**CSCE 623: Machine Learning**

**Spring 2019**

**HW4**

Due Tues, 14 May at 2359

Submit via Canvas

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

Your homework will be composed of an integrated code and report product using Jupyter Notebook. 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 generate graphs and figures or other evidence that explain how you determined the answer.

You will implement functions for model selection and regularization for regression. You will be working with ISLR’s “Hitter’s” baseball dataset in this assignment. 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 regression, cross-validation for model tuning while exploring feature selection, as well as ridge and LASSO regression, 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 should use the packages 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 estimating the value of the of the output variable (Y) on the dataset, what are the recommended input features (and regularization settings) to use for model sizes with feature counts between 1 and 6?

B) For this data, over all of the techniques explored, which size model yields the best cross-validation model performance, and what are the features of that best model?

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 A: Data setup & exploration**

1. (Code is provided for this step) Using pandas, load the “ISLR\_Hitters.csv” dataset. Clean the data. Split into test and non-test datasets.
2. Explore the non-test data further 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 Regression prediction. **Make a prediction of the top 6 features** that you think will best predict salary. Consider using pairwise plots on the features with salary, as well as correlation. State which features (which column names) do you think will be valuable for prediction, and explain why you chose them.

**Part B: Best Subset Selection: Determining the *Best* model features for each size linear regression model**

1. Write a function bestSubset(X\_nonTest,y\_nonTest, 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 in “X\_nonTest” and the class label “y\_nonTest”. Here, *k* is the size of the model (number of features in the subset) to search over. Your function should return both the list of features of the best model, and its average cross-validation performance (MSE). To pick the best size-*k* model (algorithm step 2b), your function should evaluate each possible size-*k*-subset of all the features, using 5-fold cross-validation over linear regression models. Best subset performance should be determined using the average cross-validation MSE. Your function should return at least the (average) cross-validation MSE and the best set of *k* features found for the model – which are found in the X\_nonTest dataframe feature column headers. **You must design the code for selecting the subsets and evaluating the subsets of features yourself** – don’t use a pre-developed python package to determine best subset. However, you may use a built-in cross-validation routine to execute the 5-fold cross-validation over linear regression models once you have downselected the features for the current subset being evaluated (Note –This may take a while – for a model of size *k* you will need to fit and evaluate 2*k* models.)
2. Execute the bestSubset(X\_nonTest,y\_nonTest, k) function for model size values that range from *k*= 1to 6 to obtain the 6 best subsets of features (1 set for each model size). Present the outputs of the search (e.g. in a table) of the best features per model size (*k*) – for example, a clean version of the type of 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 average cross-validation MSE of each of the 6 best models (as returned from bestSubset) vs. the size of the model *k.* Annotate your plot created in step 4 with the point that yields the best performing model. This point reveals the best *k*.
4. Report *k* and the validation set MSE on the model with the best *k* features. Describe the change in these values as the model size grows from 1 to 6. Discuss your findings from the algorithmic best subset selection method and compare the evidence to the features you eyeballed as valuable in step 1.

**Part C: Determining Model Features using forward stepwise selection with Linear Regression.**

1. Write a function forwardStepwiseSubset(X\_nonTest,y\_nonTest, k) to perform forward stepwise selection on a dataset as shown in algorithm 6.2 (page 207) steps 1 and 2. The training and validation datasets should be in the form of pandas dataframes with column headers indicating feature identifiers in “X\_nonTest” and the class label “y\_nonTest”. Here, *k* is the size of the model (number of features in the subset) to search over. Your function should return both the list of features of the best model, and its average cross-validation performance (MSE). To pick the best size-*k* model (algorithm step 2b), your function should search for the best feature in a size-1 model, then incrementally add the next best feature to the model until the model has k features (Suggestion – Recursion). To evaluate each possible model, use 5-fold cross-validation over linear regression models. Performance should be determined using the average cross-validation MSE. Your function should return at least the (average) cross-validation MSE and the stepwise set of *k* features found for the model (in the order they were added to the model) – which are found in the X\_nonTest dataframe feature column headers. **You must design the code for selecting the subsets and evaluating the subsets of features yourself** – don’t use a pre-developed python package to determine subsets. However, you may use a built-in cross-validation routine to execute the 5-fold cross-validation over linear regression models once you have downselected the features for the current subset being evaluated
2. Execute your forwardStepwiseSubset() function for model size *k* values that range from *k* = 1 to 6 to obtain the 6 best stepwise-generated sets of features (1 set for each model size). Present the outputs of the search (e.g. in a table) of the best features per model size – for example, like the output shown in the lab on page 245. Discuss how the stepwise-selected features changed compared to how the best-subset-selected features changed (Part B, step 5)
3. Update your plot from step 4 by adding a different color line to your plot to represent the forwardStepwiseSubset performance vs. model size: plot the average cross-validation MSE of each of the 6 best models (as returned from forwardStepwiseSubset) vs. *k.* Annotate your plot with the point that yields the stepwise’s best performing model (that minimizes the MSE performance you plotted). This point reveals the best model size.
4. Describe the change in these values as the model size grows from 1 to 6. Report the MSE and the features in the set for this best stepwise model. Discuss your findings from the forward subset selection method and compare the evidence to the features you eyeballed as valuable in step 1.
5. Discuss the outcomes in terms of the tradespace (accuracy, computational complexity) between the greedy feature selection approach and the optimal feature selection approach. Are the best feature sets from each algorithm (“best-subset” & “forward-stepwise”) models the same? Different? Compare their validation set classification accuracy performances. Explain these results in terms of independence or interdependence of the features on classification.

