ISyE 7406: Data Mining & Statistical Learning

HW#2 (due @Canvas @1:15pm on Thursday, January 30, 2020 for on-campus students)
There are 2 questions. The second question is a theoretical question that is required only to those PhD IE/statistics students, and is optional (no-credit) to all other students.

In this homework, please write your software codes (in R, Python, etc.) by yourself, and no collaborations allowed! It is **cheating** *if you copy and paste your classmates' codes.*

Problem 1 (R exercise for linear Regression. 50 points) Consider the data set "*fat*" in the "faraway" library of R. The data is also available at T-square or at

```
fat  <- read.table(file = "http://www.isye.gatech.edu/~ymei/7406/Handouts/fat.csv",
                   sep=",", header=TRUE);
```

The dataset *fat* has 252 observations and 18 variables, and, for more detailed description, see the link (`http://artax.karlin.mff.cuni.cz/r-help/library/faraway/html/fat.html`) or (`http://cran.r-project.org/web/packages/faraway/faraway.pdf`) (`page #37`).
For more background information, you can also see

`http://en.wikipedia.org/wiki/Body_fat_percentage`

The purpose of this homework is to help you better understand linear regression and R. Here we assume that the percentage of body fat using Brozek's equation (`brozek`, the first column) as the response variable, and the other 17 variables as potential predictors. We will use several different statistical methods to fit this dataset in the problem of predicting `brozek` using the other 17 potential predictors. For that purpose, it is useful to split it into the following sub-tasks.

(a) First, we should split the original data set into disjoint training and testing data sets, so that we can better evaluate and compare different models. One possible simple way is to random select a proportion, say, 10% of observations from the data for use as a **test** sample, and use the remaining data as a *training* sample building different models. **Note that in practice, it is more reasonable to select much larger proportion, say** 30% **or** 20%**, as testing sample**. Here we chose only 10% as the testing sample, so that we can list those testing observations explicitly below. You can do so by the following R code.

```
n = dim(fat)[1];      ### total number of observations
n1 = round(n/10);     ### number of observations randomly selected for testing data
set.seed(7406);   ### set the seed for randomization
flag = sort(sample(1:n, n1));
## If you are using other software, the 25 rows of testing observations are:
flag = c(1,  21,  22,  57,  70,  88,  91,  94, 121, 127, 149, 151, 159, 162,
         164, 177, 179, 194, 206, 214, 215, 221, 240, 241, 243);
fat1train = fat[-flag,];    fat1test  = fat[flag,];
```

(b) Second, for the training data "`fat1train`," do some **exploratory (or preliminary) data analysis** such as scatter plots or summary statistics of some variables that you feel are important (e.g., explain the unusual pattern).

(c) Based on the **training** data "`fat1train`," build the following models

(i) Linear regression with all predictors.

(ii) Linear regression with the best subset of $k = 5$ predictors variables;

(iii) Linear regression with variables (stepwise) selected using AIC;

(iv) Ridge regression;

(v) LASSO;

(vi) Principal component regression;

(vii) Partial least squares.

(d) Use the models you find in part (c) to predict the response in the **testing** data "`fat1test`" in part (a). Report the performance of each model $\hat{f}$ on the testing data, say, $\{(Y_i^{test}, \mathbf{x}_i^{test})\}_{i=1}^{n_1}$. Here $n_1 = 25$ and we assume that the performance of each model is evaluated by the following testing error

$$TE = \frac{1}{n_1}\sum_{i=1}^{n_1}[Y_i^{test} - \hat{f}(\mathbf{x}_i^{test})]^2.$$

(e) The above steps are sufficient when one has a large data set. However, for a relatively small data, one may want to do further to assess the robustness of each method. One general approach is **Monte Carlo Cross-Validation algorithm** that repeats the above computation B times ($B = 100$ say). That is, for each loop $b = 1, \ldots, B$, we randomly select, say $n_1 = 25$, observations from the original data as the testing data, and use the remaining data as a training sample. Within each loop, we first build different models from "*the training data of that specific loop*", and then evaluate their performances on "*the corresponding testing data.*" Therefore, for each model or method in part (c), we will obtain B values of testing errors on B different subsets of testing data, denote by TE_b for $b = 1, 2, \ldots, B$. Then the "average" performances of each model can be summarized by the sample mean and sample variables of these B TE values:

$$\overline{TE^*} = \frac{1}{B}\sum_{b=1}^{B} TE_b \qquad \text{and} \qquad \hat{Var}(TE) = \frac{1}{B-1}\sum_{b=1}^{B}\left(TE_b - \overline{TE^*}\right)^2.$$

Compute and compare the "average" performances of each model mentioned in part (c).

