**CSCE 623: Machine Learning**

**Spring 2020**

**HW4**

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?

To maximize learning, do not use any pre-developed code or package to perform best subset or stepwise feature selection – for example, you don’t use not use the sklearn functions for feature selection

**Part A: Data setup & exploration**

1. Load, clean, split, explore, and transform the data to prepare it for machine learning.

* (Code is provided for this step) Using pandas, load the “ISLR\_Hitters.csv” dataset. Clean the data. Split into test (1/3) and non-test (2/3) datasets.
* (some code provided) 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.
* (Code is provided for this step) To prepare the X data for machine learning, prescale it (using sklearn standard scaler). Use only the non-test data to determine scaling parameters, but apply the scaling to both test and non-test X.
* (Code is provided for this step) Explore the response variable Y. Notice it is skewed. Transform it with a log transform. Be sure to handle the transformation when fitting models and computing MSE.

**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. **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 to run when k is large – for a model of size *k* you will need to fit and evaluate 2*k* models using 5 fold crossval for each fitting activity.)
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). Warning: when testing your code for errors, suggest setting the max k to 2 or 3… setting to 6 may run for many minutes. 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 (scatter) 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 fit the best models to 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, and you may use a package such as itertools to help manage your combinations of features.
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 set of points 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 point set 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?