
നാഷണൽ ഇൻസ്റ്റിറ്റ്യൂട്ട് ഓഫ് ടെക്നോളജി കാലിക്കറ്റ്
राष्ट्रीय प्रौद्योगिकी संस्थान कालीकट
NATIONAL INSTITUTE OF TECHNOLOGY CALICUT

# Department of Computer Science and Engineering

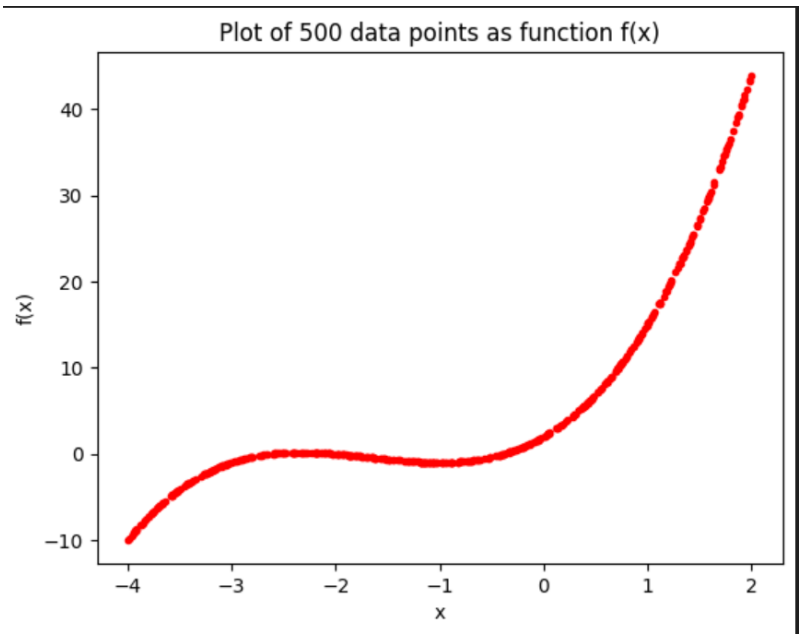
## Topics in Data Analytics (CS6155D)

ASSIGNMENT SOLUTIONS

**Submitted by,**

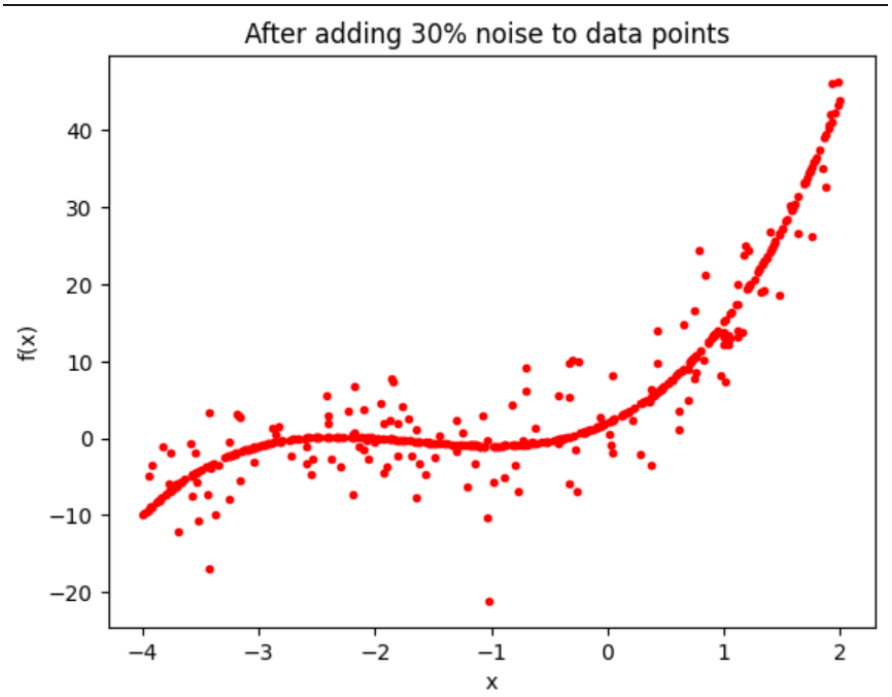
**Jay Vora (M220249CS)**

**Q1**




The resulting plot should show a curve that rises steeply on the left side, levels off near x=-2, and then falls off more gradually on the right side.

**Q2** – Using random function to add noise.




In this code, we use the np.random.normal function to generate properly distributed random noise with mean 0 and standard deviation 10, and add this noise to the selected points in y_values. The final plot ought to display a curve that resembles the initial curve, but the chosen points have some random fluctuation.

## Q3

The prediction model is built using Batch Gradient Descent.

