



NORTHEASTERN UNIVERSITY, KHOURY COLLEGE OF COMPUTER SCIENCE

---

## CS 6220 Data Mining — Assignment 5

Due: March 15, 2023(100 points)

---

YOUR NAME  
YOUR GIT USERNAME  
YOUR E-MAIL

### Naïve Bayes, Bayes Rules

The original performance of [acoustic classification for Parkinsons Disease](#) leverages speech recordings from controlled subject responses from variety of questions. The task in the competition was to detect whether or not a person  $X$  had Parkinsons disease from a sampling of data. As of 2018, the state of the art classifiers have achieved 90% correct classification on a held out dataset, both for subjects who had Parkinsons and those who did not (at equal rates). So, when classifier  $Y$  sees person  $X$ , it works correctly 90% of the time.

#### Question 1 [30 pts total]

Let's say that we run a clinic. This clinic leverages this classifier, which has 90% accuracy. Also, let us say that we know that our current patient load is that 10% of the population have Parkinsons and 90% of the population do not. Let's also say that we're seeing patient  $X$ , and the classification algorithm has detected that they have Parkinson's disease. What's the probability that indeed  $X$  has Parkinson's disease?

Come up with the numerical solution, and show your written work.

### Gradient Descent - Logistic Regression

The sinking of the Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg. Unfortunately, there were not enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew. While there was some element of luck

involved in surviving, it seems some groups of people were more likely to survive than others. In this challenge, Kaggle asks you to build a predictive model that answers the question: “what sorts of people were more likely to survive?” using passenger data (i.e., name, age, gender, socio-economic class, etc).

For **Question 2b.)**, we’ll be using logistic regression with binary cross-entropy cost function appearing in the following form:

$$\mathcal{L}(W, b) = - \sum_i y_i \log h_{W,b}(\mathbf{x}) + (1 - y_i) \log (1 - h_{W,b}(\mathbf{x}))$$

where

$$h_{W,b}(\mathbf{x}) = \sigma(W^T \mathbf{x} + b)$$

and

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

### Question 2a.) [20 pts]

Before solving anything with Python, prove (on paper) that the derivative of the sigmoid function is

$$\nabla_z \sigma(z) = \sigma(z) (1 - \sigma(z))$$

### Question 2b.) [30 pts]

We’ll explore logistic regression with data that originated from the Kaggle site (but it is *not* that exact data). Go ahead and download the data from [the homework 5 data folder](#). Train a logistic regression predicting who would survive with `titanic.train.csv`. Test to see your accuracy on `titanic.test.csv`, where accuracy is defined by:

$$\text{acc} = \frac{\text{num correct}}{\text{total}} \tag{0.1}$$

Make sure you’re not including the `survived` column in the data as one of your features (that’d be cheating!) If you’ve done the lab in class, then this simply becomes an exercise of data transformations and preprocessing. (Since there is only one label that you are predicting, you may also have to ensure that the dimensions are correct.)

Some tips and tricks:

- For features with multiple values, try making one-hot encoded data. For example, number of siblings, you can try to have a five column array.

No Siblings	1 Sibling	2 Siblings	3 Siblings	4+ Siblings
0	1	0	0	0

- Play around with your learning rate. If your learning is too high or too low, then you won't converge to the right answer.
- You don't use all the features. Just because they're there, doesn't mean they're always useful. (I ignored using the names.)
- Make sure you normalize your features to have similar ranges. You needn't use the standard scalar (i.e., z-scores), but if you don't transform some features, the logistic regression will not work well.

See if you can do better than I can with a **logistic regression**. On the test set, adopting the code that we used in class, I reached 79.2% accuracy at a threshold of 0.5, using just a few features.

For this question, submit a plot of your loss curve and print your overall accuracy on training and test sets. Include what features you used, what you did to transform them, and also include what you used as a learning rate.

### Question 2c.) [20 pts]

Using a toolbox (like Keras), try benchmarking your logistic regression and then adding a few more layers (i.e., a neural network). Diagram your neural network architecture and print out your overall **test set** accuracy. Make sure you do *not* train on any samples from your test set.