

DATA 240-21, Fall 2024

Assignment #3

Release on Nov 2nd, 2024

Due 11:59pm on Nov 19th, 2024

Notes

This assignment should be submitted in Canvas as a format of ipython notebook (assignment3.ipynb).

No late assignments will be accepted. Do not accept any other format. Minimum penalty is 2pts with acceptable excuse. You may collaborate on homework but must **independently** write code/solutions. Copying and other forms of cheating will not be tolerated and will result in a **zero score** for the homework (minimal penalty) or a failing grade for the course. Your work will be graded in terms of correctness, completeness, and clarity, not just the answer. Thus, correct answers with no or poorly written supporting steps may receive very little credit.

NOTE: Please do not use any open-source algorithm for gradient decent method. Instead, you need to write gradient descent method from scratch.

1. (5 pts) Linear regression with gradient decent

Please download housing.csv. This is a real-life dataset consisting of housing sales prices in the city of Windsor, Ontario, Canada. You can find a description of the variables in housing.txt. Our target variable is 'price'.

Linear regression using gradient descent method from scratch

This is the task of Linear regression with multi variables.

$$\hat{y} = W_0 + W_1x_1 + W_2x_2 + \dots + W_nx_n$$

You need to find the optimum Weights using gradient decent method from scratch.

Before applying gradient descent method, you might need to normalize variables, which is called feature scaling or normalization.

In gradient descent algorithm, weight needs to be updated every iteration.

1-1. (3pts) Please build a gradient descent algorithm based on the following formulas.

$$J = \frac{1}{m} \sum_{i=1}^m (\hat{y}^i - y^i)^2$$
$$W_j^t := W_j^{t-1} - \alpha \frac{\partial J}{\partial W}$$

Please print out the Root Mean Squared Error (RMSE) every iteration (until 10 epochs).

Please fit the data using the Linear regression model with the optimum Weight.

To simply the problem, please plot 'price' vs 'lotsize' as the below. The 'price' and 'lotsize' should be original scale instead of normalized scale.

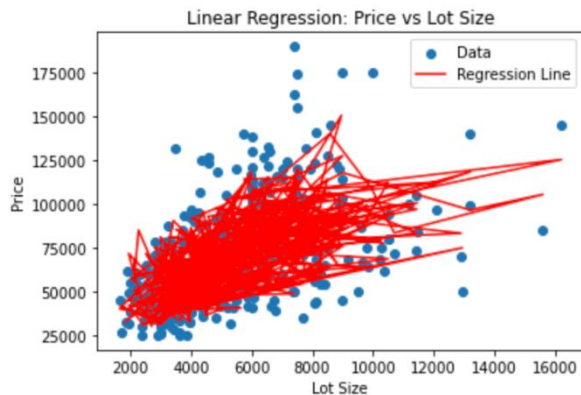


Fig. Bad example

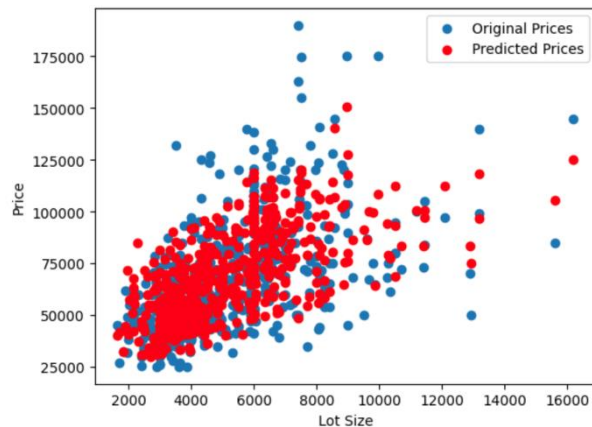


Fig. Good example

Please plot 'true-price'(y) vs 'predicted-price'(\hat{y}) and display R^2 . The y and \hat{y} should be original scale instead of normalized scale.

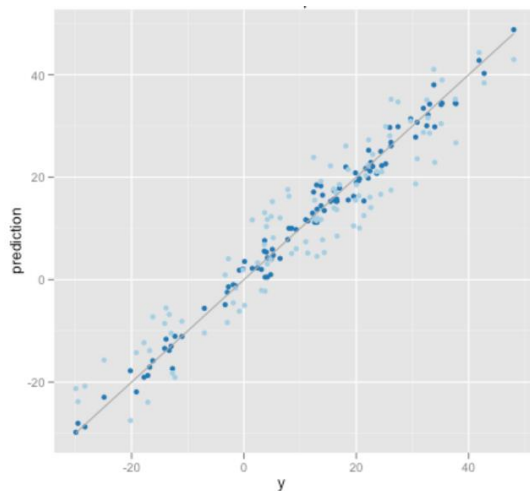


Fig. Plot of y vs \hat{y}

1-2. (2pts) Please build a gradient descent algorithm based on the following cost function.

$$J = \frac{1}{m} (\hat{Y} - Y)^T (\hat{Y} - Y)$$

$$W := W - \alpha \nabla W$$

Please print out the Root Mean Squared Error (RMSE) every iteration (until 10 epochs).

Please fit the data using the Linear regression model with the optimum Weight.

Please plot 'true-price'(y) vs 'predicted-price'(\hat{y}) and display R^2 .

2. (5 pts) Binary classification with logistic regression

Please download `heart_disease_train.csv` and `heart_disease_test.csv`. The dataset is for cardiovascular study. The target variable is 'TenYearCHD', which shows whether the patient has 10-year risk of future coronary heart disease (CHD). You can find a description of the variables in `cardiovascular.txt`.

This is the task of Binary classification with logistic regression. You need to build a logistic regression model from scratch to predict 'TenYearCHD'.

$$\hat{y} = P(y = 1|x)$$

$$P = \sigma(z) = \frac{1}{1 + \exp(-z)}$$

Before applying gradient descent method, you might need to fill out missing value and normalize variables.

Please build a gradient descent algorithm based on the following formulas. Instead, you may build an algorithm based on the matrix formula.

$$J = -\frac{1}{m} \sum_{i=1}^m [y^i \log(\hat{y}^i) + (1 - y^i) \log(1 - \hat{y}^i)]$$

$$W_j^t := W_j^{t-1} - \alpha \frac{\partial J}{\partial W}$$

Please print out the Cost function J for every iteration (until 10 epochs) for the train dataset.

Please print out confusion matrix and its corresponding accuracy, precision, and recall for the train and test dataset after the last iteration.