CS 59000SA **Fall 2018**

Assignment 2

Instructor. Ninghui Li

Due: 11:59pm Sep 21th, 2018

Please typeset your homework solutions, ideally using LaTeX. No hand written assignment will be graded.

1. (100 pts)**Simple Spam e-mail filter**

Email is a life changing productivity tool in modern life. We rely on it for communicating with our friends, colleagues, and etc. However, spam emails are headache for many of us for its hardness to detect. In this assignment, you are going to implement a simple e-mail filter using machine learning algorithms covered in the lecture.

Traning/Testing Dataset: In this assignment, a real life dataset will be used. You can download the dataset on Piazza (CSDMC2010_SPAM.zip). The format of the dataset is described in **readme.txt**, please make sure you read it carefully.

Task 1(40 pts): Naive Bayes:

Hint: `https://en.wikipedia.org/wiki/Naive_Bayes_spam_filtering`

Your first task is to implement a spam email filter using Naive Bayes method covered in the lecture. You should name it as NB.py. In addition, you should write a file named NB_readme.txt to tell us how to run NB.py.

In this part, you **must write your own naive bayes algortihm**, using any existing bayes algorithm from library is not allowed and will result in serious penalty.

Task 2(30 pts): SVM

Recommendation: `http://scikit-learn.org/stable/modules/svm.html`

Your second task is to implement a spam email filter using SVM method covered in the lecture. You should name it as SVM.py. In addition, you should write a file named SVM_readme.txt to tell us how to run SVM.py.

In this part, you can use **any** existing SVM library. Your goal is to read the documentation of your choices of library and use it. You may have to convert the input data format based on your choice.

Task 3(30pts): Report

3.1 For each of the algorithm, record the following for **training and testing data**:

- False Positive Rate
- False Negative Rate
- Recall
- Precision
- F-Score

3.2 Evaluate the results of both algorithm, which one is better from your point of view and why? Is your model overfiting/underfiting? How do you know whether your model is overfiting/underfiting or not?

3.3 Cross Validation: Make a 5-fold cross validation for your SVM. Briefly explain how you prepare the 5-fold dataset and the result

3.4 For each of the algorithm, try to come up with an spam instance with word "get", "free", and "iphone" which can circumvent the detection and describe how you construct such instance.

Grading: The score for task 1 and 2 will be graded based on f score of both training and testing data. For full credits, you should achieving at least 0.95 for both tasks.

- Task 1 40 pts
- Task 2 30 pts
- Task 3: 30 pts
  - 3.1 15 pts
  - 3.2 10 pts,
  - 3.3 5 pts
- Coding standard: Up to 10 points deduction. Your program should be easy to read. (Friendly variable names, sufficient comments, well structured, and etc)