Homework # 2: Probability, Naive Bayes, and Linear Regression

Due: Saturday, March 20, 2021, 11:59 p.m.
Total Points: 100

This assignment is designed to give you practical programming experience with probability, naive bayes, and linear regression. **Please carefully read all instructions below and also periodically check Piazza for updates.**

You have been assigned a partner to work with on this homework assignment (see end of document). A spreadsheet with student contact information is on the Syllabus page of Canvas. Use this information to contact your partner and to establish your working relationship. Submit a post to Piazza (to Instructors) if you have any issues connecting or working with your partner.

Each individual within the pairing should submit a solution to the assignment, to avoid miscommunication issues with your partner. Be sure to specify in the file, whose submission should be graded, as only one will be graded for the pairing. The submitted homework must include your names, as well as all resources that were used to solve the problem (e.g. web sites, books, research papers, other people, etc.). Academic honesty is taken seriously; for detailed information see Indiana University Code of Student Rights, Responsibilities, and Conduct.

Your assignment must be submitted by committing your code to your IU Github repository before the deadline. Please create a folder called 'hw2' (without quotes) and commit all files for this assignment to this folder. You are strongly encouraged to submit early and often to avoid any last minute issues and to confirm that you are committing your work correctly. The submission should be in the form of a Python 3 Jupyter Notebook file, along with any special instructions for running the program. **Be sure to run the file before committing, so that we can directly see your results**. Programs that fail to run will not be graded and will result in a zero.

Late submission policy: We expect all work to be finished and submitted on time. However, we do understand that there might be times when something unexpected comes up which delays you. Hence, as a late submission policy, we will allow late submissions up to 3 days late, each day carrying a 5 point penalty. If you submit 70 hours after the original deadline, the highest possible score for that assignment will be 85 points, out of 100. If the assignment is submited 1 minute after the deadline, then a 5 point penalty will be assessed. Late assignments submitted at 72:01 hours after the original deadline will not be accepted. Please, make the arrangements to start and finish your assignments early. This means you should push to Github well before the time stated on the homework. This means you must start the homework as soon as you can and continually commit.

Question 1. [50 POINTS]

The CSV file, *message.csv*, contains labeled data, where each row contains the label (e.g. spam or not) and the corresponding text message. A value of 1 for the label indicates that the message is spam. This data has already been cleaned (e.g. removed punctuation and converted to lower case).

For this problem, you will develop a Naive Bayes spam detector to classify whether the message is spam or not. Using stratified K-fold cross validation (K = 10) and an 80-20 split between training and testing data, you need to perform the following:

1. From the training set, compute the total number of unique words in the set and the count of each unique word in each message. Hence, if there are N unique words and M messages in the training set, then the count of each unique word for all messages should result in a

$M \times N$ matrix. You may want to use DataFrame and dictionary objects to accomplish this. You may also use *split()* to ignore whitespace.

2. Perform maximum likelihood estimation to determine the prior and class conditional probabilities of the training set (e.g. compute $P(y=1), P(y=0), P(x_i|y=0)$, and $P(x_i|y=1)$) , where x_i represents the i-th unique word. Be sure to confirm that these are indeed probabilities.
3. Once the above probabilities are determined, use Naive Bayes classification to classify each of the testing examples as spam or not. Ignore words from the testing set that are not contained in the training set. Report the accuracy, precision, recall and specificity, along with the confusion matrix for each fold. Also report the average accuracy, precision, recall and specificity over all folds.
4. Write a paragraph the summarizes the results and your thoughts about Naive Bayes classification for this problem

One problem with Naive Bayes classification is that the class conditional probabilities for each feature $P(x_i|y)$ may be zero in many cases. This is a result of using limited data. One way to correct this is to "smooth" the values when computing the probabilities, where Laplace smoothing is one approach.

Assuming z is a random variable that has G different possible outcomes, then the conditional probability of z given y can be calculated as below when using Laplace Smoothing:

$$P(z=g|y=0) = \frac{n_g + 1}{n + G} \tag{1}$$

n_g is the total number of times $z=g$ when $y=0$, n is the total number of examples when $y=0$.

Repeat steps 2-4 from above, where Laplace Smoothing is used when calculating the class conditional probabilities for each word. Discuss how the results after Laplace Smoothing differ from the prior results.

Extra work (Not part of grade): You can try to remove and subsequently ignore 'stop words' from the list of unique words as well, before completing steps 2-4. Check out this link for why and how this can be done *https://www.geeksforgeeks.org/removing-stop-words-nltk-python/*

Question 2. [50 POINTS]

A hospital staff member wants to determine if a patient's satisfaction with the hospital (y) can be predicted from the patient's age (x_1, in years), severity of illness (x_2), and anxiety level (x_3). Forty-six patients were randomly selected, where the data is in the file *patient_satisfaction.txt* with column 1 containing y-values, and columns 2-4 containing values for x_1, x_2 and x_3, respectively. Large values indicate more satisfaction, illness severity, and anxiety. For this problem:

1. Generate scatter plots between each of the features and label. Also compute the correlation coefficients between each feature-label pair. What do the scatter plots and correlations convey about the different relationships?
2. Divide this data into K=10 folds of training and testing sets. Using your own implementations of batch, stochastic and mini-batch gradient descent, fit a linear regression model using the training data. Generate plots of the training loss for each iteration and implementation. Experiment with different values for the learning rate and mini-batch size. Discuss how you selected the optimal values for the learning rate and mini-batch size. Display the final regression coefficients for each implementation.
3. Once trained, use the testing data and mean-square error to evaluate performance. Which approach performed best? Why?