BUFN 650 – Problem Set 1

Due on Wednesday, November 30 at 11:59 pm

Important: Please submit your homework using Canvas. Your submission needs to include **two** files: a PDF (or Word) document with all your responses **AND** a copy of your Python notebook (*.ipynb* Jupyter notebook file). To produce the latter, please click *File* → *Download .ipynb* in Google Colab, then save and upload the file on Canvas.

Each student has to submit his/her individual assignment and show all work. Legibly handwritten and scanned submissions are allowed, but they need to be submitted as a single document. Please do not submit photographs of pages in separate files.

Part I: Short-answer questions (80 points)

Please provide a **concise** answer for each of the questions below. Usually **one or two short sentences** should suffice. Do not write novels.

1. (12 points) For each of parts (a) through (d), indicate whether we would generally expect the performance of a flexible statistical learning method to be better or worse than an inflexible method. Justify your answer.

   (a) The sample size n is extremely large, and the number of predictors p is small.

   (b) The number of predictors p is extremely large, and the number of observations n is small.

   (c) The relationship between the predictors and response is highly non-linear.

   (d) The variance of the error terms, i.e. $\sigma^2 = \text{var}(\varepsilon)$, is extremely high.

2. (9 points) Explain whether each scenario is a classification or regression problem, and indicate whether we are most interested in inference or prediction. Finally, provide n and p.

   (a) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect CEO salary.

(b) We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.

(c) We are interested in predicting the % change in the USD/Euro exchange rate in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2019. For each week we record the % change in the USD/Euro, the % change in the US market, the % change in the British market, and the % change in the German market.

3. (12 points) I collect a set of data ($n = 100$ observations) containing a single predictor and a quantitative response. I then fit a linear regression model to the data, as well as a separate cubic regression, i.e.,

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon.$$

(a) Suppose that the true relationship between X and Y is linear, i.e.

$$Y = \beta_0 + \beta_1 X + \epsilon.$$

Consider the training residual sum of squares (RSS) for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

(b) Answer (a) using test rather than training RSS.

(c) Suppose that the true relationship between X and Y is not linear, but we don't know how far it is from linear. Consider the training RSS for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

(d) Answer (c) using test rather than training RSS.

4. (6 points) Consider the k-fold cross-validation.

(a) Briefly explain how k-fold cross-validation is implemented.

(b) What are the advantages and disadvantages of k-fold cross-validation relative to the validation set approach?

5. (3 points) Suppose that we use some statistical learning method to make a prediction for the response Y for a particular value of the predictor X. Carefully describe how we might estimate the standard deviation of our prediction.

6. (11 points) We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain $p+1$ models, containing $0, 1, 2, ..., p$ predictors. Explain your answers:

   (a) Which of the three models with k predictors has the smallest *training* RSS?

   (b) Which of the three models with k predictors has the smallest test RSS?

   (c) True or False (no explanation necessary; 1 point each):

      i. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by forward stepwise selection.

      ii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by backward stepwise selection.

      iii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by forward stepwise selection.

      iv. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by backward stepwise selection.

      v. The predictors in the k-variable model identified by best subset are a subset of the predictors in the $(k+1)$-variable model identified by best subset selection.

7. (12 points) The lasso, relative to least squares, is:

   (a) More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

   (b) More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

(c) Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

(d) Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

8. (15 points) Suppose we estimate the regression coefficients in a linear regression model by minimizing

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j}\right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p}|\beta_j| \le s,$$

for a particular value of s. For parts (a) through (e), indicate which of i. through v. is correct. Justify your answer.

(a) As we increase s from 0, the training RSS will:

i. Increase initially, and then eventually start decreasing in an inverted U shape.

ii. Decrease initially, and then eventually start increasing in a U shape.

iii. Steadily increase.

iv. Steadily decrease.

v. Remain constant.

(b) Repeat (a) for test RSS.

(c) Repeat (a) for variance.

(d) Repeat (a) for (squared) bias.

(e) Repeat (a) for the irreducible error.

Part II: Predict the number of applications received by colleges (120 points)

This exercise relates to the College data set, which can be found in the file *College* (*Hint:* the easiest way to import this file in Python is to copy the link and use the Panda's *read_csv()* function to read the file directly from this URL.) The file 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

Before reading the data into Python, it can be viewed in Excel or a text editor (if you download it).

