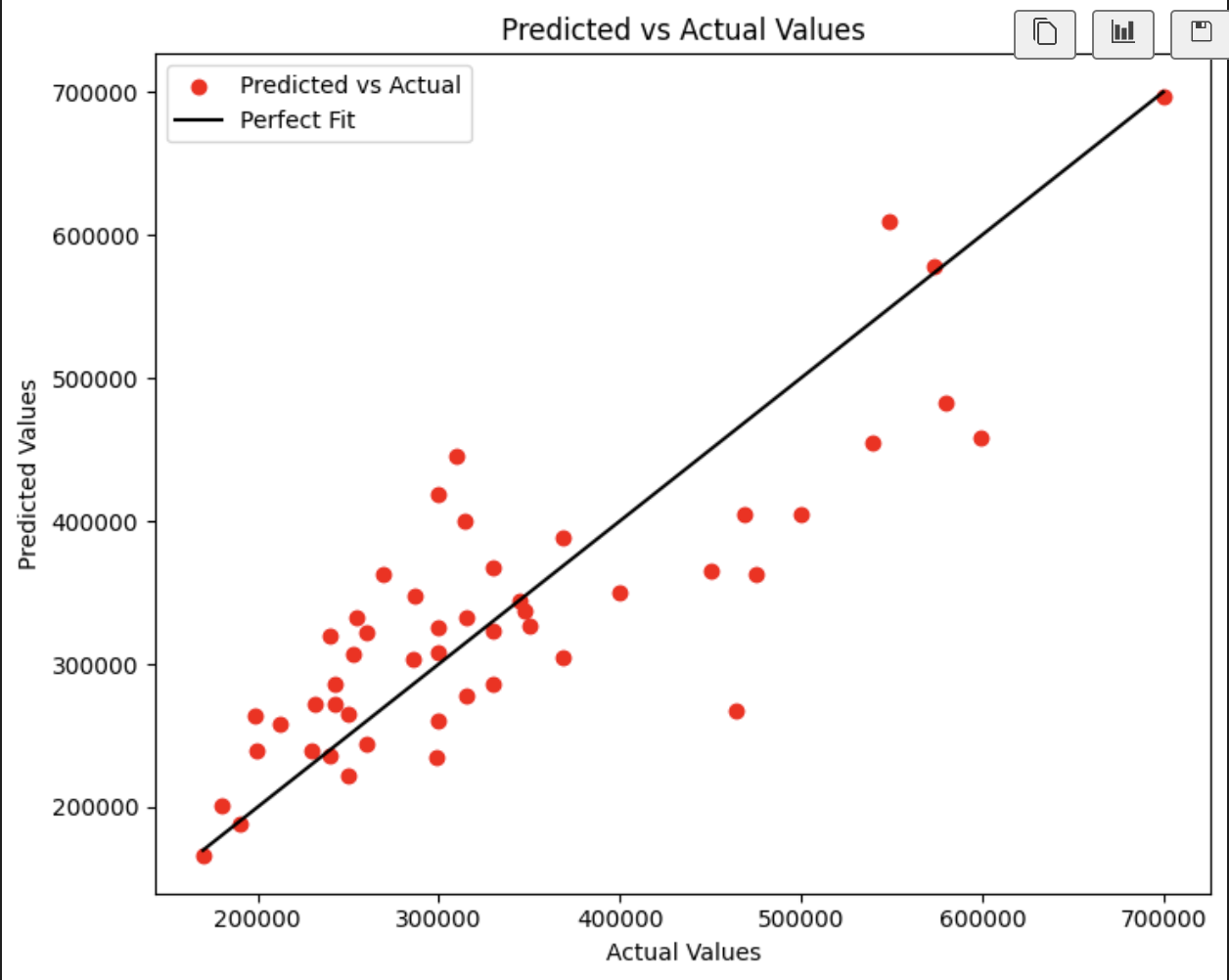
* Predicted the house prices using the hypothesis function.



* House prices predicted by the model.



* I then plotted the results graphically using a python library matplotlib.



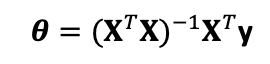
**Explanation:** This graph shows the comparison between predicted house prices (y-axis) and actual house prices (x-axis) using our linear regression model. The red dots represent individual houses, and the black line shows where predictions would match the actual values perfectly. The closer a red dot is to the black line, the more accurate the model's prediction for that house.

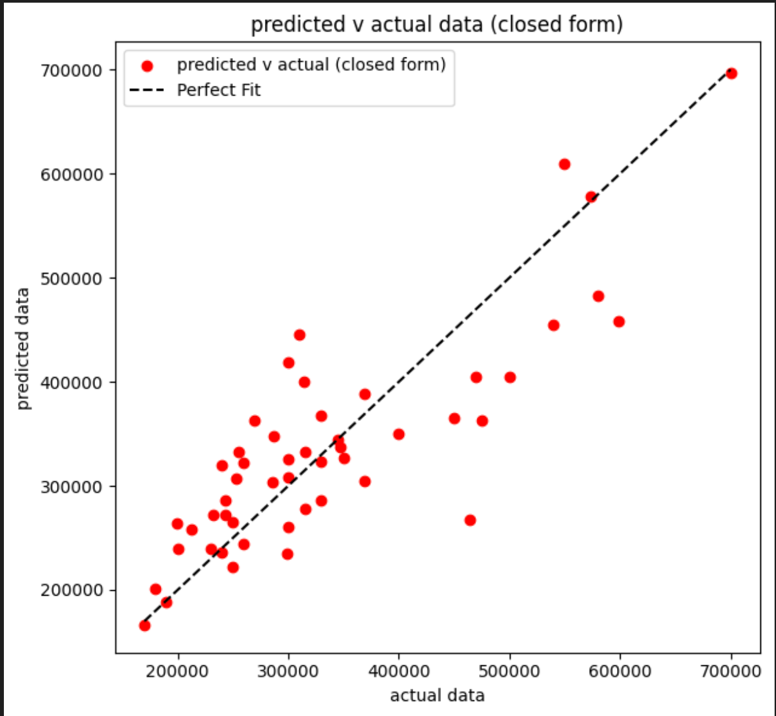
Most of the predictions are close to the black line, indicating the model performs well, though some higher-priced houses deviate more. This means the model is relatively accurate for lower-priced houses but struggles more with predicting higher prices.

