15.071: The Analytics Edge **Spring 2020**
Homework 5: Regularization, Boosting, and Clustering

Out: March 30; Due: April 10, 5:00 pm.
Please post the assignment in pdf format with file name "Lastname_15071_HW5.pdf".
For each question, please include the main `R` *commands that you used in your submission.*

Problem 1: Moneyball Analytics [75 pts]

Sport Analytics started with—and were popularized by—the data-driven approach to player assessment and team formation of the Oakland A's. In the 1990s, the Oakland A's were one of the poorest teams in the Major Baseball League. Player selection was mainly done through *scouting*: baseball experts would watch high school and college games to identify future talent. Under Billy Beane's and Paul DePodesta's leadership, the Oakland A's started to use data to detect undervalued players. Quickly, they met success on the field, reaching the playoffs in 2002 and 2003 despite a much lower payroll than their competitors. This started a revolution in sports: analytics have now become a central component of every team's strategy.[1]

In this problem, you will predict the salary of baseball players. You will work with the dataset provided in the `Hitters.csv` file, which contains information on 263 players from the Major League Baseball (MLB) in 1986. The first column reports the player names. The second column reports the player annual salaries (in $'000), which we aim to predict. The other variables report four sets of variables: offensive statistics during the season, offensive statistics over the player's career, defensive statistics, and team information.[2]

- Name: the player's name
- Salary: the player's annual salary (in $'000)
- AtBat: Number of times at bat in the season
- Hits: Number of hits in the season
- HmRun: Number of home runs in the season
- Runs: Number of runs in the season
- RBI: Number runs enabled in the season
- Walks: Number of walks in the season
- Years: Number of years played in MLB
- CAtBat: Number of times at bat in the career
- CHits: Number of hits in the career
- CHmRun: Number of home runs in the career
- CRuns: Number of runs in the career
- CRBI: Number runs enabled in the career
- CWalks: Number of walks in the career
- PutOuts: Number of putouts in the season
- Assists: Number of assists in the season
- Errors: Number of errors in the season
- League: League in which team plays
- Division: League in which team plays
- NewLeague: League in which team plays next year

First, import the dataset:

```
Hitters_raw <- read.csv("Hitters.csv")
```

a) Explore the dataset using the following steps.

i) Report the correlation matrix between the numerical predictors (i.e., all predictors except Name, League, Division and NewLeague). You can restrict the dataset to the numerical predictors with `Hitters_raw[,2:18]`. What do you observe? **[5 pts]**
[**Hint:** For a nice visualization of the correlation matrix, consider using the function `ggcorrplot`.]

ii) Plot the player salaries as function of each one of 2-3 predictors that you consider important (based on question a) part *i*)). How does a change in each predictor affect the salary? **[5 pts]**

b) Feature normalization is often important in regularized regression problems. Normalize the data, and split them into a training set and a test set, using the following commands:

[1] For more details, see the *Moneyball: The Art of Winning an Unfair Game* book by Michael Lewis and the *Moneyball* film.
[2] No knowledge of baseball is required to complete this problem!

```
pp <- preProcess(Hitters_raw, method=c("center", "scale"))
Hitters <- predict(pp, Hitters_raw)
set.seed(15071)
train.obs <- sort(sample(seq_len(nrow(Hitters)), 0.7*nrow(Hitters)))
train <- Hitters[train.obs,2:21]
test <- Hitters[-train.obs,2:21]
```

The first models you will try are simple linear regression models.

i) Fit a linear regression model with all the predictors using the training set, and make predictions on the test set. Report the in-sample and out-of-sample R^2. Comment briefly on the sign and significance of the variables and the R^2 values. Does this make sense, in view of your earlier observations? **[5 pts]**

ii) Fit a restricted linear regression model by manually selecting only the predictors that are significant in the full model. (We will consider a variable significant only if the p-value is below 0.1.) Report the in-sample and out-of-sample R^2. **[5 pts]**

c) Let's now examine if regularization can help us do better.

