**CSCE 623: Machine Learning**

**Spring 2019**

**HW2**

Due Tue, 23 Apr at 2359

Submit via Canvas

**(**This Homework is worth 5 points toward your final grade**)**

Your homework will be in jupyter/ipython notebook format - composed of an integrated written portion (markdown) and python programming. Include any function definitions in this file (one function per cell). Ensure your cells are well labeled with the steps listed in this instruction set.

You will be using machine learning techniques on several datasets provided. In your answers to written questions, even if the question asks for a single number or other form of short answer (such as yes/no or which is better: a or b) you must provide supporting information for your answer to obtain full credit. Use python to perform calculations or mathematical transformations, or provide python-generated graphs and figures or other evidence that explain how you determined the answer.

The 3 synthetic datasets (dataset1.csv, dataset2.csv, dataset3.csv) contain observations rows with 2 numerical features (X) and labels (y = 0 or 1). Your task is classification. You will evaluate the efficacy of several machine learning algorithms (logistic regression, LDA, QDA) using assessments and tools such as accuracy, precision, recall, F-measure and ROC curves. You will also gain familiarity of working with training and testing sets. You will find hints in the ISLR book lab for chapter 4.

**The Backstory: You are a potential vendor trying to convince a customer that your company is capable of providing machine learning services (including consultation). The customer decides to give you a few datasets and ask you to develop a report (and associated code) for answering some questions:**

For each dataset:

A. Which classification model is the best overall model to use – and why?

B. For that classification model, what is the best threshold parameter setting for *c* in Pr(Y=1|X=x)>=*c* … and why?

Grading Criteria: Your Instructor will be evaluating your work in two parts – holistically, and individually, by pieces. The holistic component is worth 2 points and the piecewise component is worth 3 points.

**Holistic Grade Component (2 points)**: The instructor determines this grade *as if they were a customer considering whether to invest in your company based on the quality, correctness and completeness of your report*. The component is worth up to 2 (integer) points: Does the report provide recommendations and convincing evidence for the conclusions drawn when answering questions A and B above?

Points = 0 if some recommendations are missing OR all recommendations are provided, but at least some of the recommendations provide no supporting evidence.

Points = 1 if all recommendations are present and provide evidence, but at least some evidence is confusing, misleading, or doesn’t support the conclusion.

Points = 2 if all recommendations are present *and* convincing evidence is provided for each conclusion.

**Piece-wise Grade Component (3 points)**: The instructor will grade your report based on correctness/completeness of the steps outlined in the assignment.

**Comparing 2-feature Logistic Regression, LDA & QDA performance**

Each step listed below should correspond to a numerical step identified in your code and a section of text in your report. One python notebook will be used to handle the entire code and report.

**For EACH dataset (**dataset1.csv, dataset2.csv, dataset3.csv**)** follow these steps. Note that you should interleave the steps (each step contains each dataset) to allow maximum capability to compare differences among the datasets and the performance of the methods on each dataset:

