Assignment 03

(5 points)

In this assignment, you use the bankruptcy practice project (see the notebook) with logistic regression. The data sets are provided on HuskyCT as well. The following is the information of the attributes/variables from income statements and balance sheets of the companies:

1) Size
    a. Sales
2) Profit
    a. ROCE: profit before tax=capital employed (%)
    b. FFTL: funds flow (earnings before interest, tax & depreciation)=total liabilities
3) Gearing
    a. GEAR: (current liabilities + long-term debt)=total assets
    b. CLTA: current liabilities=total assets
4) Liquidity
    a. CACL: current assets=current liabilities
    b. QACL: (current assets – stock)=current liabilities
    c. WCTA: (current assets – current liabilities)=total assets
5) LAG: number of days between account year end and the date the annual report and accounts were filed at company registry.
6) AGE: number of years the company has been operating since incorporation date.
7) CHAUD: coded 1 if changed auditor in previous three years, 0 otherwise
8) BIG6: coded 1 if company auditor is a Big6 auditor, 0 otherwise

The target variable is FAIL, either = 1 or 0.

The first part of the provided starter notebook shows how to use scikit-learn logistic regression. Read and try the code to get understanding of needed steps. In practice, data scientists use many open source packages like scikit-learn. In this course, you also practice writing your own code. Start working in "Your Work" section. Copy code from Preprocessing section to "Your Work" section so you can use the preprocessed data in the following questions.

1. (3 points) Write the computeCost() function as follows.

def computeCost(theta, X, y, lambdad=0):

    …

    return cost

This function computes the regularized cost function (given the parameters θ, features X, target variable y, and the regularization parameter λ:

$$J(\theta) = \frac{1}{m}\left[\sum_{i=1}^{m}\left(-y^{(i)}log h_\theta\left(x^{(i)}\right) - \left(1-y^{(i)}\right)log\left(1-h_\theta(x^{(i)})\right)\right) + \frac{\lambda}{2}\sum_{j=1}^{n}{\theta_j}^2\right]$$

2. (3 points) Write your own gradient descent function to update θ, namely optimizeCost())

Repeat {

$$\theta_0 \coloneqq \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x0^{(i)}$$

$$\theta_j \coloneqq \theta_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)xj^{(i)} + \frac{\lambda}{m}\theta_j\right]$$

} Until convergence

The function would take X_train, y_train for training the model and returns θ. You will decide other parameters on your own (such as α, λ).

3. (1 point) Write the sigmoid() function

def sigmoid(z):

…

return g

which returns a sigmoid function of z

$$h_\theta(x) = \frac{1}{1 + e^{-z}}$$

In which,

$$z = \theta^T x$$

4. (2 points) Write needed main code that calls functions above to train a model using the training data and return θ, then test the model using the test data, compute accuracy, confusion matrix, precision, recall (Note that in this assignment, model validation is not required).

5. (1 point) With the provided no-label test set, predict the label using the trained model. You should select only variables that were selected for training.

Submission: A zip file including the article in a Word file together with comments given to/received from other groups, code, files, images to HuskyCT.