**Part D: Determining Model Features using LASSO Regularization.**

1. Write a function LASSOSubset(X\_nonTest,y\_nonTest, k) to perform a LASSO-based regularization of a linear regression model such that you can determine the best *k* features to use for linear regression. For this step you can use the built-in sklearn functions to perform LASSO as you see fit. Your goal is to use LASSO with a set of (logarithmically spaced) alphas to regularize the fit of the linear regression coefficients and find an alpha value for which exactly *k* features have non-zero coefficients in the model. Your function should then perform a 5-fold cross-validation using the set of *k*-features identified to determine the average MSE of the LASSO-regularized linear regression model with this alpha. Your function should return at least the (average) cross-validation MSE, the set of *k* features found for the model, and the value of alpha for the model.
2. Execute your LASSOSubset() function for model size *k* values that range from *k* = 1 to 6 to obtain the 6 best LASSO-generated sets of features (1 set for each model size). Present the outputs of the search (e.g. in a table) of the best features per model size – for example, like the output shown in the lab on page 245.
3. Update your plot from step 5 by adding another different color line to your plot to represent the LASSO-regularized-models performance vs. model size: plot the average cross-validation MSE of each of the 6 best models (as returned from LASSOSubset) vs. *k.* Annotate your plot created in step 9 with the point that yields the LASSO’s best performing model (that minimizes the MSE performance you plotted). This point reveals the best model size.
4. Describe the change in these values as the model size grows from 1 to 6. Report the MSE and the features in the set for this best LASSO model. Discuss your findings from the LASSO method and compare the evidence to the features you eyeballed as valuable in step 1.

**Part E: Customer Questions**

1. Now answer the customer’s 2 questions based on your exploration of 3 different techniques for feature selection. Remember to provide clear evidence and rationale for your decisions:
   1. For estimating the value of the of the output variable (Y) on the dataset, what are the recommended input features (and regularization settings) to use for model sizes with feature counts between 1 and 6?
   2. For this data, over all of the techniques explored, which size model (and feature set) yields the best model performance?

**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 will submit a python notebook (jupyter), which contains markdown text, code and output/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.

Explain what you are doing in text in the markdown as well as in the comments within code cells. 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.

If datasets are provided by the instructor, place the dataset files in the same directory with your python notebook, and ensure that your python code loads and processes these files – your instructor will set up the same file structure when evaluating your code.

Do not hardcode a path on your computer to get to load data files – the instructor will not be using the same path you placed them in. Do not edit the datafiles – the instructor will use the original file when running your code for evaluation.

**Pre-submission Checklist:**

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

Ensure that each step in the homework description is clearly indicated in your python notebook. Your notebook steps should match the order of the steps indicated in the assignment. If you deviate from the order, include a markdown text description of the deviation and the rationale so your instructor knows where to look for your answer

Before submitting, save the file, close jupyter, shutdown the kernel, and then restart it to make sure you have a clean environment.

Once you have a clean environment, make sure you **run all** cells in from the beginning, and read through the output carefully to ensure your final product reflects what you intend to submit. Your instructor will rerun your code in a clean environment to make sure it works.

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

**Naming Conventions**

Your homework file name should be: “LASTNAME\_HW4.ipynb” where LASTNAME is your last name. (include your first name if there are two or more people in the class with the same last name.

**How to Submit**

Submit your zip file to Canvas.

**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 Canvas – make sure it is complete.