1. Load the dataset
2. Explore the dataset by plotting the data points from both classes as a function of X1 (x-axis) and X2 (y-axis) scores in colors according to their labels (for example, one class is red, the other class is blue)
3. Discuss the dataset. What do you notice about the distribution of the data? What can you say about the covariance of the two classes? Within each class, are the variances for each feature equal? Between classes, are the variances of a single feature equal? How well are the classes separated? Which predictor do you think will work best under this condition (Logistic Regression, LDA, or QDA)… and why?
4. Make a function to return a test set and training set from the full dataset. Your split should be parameterized so that you can declare how many datapoints to use as training. For now, set the number of training points to half and the number of test points to half. Be careful to ensure that you don’t end up with uneven distributions of classes in each of the two sets (the training and testing sets should have equivalent proportions from each class).
5. Fit a model for each of the three classifiers (Logistic Regression, LDA, QDA) using only the training set.
6. For each trained classifier, use the test set to determine the *probabilities* for which each classifier believes the datapoint belongs to class 1: Pr(Y=1|X=x) where x is the datapoint observation
7. Build a function with the signature: def getROCdata(truthVals,probs,thresholds)  
   where truthVals is a column vector that contains the *correct* classification for all test datapoints; probs is a column vector that contains the probability that the model believes the datapoint to be of class 1; and thresholds is a vector of probability thresholds to use when deciding to predict that it is class=1 if Pr(Y=1|X=x)>threshold[i], and class=0 otherwise.   
   This function should return a pandas dataframe with rowcount = len(thresholds), and a total of 10 columns named appropriately as outlined below (a through j). Each row includes a probability threshold in the left column followed by columns containing the 9 performance measures listed below (computed at that probability threshold). The function should thus return these 10 columns in the dataframe:
   1. Probability threshold (from function input)
   2. True Positive count
   3. False Positive count
   4. True Negative count
   5. False Negative count
   6. True Positive Rate (aka Recall)
   7. False Positive Rate
   8. Accuracy
   9. Precision
   10. F‑measure
8. For each model, smartly\* generate a vector of 100 probability threshold values to test and call your getROCdata function to obtain the response. There should be 100 rows in the returned dataframe - which represent the values computed for each of those possible probability thresholds (\*note – make sure you choose your range of probabilities carefully since choosing a probability threshold below the minimum or above the maximum found in the model will lead to a degenerate prediction set (all predicted positive or all predicted negative).
9. Using the response from the getROCdata function, plot Receiver Operating Characteristics (ROC) curves for each of the three classifiers on a single plot. Each ROC curve should use a different color. Make your axes labels and legend appropriately to clearly identify the mapping between color and classifier.
10. Develop (possibly using math found on the internet) a function to compute Area under the Curve (AUC) for ROCs and report AUC for each classifier. You will need to deal with partial information since the curves may not extend the full range from 0 to 1 in both True Positive Rate and False Positive Rate. State your assumptions about how you built the AUC computation in a jupyter notebook markdown cell.
11. Using the ROCdata from your function, for each model (Logistic Reg, LDA, QDA) determine the probability threshold(s) for which each of the following performance measures is maximized: Accuracy, Precision, Recall, F-measure (there might be as many as 4 probability thresholds per classifier). Then report a confusion matrix table of predicted class vs. true class (like table 4.5 in the text) at each threshold value. Examining the confusion matrices, explain what tradeoff is occurring when we set a probability threshold differently to maximize each of those performance measures.
12. Add text to the ROC graph to annotate points on the ROC graph which represent the maximum Accuracy, Precision, Recall and F-measure points on the ROC graph for each model. What do you notice about these points? Where are they along the ROC curve?
13. Now answer the Customer’s Questions:
    1. For each dataset, describe which model you recommend the school use for their decision-making (and why).
    2. Indicate which probability threshold value (or values) you would recommend they set the classifier to use if they wanted to balance the risk of false positives and false negatives.

Hints… Suggested Python imports:

numpy

matplotlib.pyplot

matplotlib.colors

pandas

sklearn.linear\_model.LogisticRegression

sklearn.discriminant\_analysis.LinearDiscriminantAnalysis

sklearn.discriminant\_analysis.QuadraticDiscriminantAnalysis

**Rules of Engagement for this Homework Assignment:**

**Using external sources:**

The use of pre-existing solutions to answer assignments is not allowed. This includes the use of other students’ answers, answers found on the internet, solution manuals, and any other source of information which does not reflect your own work.

You may use the internet or get help from peers when determining basic things like “how do I add points to a plot in python” or how do I use sklearn, but don’t try to search for specific answers to problems I ask in the homework.

You may use any pseudocode or concepts learned in class to solve the problem.

The code you write must be original work.

**Submission Contents:**

You must submit a python notebook (jupyter), which contains text, code and results in a single file.

**Programming Conventions**

In code, good software engineering principles apply: self-documenting code (meaningful function & variable names), additional comments and whitespace should be standard in all code you turn in.

When developing code, place the dataset files in the same directory you are working in, and ensure that your python code loads and processes these files – your instructor will set up the same file structure when evaluating your code.

You should explain what you are doing in text in the markdown as well as in the comments within code chunks. A rule of thumb is to have line-level comments in the code cells and save the larger high level comments/discussion for the markdown text outside of the cells.

**Pre-submission Checklist:**

Ensure your text, code, and figures are present in your notebook before submission. Do not submit the datafiles (CSV).

Make sure you run all cells in a clean environment from the beginning, and generate and read through the output carefully to ensure your final product reflects what you intend to submit. Your instructor will evaluate both the output and the python code cells.

Make sure your name is in the text of the output document.

**Naming Conventions**

Your homework file name should be: “LASTNAME\_HW2.ipynb” where LASTNAME is your last name.

**How to Submit**

Submit your zip file to Blackboard.

**Resubmissions (error correction)**

Note that if you discover an error before the due date and change a problem solution and re-submit, keep in mind that your instructor will only review your latest submission on blackboard – make sure it is complete.