DATA 240-21, Fall 2024

Bonus Assignment #2

Release on Nov 8th, 2024
Due 11:59pm on Dec 3rd, 2024

Notes

This assignment should be submitted in Canvas as a format of ipython notebook (bonus2.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.
NOTE: Please do not use any package/library including scikit-learn library except NumPy, Pandas, Matplotlib.

This is the following assignment from HW3/#2.
(From HW3/#2) You can copy your work from HW3/#2.

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.

(3pts) Bonus assignment

Please tune your hyperparameters (learning rate, batch size, number of epochs, etc.) to get the best results of accuracy, precision, and recall for test dataset.

Please plot ROC curve and calculate AUC for train and test dataset for your best result.

NOTE: You need to get more than 10 data points for ROC curve.
Please include the curve corresponding to the base binary classification model.
Please do not use any package for ROC curve and AUC calculation. You need to draw ROC curve using Matplotlib and calculate AUC from scratch.