Write a report to summarize your findings. The report should include (i) **Introduction,** (ii) **Exploratory (or preliminary) Data Analysis** of training data in part (a), (iii) **Methods**, (iv) **Results** and (v) **Findings.** Also see the guidelines on the final report of our course project. Please attach your R code (without, or with limited, output) in the appendix of your report, and please do not just dump the R output in the body of the report.

Remark: In Part (c) and (e), please see the update R code for linear regression at Canvas. Note that in part (e), the same original data is repeatedly used B times as a whole, but it is used differently at different loops due to the different split of training and testing data. The idea of repeating the similar data analysis process B times is essential in many well-known statistical tools such as **bootstrapping** and **Random Forest**, and has been widely used in other fields such as **bioinformatics** or **computational biology**.

For your convenience, I also post some R codes at the pdf file of this homework at Canvas that might be useful. Please feel free to modify those R codes if you want. To encourage everyone learn the materials, each student must write their R or any other software codes by themselves, and no collaborations allowed! It is **cheating** if you copy and paste your classmates' computing codes.

Problem 2 (Overfitting and Underfitting. Only required to PhD IE/statistics students. Optional (no credit) to all other students. 10 points) Suppose that there is a data set with n observations, denoted by (y_i, x_{i1}, x_{i2}) for $i = 1, \ldots, n$, and assume that the true generative model for Y is $y_i = \beta_0^* + \beta_1^* x_{i1} + \epsilon_i$, where the errors are iid with $\mathbf{E}(\epsilon_i) = 0$ and $Var(\epsilon_i) = \sigma^2 > 0$.

Suppose that three ISyE 7406 students are given the data set (y_i, x_{i1}, x_{i2})'s, and are asked to predict Y when $X_1 = x_1^*$ and $X_2 = x_2^*$. However, the true model and the true regression coefficients β_0^* and $\beta_1^* (\neq 0)$ are unknown to the students. After independent thinking, the three students choose three different modeling strategies when analyzing the data:

- *Student A:* Fit the full model $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$. Denote by the corresponding ordinary least squares estimator by $(\hat{\beta}_{0A}, \hat{\beta}_{1A}, \hat{\beta}_{2A})$ and then predict the Y value by $\hat{Y}_A = \hat{\beta}_{0A} + \hat{\beta}_{1A} x_1^* + \hat{\beta}_{2A} x_2^*$.
- *Student B:* Fit a subset model $y_i = \beta_0 + \beta_1 x_{i1} + \epsilon_i$. Denote by the corresponding ordinary least squares estimator by $(\hat{\beta}_{0B}, \hat{\beta}_{1B})$ and then predict the Y value by $\hat{Y}_B = \hat{\beta}_{0B} + \hat{\beta}_{1B} x_1^*$.
- *Student C:* Fit a subset model with only constant term: $y_i = \beta_0 + \epsilon_i$. Denote by the corresponding ordinary least squares estimator by $\hat{\beta}_{0C}$ and then predict the Y value by $\hat{Y}_C = \hat{\beta}_{0C}$.

(a) Show that for the model used by *Student B*, for any given (x_1^*, x_2^*),

$$\mathbf{E}(\hat{Y}_B) = \mathbf{E}\Big(\hat{\beta}_{0B} + \hat{\beta}_{1B} x_1^*\Big) = \beta_0^* + \beta_1^* x_1^*$$

and

$$Var(\hat{Y}_B) = \Big(\frac{1}{n} + \frac{(x_1^* - \bar{x}_1)^2}{\sum_{i=1}^n (x_{i1} - \bar{x}_1)^2}\Big)\sigma^2, \qquad \text{where } \bar{x}_1 = \textstyle\sum_{i=1}^n x_{i1}/n.$$

