**CS 467/667/767**

**MACHINE LEARNING**

**Mini-project3: Logistic Regression with Newton’s Method**

**Deadline: November 15 (Friday), 2019, 11:59pm**

**Project Report**

# INTRODUCTION

While Python’s popular machine learning library comes equipped with easy to use and efficient Logistic Regression class, through which you can split, train, validate and test your dataset in order to have a better understanding of where to take your next step or proceed via better prediction accuracy. But implementing Logistic Regression with using the Newton’s Raphson method avoids the use of step length. Using quadratic approximation on the function helps to make a better guess of where to head next which can also improve our comprehension of how logistic regression works.

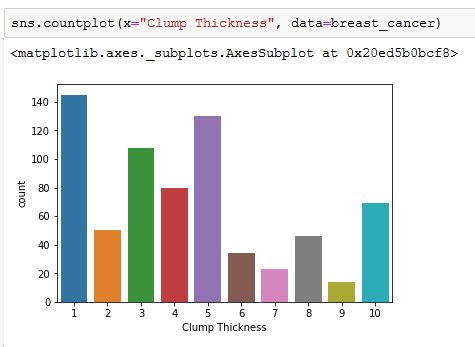
# DATA-SET

Here we will use a well known “Breast Cancer Wisconsin” Data set to calculate the accuracy of the cell classified as benign (1) or malignant (0) based on all the 10 features provided in the data set.

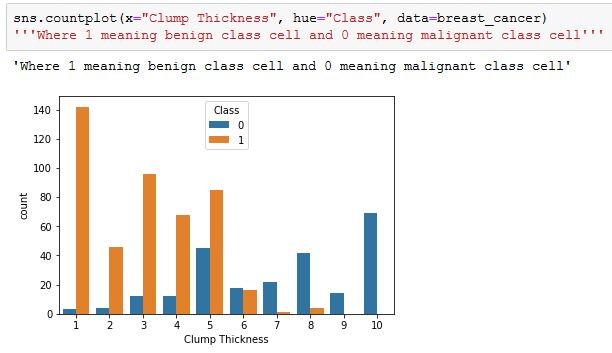
First, we will begin by doing some basic data wrangling, cleaning and preparation by removing unwanted or missing values in the data set and then using it in newton’s method formulas. After splitting the data into 80% for training and 20% for testing sets, we will use the famous classification in machine learning called Logistic Regression for classifying problems to output discrete values. For instance, given an input of the cell class, the logistic function can classify it as malignant or benign.

# Visualizing Dataset for better understanding

Over here we take “Clump Thickness” column to understand the visualization better. On our X-axis we have “Clump Thickness” and Count on the Y-axis, showing all the cell id’s in the plot with unique color identification meaning that “Clump Thickness” with id (1) are in majority and id (9) in much less count as compared to others.



In the second figure below, we have classified cell type in (1: benign) meaning not harmful in effect and (0: malignant) meaning very virulent or infectious. Here in the “Clump Thickness” property, we can see that id 1 cells are majorly classified as benign class with few malignant class and id 10 with majority classified as malignant cells and none benign class.



# Newton's Method

Newton's method is a second-order optimization algorithm that can help us find the best weights in our logistic function in fewer iterations compared to batch gradient descent.

The generalization of Newton’s method to a multidimensional setting (also called the Newton-Raphson method) is given by:



Where the Hessian is represented by:

https://github.com/jrios6/Math-of-Intelligence/raw/e8003e67e0f112d4b3213f4597bbfbfc739bc7fc/2-Second_Order_Optimization/images/hessian.png

For Logistic Regression, the Hessian is given by:



and the gradient is:



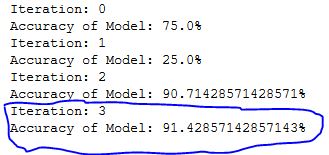
Where:



and *p* are the predicted probabilities computed at the current value of β.

# Testing

After applying all the necessary formulas of newton’s method by using popular python Numpy library, the accuracy of our model trained using Newton’s method is:



We also verify the accuracy by calculating it with Logistic Regression via python’s famous machine learning library called Scikit library:



As observed, using Newton’s Method, the model achieves a test accuracy of 91.428% in few iterations whereas Scikit library achieves a slightly higher accuracy of 96.428%. Both models are good for prediction/predictive analysis and for verifying as well.

# REFRENCES

1. https://ml-cheatsheet.readthedocs.io/en/latest/loss\_functions.html
2. https://scikitlearn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html
3. http://www2.stat.duke.edu/~sayan/CBB2012/binreg.pdf