**RESULTS OF PART (b)**

**CLOSED FORM SOLUTION:**

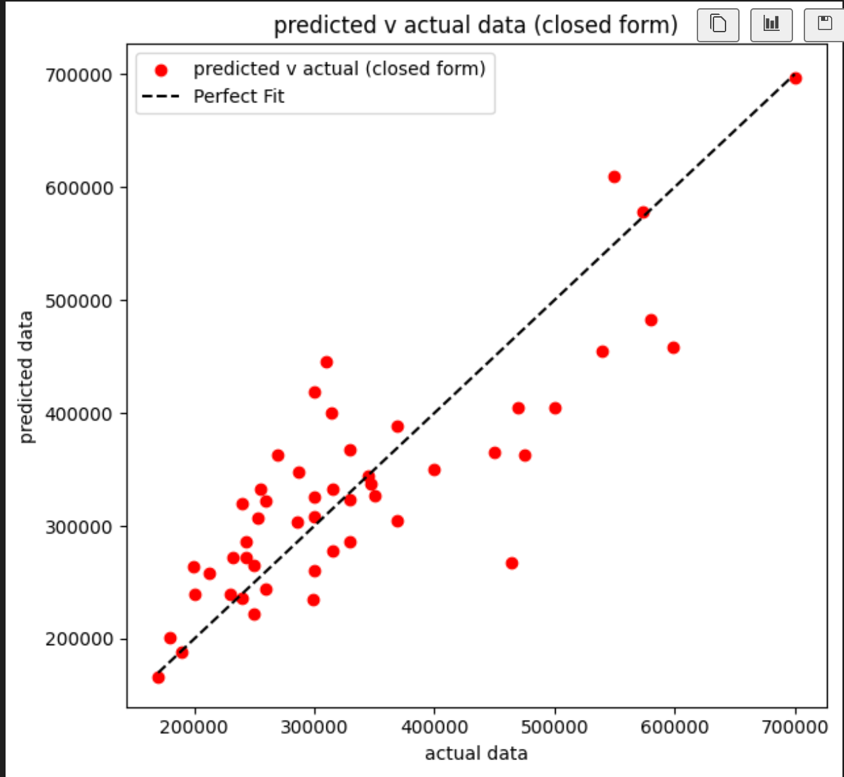
* I used the closed form solution to directly calculate the values of regression coefficients.





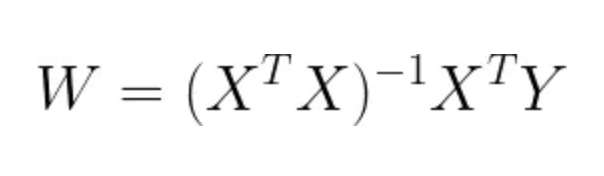
**PART (d)**

**COMPARISON OF THE RESULTS OBTAINED FROM PART (a) AND (b)**

****

**Explanation:** In both the closed-form solution and gradient descent, I noticed that the predictions are quite similar. When I compared the two graphs, the red dots in both are close to the perfect fit line, which means both methods are making pretty accurate predictions for house prices. There’s only a small difference, especially in the higher price range, but overall, they both perform almost the same.

I used the closed form solution.



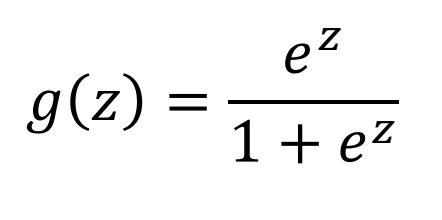
This allowed me to directly calculate the optimal parameters without having to go through multiple iterations like in gradient descent. Despite using different methods, the results were very close, showing that both approaches work well for this dataset.

**QUESTION NO 2:**

For question number 2, I classified the houses as "costly" or "not costly" using logistic regression.

I used the same preprocessing as in question 1, which involved z-score normalization and converting the data into NumPy arrays. This standardization helps in improving the performance of the logistic regression model.

I implemented logistic regression from scratch, including a sigmoid function to map values between 0 and 1, a cost function to measure the model's performance, a gradient function for optimization, and a logistic regression function using gradient descent.



After training, I used the model to predict whether each house is "costly" (1) or "not costly" (0). The prediction is based on whether the sigmoid output is greater or less than 0.5.

The results are presented in a Data Frame showing both the actual classifications and the predicted classifications for each house. This allows for easy comparison between true labels and predictions, which can be used to evaluate the model's accuracy.

A highlight of the results from the logistic regression model is this