i) Train ridge regression and LASSO models with 10-fold cross-validation to select the appropriate value of the shrinkage parameter λ, using the Mean Squared Error as the performance metric (these are the default options in the `cv.glmnet()` function). Plot the cross-validated Mean Squared Error as a function of λ. Report the value of λ that minimizes the Mean Squared Error for each method. **[10 pts]**

[**Hint**: Input the values of λ manually to ensure that it covers a wide enough range. You can use:

```
all.lambdas <- c(exp(seq(15, -10, -.1)))
```

Then, you can use the `lambda=all.lambdas` option in `cv.glmnet()`.]

ii) With the selected values of λ, re-train your ridge regression and LASSO models on the full training set. Report each model's coefficients and comment on the effects of ridge regression vs. LASSO. Use each model to make predictions on the test set. Report the values of the in-sample R^2 and the out-of-sample R^2. Comment on your results. **[10 pts]**

iii) As we discussed in class, LASSO often results in models where many coefficients are set to 0. Is this the case with the LASSO model you trained in question c) part *ii*)? **[5 pts]**

d) Let's try to improve the out-of-sample performance of our model by performing explicit variable selection. Let `nvars` be the number of predictors you manually selected in question b) part *ii*).

i) Select the value of λ that led to the best (in terms of validation error) LASSO model consisting of at most `nvars` nonzero coefficients. You can do that using the following commands (provided that you used the `cv.glmnet()` function to run cross validation for LASSO):

```
all_constrained_validation_runs <- lasso.cv$nzero <= nvars
best_constrained_run <- which.min(lasso.cv$cvm[all_constrained_validation_runs])
best_constrained_lambda <- lasso.cv$lambda[best_constrained_run]
```

With the selected value of λ, re-train your LASSO model on the full training set and make predictions on the test set. Report the values of the in-sample R^2 and the out-of-sample R^2. Did you manage to get a higher out-of-sample R^2? Compare the predictors chosen by your model with the ones you manually selected in question b) part *ii*). **[10 pts]**

ii) Recall that best subset selection provides a more natural way to include a subset of the variables in our model. Use forward stepwise selection with 10-fold cross-validation to select the best value of the subset size parameter (between 1 and 15). To do that, you may use the `train` function from the `caret` package, as follows:

```
fs <- train(Salary∼., train,
        method = "leapForward",
        trControl = trainControl(method = "cv", number = 10),
        tuneGrid = expand.grid (.nvmax=seq(1,15))
```

What is the size of the best subset you found? Which variables are contained therein (use `summary(fs)` to see this)? Does this result agree with the selection of variables you found in question b) part *ii*) and in question d) part *i*)? Use the final model to make predictions on the test set and report the values of the in-sample R^2 and the out-of-sample R^2. **[10 pts]**

e) Finally, we'll use the state-of-the-art XGBoost framework. Use 5-fold cross-validation to tune an XGBoost regressor. Report the best values for the hyperparameters that you cross-validated. Use the final model (trained on the full training set) to make predictions on the test set and report the values of the in-sample R^2 and the out-of-sample R^2. Comment on your results. **[10 pts]**

Problem 2: Clustering Stock Returns [50 pts]

When building portfolios of stocks, investors seek to obtain good returns while limiting their variability. This can be achieved by selecting stocks that show different patterns of returns—a technique known as *diversification*. To support these decisions, we will identify clusters of stocks that exhibit similar patters.

For this problem, we will use the dataset `returns.csv`. This file contains monthly returns from some stocks among the S&P500 from March 2006 through February 2016. Each observation (row) corresponds to a company. The variables in the dataset are described in Table 1.

Table 1: Variables in the dataset `returns_final.csv`.

