**CSCE 623: Machine Learning**

**Spring 2017**

**HW4**

Due Friday, 12 May at 2359

Submit via Blackboard

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

Your homework will be composed of a written portion and a programming component (unless you choose to do the assignment in a single python notebook – Jupyter). You will be using the ds1.csv dataset 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. Your entire response (including text and figures, tables, etc) will be contained in the written report.

You will implement functions for model selection and regularization. You will be working with multiple synthetic datasets in this assignment. For each dataset, you will explore the behavior of the different techniques to build good models and make inferences about the features. This assignment requires you to apply techniques from classification, the train/validate/test paradigm, and the material on model selection and regularization from chapter 6. You will be evaluated on the choice of techniques and methodology for application, as well as the evidence you present and conclusions you draw with respect to the datasets and models.

You may also wish to look at the package sklearn for machine learning, pandas for dataframe wrangling and matplotlib.pyplot for graphics. Remember to control your randomness for reproducibility using seeds

**Your customer is asking the following questions – you should clearly answer these questions and support your answers with clear evidence in your report:**

A) For prediction of the output variable on each dataset, what are the recommended input features to use for feature counts between 1 and 10?

B) For this type of data, is the extra computational complexity of a best subset approach (over a stepwise approach) offset by the improvement of classification accuracy?

Holistic Grading Criteria: Your instructor will be evaluating your work by answering the following question *as if they were on the customer’s acceptance team reviewing your report*. The question is worth up to 2 (integer) points.

Question: Does the report provide recommendations and convincing evidence for the conclusions drawn?

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

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

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

Detailed Grading Criteria: The steps below are worth 3 points toward your final grade. Each step listed below should correspond to code and/or text in your report. (**Note** – **you may not use any pre-developed code or package to perform best subset or stepwise feature selection** – **for example, you may not use the sklearn functions for feature selection**)

**Part 2: Data setup & exploration**

1. Using pandas, load the “ds\_10.csv” dataset which contains a synthetic dataset with a response variable Y and several features/predictors p1…p10). Note that the dataset is sorted such that all the class 1 points are in the first 150 rows and all the class 2 points are in the remaining 150 rows (*you* will need to shuffle the data before splitting it into training, validation, and testing sets (50 observations from each class in each set). Hint – use train\_test\_split twice.
2. Explore the data using techniques from class and previous homework. Your goal for this exploration step is to try to determine (with your eyeballs) salient features that you think will make good features/predictors for a Linear classification model such as Logistic Regression or LDA.
3. What do you notice about the distribution of the data? State which features (which column names) do you think will be valuable for prediction, and explain why you chose them.

**Part 2: Best Subset Selection: Determining the *Best* model features for each size model**

1. Build a function bestSubset(train, val, k) to implement part of algorithm 6.1 (page 205): steps 1 and 2. The training and validation datasets should be in the form of pandas dataframes with column headers indicating feature identifiers and the class label “Y”. Your function should return both the validation set classification accuracy and the best set of *k* features found for the model – which are the dataframe feature column headers. *Best* is defined as the model which, when trained on the training data, has the best classification accuracy on the validation data. Use LDA for your classifier. This may take a while – for a model of size *k* you will need to fit and evaluate 2*k* models. Note – I want you to design the code for evaluating the best subset of features yourself – don’t use some python package to determine best subset.
2. Execute the bestSubset() function for model size values that range from 1 to *p* to obtain the *p* best sets of features (1 set for each model size). Print out a table of the best features per model size (*k*) – for example, like the output shown in the lab on page 245. Discuss any interesting changes in what the model chooses as features – for instance, did a feature which was selected when *k* = 3 not get selected when *k* > 3? If so, explain why?
3. Create a plot of the validation set classification accuracy of each of the *p* best models (as returned from bestSubset) vs. *k.* Describe the change in these values as the model size grows from 1 to 10.
4. Annotate your plot created in step 6 with the point that yields the best performing model (that maximize or minimize the performance estimation criteria you plotted). This point reveals the best *k*.
5. Report the validation set classification accuracy on the model with the best *k* features.
6. Discuss your findings from the algorithmic best subset selection method and compare to the features you eyeballed as valuable in step 3.

**Part 3: Determining Model Features using forward stepwise selection on LDA.**

1. Write a function forwardStepwiseSubset(train, val, q)to perform forward stepwise selection on a dataset as shown in algorithm 6.2 (page 207) steps 1 and 2. Your function should return both the validation set classification accuracy and the step-wise-selected set of *q* features found for the model – which are the dataframe feature column headers. You should design your code such that in step 2(b) you fit LDA for each model using the *k* features currently under consideration. The “step-wise-best” model here is the one with the highest classification accuracy on the validation set.
2. Execute the forwardStepwiseSubset() function for model size *q* values that range from 1 to *p* to obtain the *p* best stepwise-generated sets of features (1 set for each model size). Print out a table of the best features per model size (*q*) – for example, like the output shown in the lab on page 245. Discuss how the stepwise-selected features changed compared to how the best-selected features changed (Part 2, step 5)
3. Add a different color line to your plot from step 6: plot the validation set classification accuracy of each of the *p* best models (as returned from forwardStepwiseSubset) vs. *q.* Describe the change in these values as the model size grows from 1 to 10.
4. Annotate your plot created in step 12 with the point that yields the stepwise best performing model (that maximize classification accuracy you plotted). This point reveals the best *q*. Report the classification accuracy on the validation set for this best model.
5. Using the best validation accuracy feature sets from each algorithm (best subset; forward stepwise), fit a LDA model on the combined training+validation data. Then use each of two fit models to compute the LDA classification accuracy on the test set. Report the results. Did the optimal subset best model beat the greedy subset model?
6. Discuss the outcomes in terms of the tradespace (accuracy, computational complexity) between the greedy feature selection approach and the optimal feature selection approach. In particular, discuss your findings from the algorithmic forward stepwise selection method and compare to the features you eyeballed as valuable in Part 1, step 3 and to the best subset features chosen in Part 2, step 5. Are the best feature sets from each algorithm (“best-subset” & “forward-stepwise”) models the same? Different? Compare their test set classification accuracy performances. Explain these results in terms of independence or interdependence of the features on classification.

**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:**

The minimum requirement for this assignment is a typewritten report (with figures) – (MS Word or PDF), and a separate python script (.py file) which is executable.

Alternatively, you may 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 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.

If you turn in a jupyter notebook, 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 chunks and save the larger high level comments/discussion for the text outside of the chunks.

**Pre-submission Checklist:**

Ensure your text, code, and figures are present. Do not submit the datafiles.

If you turn in a jupyter notebook, make sure you run all chunks in a clean environment from the beginning, and generate an output document (at least to HTML) 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 chunks.

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

**Naming Conventions**

Your homework file name should be: “LASTNAME\_HW4.zip” where LASTNAME is your last name. Files within the zip should follow similar conventions: “LASTNAME\_HW4.XXX”

**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.