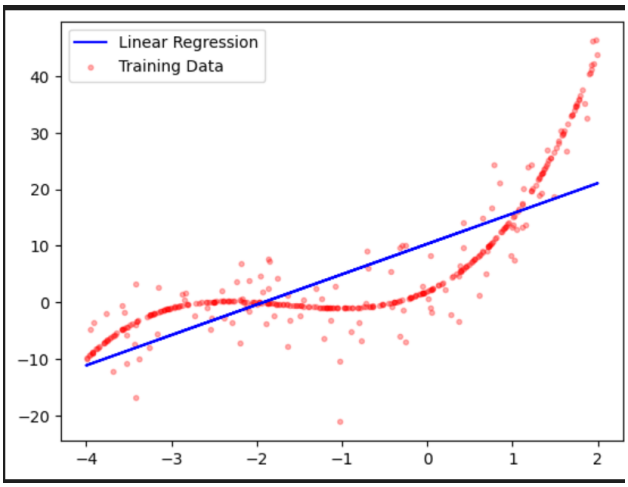
Learning Rate – 0.01

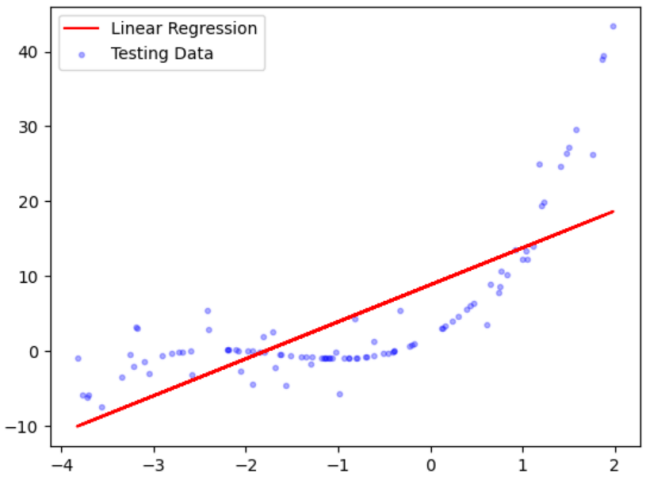
Epoch – 10000

## Q4

Train Error(RMSE) - 7.109821765248552

Test Error(RMSE) - 7.185593442586261




## Q5

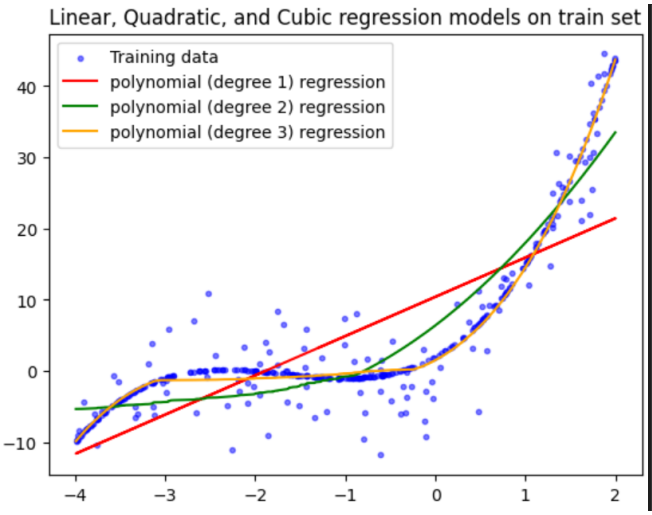
RMSE values

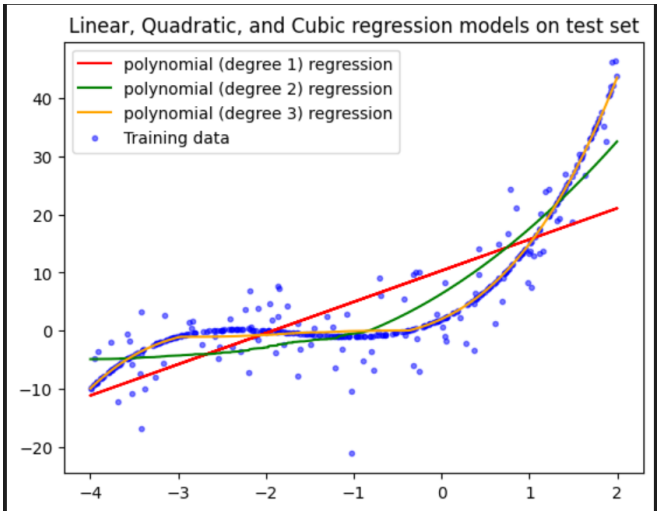
Degree 1 - 7.109821765248552

Degree 2 - 4.5469270673059405

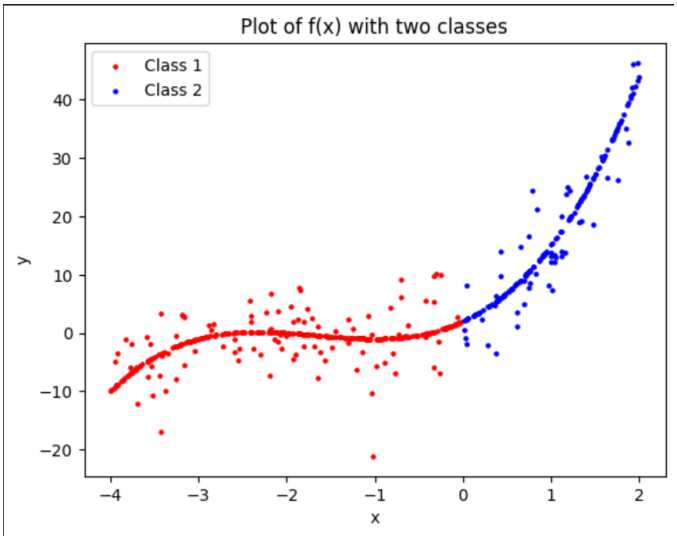
Degree 3 - 2.5673136972027373

Degree 3 polynomial will be giving the best result as it has a lower RMSE showing the best results.



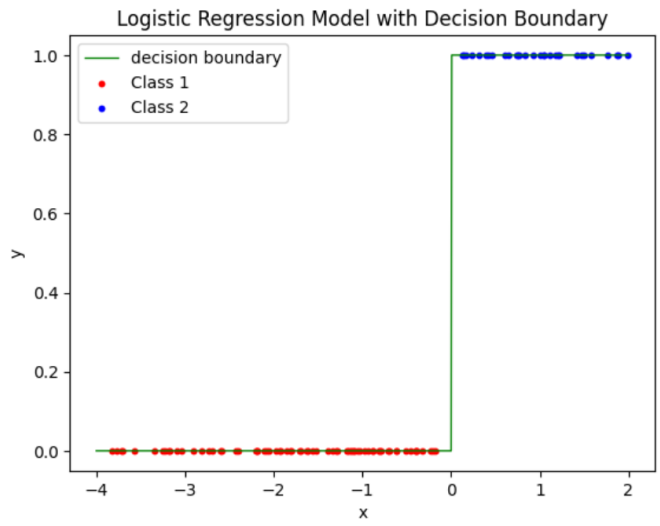




**Q6**



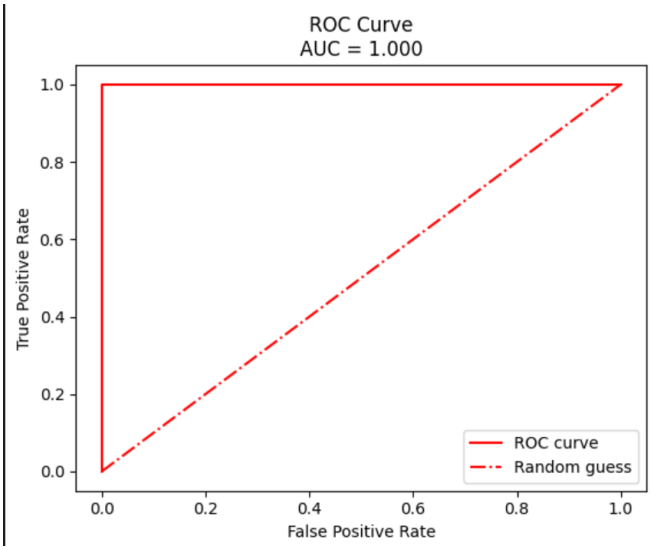

This code first classifies the points into two classes based on their x values, and then plots the points using blue color for Class 1 points and red color for Class 2 points. Finally, it adds labels and a title to the plot and displays it.

**Q7**



In the code above, we first classify the given points into two classes based on their x values, with points less than 0 belonging to Class 1 and points greater than or equal to 0 belonging to Class 2. Then, we apply logistic regression to discover a decision boundary that distinguishes between the two classes. We build a logistic function to convert the output of the linear function (which sets the decision boundary) to a probability value between 0 and 1. The weights that minimize the loss in cross-entropy between the true labels and the predicted probability are then determined using gradient descent.

**Q8**




We may utilize the roc_curve and auc methods from the sklearn.metrics module to compute the TPR, FPR, and AUC. Initially, this algorithm forecasts the probabilities for utilising the test set and the logistic regression model that we previously built. It then determines the using the roc_curve function, calculates the FPR, TPR, and threshold values. A ROC curve is displayed using these values and a dashed line to reflect guesswork. In the AUC, computed and printed on the terminal using the AUC function.