Variable	Description
`symbol`	The symbol identifying the company of the stock.
`Industry`	The industry sector under which the stock is classified.
`avg200603 - avg201602`	The return for the stock during the variable's indicated month. The variable names have format "avgYYYYMM", where YYYY is the year and MM is the month. For instance, variable avg200902 refers to February 2009. The value stored is a net increase or decrease of the end of month stock price over the stock price at the beginning of the month. For instance, a value of 0.05 means the stock had a net increase on average of 5% during the month, while a value of -0.02 means the stock had a net decrease on average of 2% during the month. There are 120 of these variables, for the 120 months in our dataset.

Import the dataset using the following command.

```
data = read.csv("returns.csv")
```

Note that stock returns are provided in Columns 3 through 122. You may find it useful to create a second dataset with only these values.

```
returns = data[,3:122]
```

a) How many companies are there in each industry sector? Entering the "Great Recession" of 2008-2009, most stocks lost significant value, but some sectors were hit harder than others. For each sector, plot

the average stock return between 01/2008 and 12/2010. In September 2008, which industries had the worst average return across the stocks in that industry? Is industry information sufficient for investors to build a portfolio diversification strategy? **[10 pts]**

[**Hint**: To average quantities by industry, consider using the `aggregate` function.]

b) Let us now cluster the stocks according to their monthly returns. In this analysis, we will not normalize our data prior to clustering. Why is this a valid approach for this problem and dataset? Cluster the data using hierarchical clustering, using the "Ward D2" measure of cluster dissimilarity. Plot the dendrogram and the scree plot. What do you think is a reasonable number of clusters for this problem? Justify your choice. **[10 pts]**

c) Use the `cutree` function to cut the dendrogram tree to the desired number of clusters and extract cluster assignments. Compute the number of companies in each cluster and the number of companies per industry sector in each cluster. What are the average returns by cluster in October 2008 and in March 2009? Characterize each cluster qualitatively based on these results, and any other relevant information. **[10 pts]**

[**Hint**: To compute the number of companies in each cluster and the number of companies per industry sector in each cluster, consider using the `table` function.]

d) Cluster the data using the k-means clustering algorithm, using the same number of clusters as in Question c. What are the cluster centroids in October 2008 and in March 2009? Extract the cluster assignments and compare them to those obtained with hierarchical clustering. In what ways are the clusters similar vs. different across the two models? **[10 pts]**

[**Hint**: To answer this question, you should look at the number of observations in each cluster and the industry sectors of the companies in each cluster. Do some k-means clusters match a hierarchical cluster? Do some k-means clusters look very different from every one of the hierarchical clusters?]

e) The **silhouette metric** measures how similar an observation is to its own cluster compared to other clusters; this is done by comparing the distance from the observation to other observations in its cluster with the distance from the observation to other observations in the second closest cluster. More precisely, the silhouette metric for observation i is computed as

$$s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))},$$

where $a(i)$ is the average distance from observation i to the other points in its cluster, and $b(i)$ is the average distance from observation i to the points in the second closest cluster. This score ranges from -1 to 1 and a higher score is better. Observe that the silhouette metric is computed individually for each observation in the data - these individual scores can be averaged to reflect the quality of the global assignment.

Use the function `silhouette(cluster_assignment,distances)` to compute the silhouette scores that correspond to the assignments obtained via hierarchical clustering (question c)) and via k-means clustering (question d)). The first argument of the method is a cluster assignment (e.g., the `$cluster` component of the object returned by the `kmeans` function) and the second argument is the matrix of pairwise distances between all data points. Report the mean silhouette score of each assignment, both per cluster (using the `aggregate` function) and overall. Plot the results using `plot(cluster_assignment, col=1:selected.k, border=NA)`, where `selected.k` is the selected number of clusters. **[5 pts]**

[**Hint**: Run `?silhouette` to see exactly what is returned by the `silhouette` function.]

f) Write a short paragraph to an investor describing your model and how it can be used to inform investment strategies. **[5 pts]**