(b) As mentioned in class, one possible consequence of overfitting (e.g., including irrelevant variables in a regression model) is that the predictions are not efficient, i.e., have larger variances, although they are unbiased. Show that for any given (x_1^*, x_2^*),

$$\mathbf{E}(\hat{Y}_A) = \mathbf{E}\Big(\beta_{0A} + \beta_{1A} x_1^* + \beta_{2A} x_2^*\Big) = \beta_0^* + \beta_1^* x_1^*$$

and

$$Var(\hat{Y}_A) \geq Var(\hat{Y}_B).$$

Hints: let $\mathbf{X}_{[0]} = (1, 1, \cdots, 1)^t, \mathbf{X}_{[1]} = (x_{11}, x_{21}, \cdots, x_{n1})^t$ and $\mathbf{X}_{[2]} = (x_{12}, x_{22}, \cdots, x_{n2})^t$ be the column vectors of the matrix $\mathbf{X}_{n\times 3}$, where u^t denotes the transpose. It is useful to think the projection, and begin with the special case when $\mathbf{X}_{[2]}$ is orthogonal to both $\mathbf{X}_{[0]}$ and $\mathbf{X}_{[1]}$, i.e., $\mathbf{X}_{[0]}^t \mathbf{X}_{[2]} = 0$ and $\mathbf{X}_{[1]}^t \mathbf{X}_{[2]} = 0$. For the general case, we can consider the projection of $\mathbf{X}_{[2]}$ onto the plane spanned by $\mathbf{X}_{[0]}$ and $\mathbf{X}_{[1]}$, e.g., $\hat{\mathbf{X}}_{[2]} = \gamma_0 \mathbf{X}_{[0]} + \gamma_1 \mathbf{X}_{[1]}$. Then $U_{[2]} = \mathbf{X}_{[2]} - \mathbf{X}_{\hat{[2]}}$ is orthogonal to both $X_{[0]}$ and $X_{[1]}$. Now consider the linear regression of $\mathbf{Y}$ on the new data set $\mathbf{X}_{[0]}$, $\mathbf{X}_{[1]}$ and $\mathbf{U}_{[2]}$, which should yield the same prediction $\hat{Y}_A$.

(c) One consequence of underfitting (e.g., excluding important variables in a regression model) is that the predictions are biased, though they have smaller variables. Show that when $x_1^* \neq \bar{x}_1$,

$$\mathbf{E}(\hat{Y}_c) = \mathbf{E}\Big(\beta_{0C}\Big) \neq \beta_0^* + \beta_1^* x_1^*$$

and $Var(\hat{Y}_C) < Var(\hat{Y}_B)$.

Appendix: the following R code might be useful for Problem 1 of Homework #2:

```
### Read the data
fat  <- read.table(file = "http://www.isye.gatech.edu/~ymei/7406/Handouts/fat.csv",
                     sep=",", header=TRUE);
### Split the data as in Part (a)
n = dim(fat)[1];      ### total number of observations
n1 = round(n/10);     ### number of observations randomly selected for testing data
set.seed(7406);   ### set the seed for randomization
flag = sort(sample(1:n, n1));
## If you are using other software, the 25 rows of testing observations are:
flag = c(1,  21,  22,  57,  70,  88,  91,  94, 121, 127, 149, 151, 159, 162,
         164, 177, 179, 194, 206, 214, 215, 221, 240, 241, 243);
fat1train = fat[-flag,];    fat1test  = fat[flag,];

###In Part (b)-(d), Please see the update R code for linear regression at T-square.
### Please write your own R or other software code to analyze the training data "fat1train"
### and evaluate different models on the testing data "fat1test".

### Part (e): the following R code might be useful, and feel free to modify it.
###    save the TE values for all models in all $B=100$ loops
B= 100;            ### number of loops
TEALL = NULL;      ### Final TE values

for (b in 1:B){
  ### randomly select 25 observations as testing data in each loop
  flag <- sort(sample(1:n, n1));
  fattrain <- fat[-flag,];
  fattest  <- fat[flag,];

  ### you can write your own R code here to first fit each model to "fattrain"
  ### then get the testing error (TE) values on the testing data "fattest"
  ### Suppose that you save the TE values for these five models as
  ###   te1, te2, te3, te4, te5, te6, te7, respectively, within this loop
  ###   Then you can save these 5 Testing Error values by using the R code
  ###
  TEALL = rbind( TEALL, cbind(te1, te2, te3, te4, te5, te6, te7) );
}
dim(TEALL);  ### This should be a Bx7 matrices
### if you want, you can change the column name of TEALL
colnames(TEALL) <- c("mod1", "mod2", "mod3", "mod4", "mod5", "mod6", "mod7");

## You can report the sample mean and sample variances for the five models
apply(TEALL, 2, mean);
apply(TEALL, 2, var);
### END ###
```