16-791 Applied Data Science – Spring 2018, Section A

Homework #3

Reports are due at 12:00pm (noon) on Monday, February 26.
Please upload your report and code on canvas.
Please remember to properly format for printing all documents submitted electronically.
Individual work will be strictly enforced!

R packages: zoo – time series analysis**,** plyr – data manipulation **,** MASS – stepwise regression**,** leaps – exhaustive search for subsets of features
Python libraries: scikit-learn, numpy, scipy, matplotlib, pandas

Problem 1
In this problem, we apply Principal Component Analysis to stock market index data. A few useful functions for this sort of analysis are listed below.

```
#R
prcomp(x, retx = TRUE, center = TRUE, scale = FALSE)
x: data frame; retx: return the projected x? ; center: center x beforehand? ;
scale: scale x? – if scale is TRUE, then PCA operates on the correlation matrix (instead of covariance)
```

```
#Python
# use sklearn.decomposition.PCA
# sklearn uses covariance matrix. To use correlation matrix, we would do:
Xstd = StandardScaler().fit_transform(X)
pca.fit_transform(Xstd)
```

We will use record of daily closing prices of S&P 500 stocks from January 1, 2011 to December 31, 2014 retrieved through Yahoo Finance. The data is stored in the attached file named "SP500_close_price.csv". There are actually only 471 stocks in this file, the rest of the stocks were not included either because Yahoo Finance returned an error or because the stock was not listed as of January 1, 2011. The file named "SP500_ticker.csv" contains ticker information for each included stock, as well as the corresponding company name and its industry sector assignment. The R template file contains code to retrieve raw data. Feel free to try it out if you are curious. But you do not have to do so.

Items to address:

a) Fit a PCA model to log returns derived from stock price data. The code for deriving log returns is provided in the template files. Having built the model, please do the following:
    1. Plot a scree plot which shows the distribution of variance contained in subsequent principal components sorted by their eigenvalues.

2. Create a second plot showing cumulative variance retained if top N components are kept after dimensionality reduction (i.e. the horizontal axis will show the number of components kept, the vertical axis will show the cumulative percentage of variance retained).
3. How many principal components must be retained in order to capture at least 80% of the total variance in data?
4. What is the magnitude of the estimated reconstruction error if we only retain top two of the PCA components?

b) Analysis of principal components and weights

1. Compute and plot the time series of the 1st principal component and observe temporal patterns. Identify the date with the lowest value for this component and conduct a quick research on the Internet to see if you can identify event(s) that might explain the observed behavior.
2. Extract the weights from PCA model for 1st and 2nd principal components.
3. Create a plot to show weights of the 1st principal component grouped by the industry sector (for example, you may draw a bar plot of mean weight per sector). Observe the distribution of weights (magnitudes, signs). Based on your observation, what kind of information do you think the 1st principal component might have captured?
4. Make a similar plot for the 2nd principal component. What kind of information do you think does this component reveal? (Hint: look at the signs and magnitudes.)
5. Suppose we wanted to construct a new stock index using one principal component to track the overall market tendencies. Which of the two components would you prefer to use for this purpose, the 1st or the 2nd? Why?

Problem 2

a) Let us experiment with a few feature selection methods. We will use data stored in "BMI.csv" file. This data contains measurements of Body Mass Index (BMI) obtained for a number of human subjects. The goal is to predict fat percentage (fatpctg) using all other features available in data. For items (b) and (c) please refer to a short R tutorial available here: http://www.statmethods.net/stats/regression.html. For python, check out http://scikit-learn.org/stable/modules/feature_selection.html.

b) Correlation based feature selection. Write a function called filter.features.by.cor which takes a data frame as input (assume that the last column is the output dimension). This function will screen the available input features in a loop and compute the absolute values of linear correlation coefficients between each input and the output. The function will output a sorted data frame containing the names of features and the corresponding coefficients. Execute the function with the BMI data and identify the top 3 features with respect to the correlation metric.

c) Subset selection

a. **In R**:Using 'leaps' package execute exhaustive search for the best subsets of features in BMI data. Identify the best sets of features of size 1, 2 and 3. (Hint: nbest, nvmax are relevant parameters.) Comment on why this is different from univariate feature selection and what the implications are.

b. **In python**: Unfortunately, the python packages we work with (currently) do not have an equivalent package compared to 'leaps'. You have 2 options:

   i. **Either** you use RPy2 and call the R 'leaps' function from within python and then do exactly what question b) asks you to do in R. (This is certainly an interesting skill to learn since there are many R packages that do not have an equivalent in python. For example the fisher exact test in R is much more sophisticated than the one in scipy.)

   ii. **Or** you use Univariate feature selection in sklearn.feature_selection.SelectKBest together with a reasonable score function (e.g. f_regression) to select feature sets of size 1, 2 and 3. Comment on why this is different to the exhaustive subset search and what the implications are.

d) Search for the best model using backward stepwise regression. Which variables are included in the model? **R:** Use function stepAIC from the MASS package. **Python:** Use sklearn.feature_selection.RFE or sklearn.feature_selection.RFECV in combination with a linear regression model or linear support vector regression.

Submission Guidelines

Your submission should two components. Your code (txt or notebook+html notebook) and your write-up (PDF):

I. Start all filenames with YourAndrewID_HW3
   a. If you submit a. R or .py file, **change the file extension to *.txt in order for it to be acceptable to the TurnitIn system.**
   b. If you submit a python or R notebook (.rmd or .ipynb) ALSO SUBMIT AN **HTML** VERSION of the notebook

   Note that we will execute your code to test it.

II. A pdf file named YourAndrewID_HW3.pdf containing summaries of obtained results.