1. (20 points) Exploring the data:

   (a) Use the *pd.read_csv('*`https://www.statlearning.com/s/College.csv`*')* function to read the data into Python. Call the loaded data *college*. Look at the data using the *college.head*() function. You should notice that the first column is just the name of each university. We don't really want Python to treat this as data. However, it may be handy to have these names for later. Set is as an index by passing an *index_col=0* parameter to the *read_csv()* call above. Alternatively, you may use the *college.set_index()* command. In the future, you can extract college names using *college.index*.

   (b) Use the *college.describe()* function to produce a numerical summary of the variables in the data set.

   (c) Import the *seaborn* package and alias it as *sns*. Use the *sns.pairplot()* function to produce a scatterplot matrix of the first five columns or variables of the data. Recall that you can reference the first five columns using *college.iloc[:,:5]*.

   (d) Use the *sns.boxplot(x=college['Private'], y=college['Outstate'])* function to produce side-by-side boxplots of *Outstate* versus *Private* (two plots side-by-side; one for each Yes/No value of *Private*).

   (e) Create a new qualitative variable, called *Elite*, by binning the *Top10perc* variable. We are going to divide universities into two groups based on whether or not the proportion of students coming from the top 10% of their high school classes exceeds 50%.

   Use the *sum()* function to see how many elite universities there are. Now use the *sns.boxplot()* function to produce side-by-side boxplots of *Outstate* versus *Elite*.

   (f) Use the *college.hist()* function to produce some histograms for a few of the quantitative variables. You may find parameters *bins=20,figsize=(15,10)* useful.

2. (100 points) Now, let's predict the number of applications received (variable *Apps*) using the other variables in the *College* data set:

   (a) (5 points) Replace any text variables with numeric dummies. You may use *pd.get_dummies(college, drop_first=True)* to achieve this.

   (b) (5 points) Construct response y (*Apps*) and predictors X (the rest of variables). You are worried that non-linearities in X could be important and decide to add all second-order terms to your predictors (i.e., $x_1, x_2, x_1x_2, x_1^2, x_2^2$ etc.). Add these terms to your X. Hint: you can use *PolynomialFeatures(include_bias=False).fit_transform(X)* function from *sklearn.preprocessing* package. If you did everything correctly, the set of variables in X should now be expanded from 18 to 189 features (all second-order terms, including interactions).

(c) (5 points) Split the data set into a training set and a test set. Never use the test set for anything but reporting the test error when asked below.

(d) (5 points) Standardize all explanatory variables (subtract their means and divide by standard deviation). Verify that all variables now have zero mean and unitary standard deviation.

(e) (10 points) Fit a linear model using least squares on the training set, and report the test error obtained. Warning: set *fit_intercept=True* for OLS and all other models below, if no intercept was included in your X (that is, if you set *include_bias=False* above).

(f) (15 points) Fit a ridge regression model on the training set, with λ chosen by cross-validation. Cross-validation should be performed using only the training set portion of the data in (a). Plot cross-validated MSE as a function of λ. Plot paths of coefficients as a function of λ. Report the test error obtained. *Hint:* I showed how to perform many of these steps in class in the *Chapter 6.ipynb* notebook.

(g) (15 points) Repeat (f) using lasso regression. You will likely receive convergence warnings or experience slowness. Use the original characteristics (with no second-order terms) if you do. Report the number of non-zero coefficients.

(h) (15 points) Repeat (f) using random forest. Recall that random forests and regression trees allow for interactions and non-linearities in X by design. Therefore, use the original set of characteristics here (with no second-order terms). Experiment with the *max_depth* parameter.

(i) (15 points) Fit an elastic net model on the training set, with λ chosen by cross-validation. Use the original characteristics. Report the test error obtained. *Hint:* Use *ElasticNetCV()* estimator from *sklearn.linear_model* to cross-validate and fit a model. You can read more *here*. Elastic net needs to cross-validate two parameters. You can do this automatically by adding *l1_ratio=np.linspace(.05, 1, 20)* as a parameter.

(j) (10 points) Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these five approaches?