HW5: Linear Model Selection

Programming assignment.

Directions.

- Comment the functions to receive full credit.
- Provide a single Python file with the format name_391_hw5.py, where name is your full name.
- Send your file to nehemyl@uw.edu and zhangkh@uw.edu.
- Please specify your name in the heading of the email. This assignment is due on Thursday, May 11th at 11:59pm PST.

Note: Throughout the assignment, you are allowed to use the LinearRegression class from scikit-learn.

(a) Write a function best_subset that implements the first two steps of Best Subset Selection. The function takes two input arguments: a two-dimensional Numpy array X of n rows and p columns (design matrix), and a one-dimensional Numpy array y of length n (vector of responses). best_subset returns two output arguments:

- a two-dimensional Numpy array of booleans var_in_model of p rows and p columns. More precisely, for k, j = 0, ..., p – 1, var_in_model[k, j] = True if the j-th predictor is included in the best model that contains exactly k predictors, and var_in_model[k, j] = False if it is not. Your code should contain at most two for or while loops;
- a one-dimensional Numpy array rss_model of length p+1 that contains the residual sum of squares of each best model.

(b) Write a function forward_stepwise that implements the first two steps of Forward Stepwise Selection. The function takes two input arguments: a two-dimensional Numpy array X of n rows and p columns (design matrix), and a one-dimensional Numpy array y of length n (vector of responses). best_subset returns two output arguments:

- a two-dimensional Numpy array of booleans var_in_model of p rows and p columns. More precisely, for k, j = 0, ..., p – 1, var_in_model[k, j] = True if the j-th predictor is included in the best model that contains exactly k predictors, and var_in_model[k, j] = False if it is not. Your code should contain at most two for or while loops;
• a one-dimensional NumPy array `rss_model` of length `p+1` that contains the residual sum of squares of each best model.

(c) Write a function `backward_stepwise` that implements the first two steps of Backward Stepwise Selection. The function takes two input arguments:
a two-dimensional NumPy array `X` of `n` rows and `p` columns (design matrix), and a one-dimensional NumPy array `y` of length `n` (vector of responses).

`best_subset` returns two output arguments

• a two-dimensional NumPy array of booleans `var_in_model` of `p` rows and `p` columns.
  More precisely, for `k, j = 0, ..., p - 1`, `var_in_model[k, j] = True` if the `j`-th predictor is included in the best model that contains exactly `k` predictors, and `var_in_model[k, j] = False` if it is not.
  Your code should contain at most two for or while loops;

• a one-dimensional NumPy array `rss_model` of length `p+1` that contains the residual sum of squares of each best model.

(d) Write a function `compute_Cp_bic_adjR2` that takes four input arguments: a one-dimensional NumPy array `y` of length `n` (vector of responses), an integer `d` (number of predictors in the considered model), `rss` (RSS for the considered model), and `var_err` (an estimate of the variance of the error term). The function returns three output arguments, namely $C_p$, BIC, and adjusted $R^2$ for the considered model.
 Directions. Show and explain all work to receive full credit. Homework is due on Thursday, May 11th at the beginning of class by 12:00 pm.

Problem 1. In this exercise, we will generate simulated data, and will then use this data to perform best subset selection. 
Show your computations by displaying the function calls.

(a) Use the the Numpy function `random.randn` to generate a predictor $X$ of length $n = 100$, as well as a vector noise $\varepsilon$ of length $n = 100$.

(b) Generate a response vector $Y$ of length $n = 100$ according to the model

$$ Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon, $$

where $\beta_0, \beta_1, \beta_2,$ and $\beta_3$ are constants of your choice.

(c) Perform best subset selection in order to choose the best model containing the predictors $X, X^2, \ldots, X^{10}$. What is the best model obtained according to $C_p$, BIC, and adjusted $R^2$? Show some plots to provide evidence for your answer, and report the coefficients of the best model obtained.

Note: To compute $C_p$ and BIC, you need an estimate of the variance of the error term. It is common to use the squared residual standard error, given by the formula: $\hat{\sigma}^2 = RSE^2 = RSS/(n - p - 1)$, where RSS is computed on the full model that contains all $p$ predictors.

(d) Repeat (c), using forward stepwise selection and also using backward stepwise selection. How does your answer compare to the results in (c)?

Problem 2. This problem relates to the College data set, which can be found in the file College.csv. It contains a number of variables for 777 different universities and colleges in the US. The variables are

- **Private**: Public/private indicator
- **Apps**: Number of applications received
- **Accept**: Number of applicants accepted
- **Enroll**: Number of new students enrolled
- **Top10perc**: New students from top 10
- **Top25perc**: New students from top 25
- **F.Undergrad**: Number of full-time undergraduates
- **P.Undergrad**: Number of part-time undergraduates
- **Outstate**: Out-of-state tuition
• **Room.Board**: Room and board costs
• **Books**: Estimated book costs
• **Personal**: Estimated personal spending
• **PhD**: Percent of faculty with Ph.D.'s
• **Terminal**: Percent of faculty with terminal degree
• **S.F.Ratio**: Student/faculty ratio
• **perc.alumni**: Percent of alumni who donate
• **Expend**: Instructional expenditure per student
• **Grad.Rate**: Graduation rate

In this problem, we will predict the number of applications received using the other variables.

Place the file in your working directory. You can read the data using the following commands:

```python
import pandas as pd

# College data set
college_data_raw = pd.read_csv('College.csv')
mapping = {'Yes': 1, 'No': 0}
college_data = college_data_raw.replace({'Private': mapping})

cols = [col for col in college_data.columns if col not in ['Unnamed: 0', 'Apps']]
X = college_data[cols]
y = college_data['Apps']
```

Show your computations by displaying the function calls.

(a) Use forward stepwise selection in order to choose the best model containing a subset of the predictors. What is the best model obtained according to 10-fold cross-validation? Show a plot to provide evidence for your answer.

*Note:* You are allowed to use the `model_selection` and `metrics` modules from `scikit-learn`. If you are using the class `cross_val_score`, specify appropriately the scoring parameter to obtain the mean squared error.

(b) Repeat (a) using backward stepwise selection.