Analytics 512: Take Home Final Exam

200 points in seven problems. This is the take-home portion of the exam. You may use your notes, your books, all material on the course website, and your computer or any computer in the departmental computer lab. You may also use official documentation for `R`, built-in or on https://cran.r-project.org/, but no other material on the Internet. Provide proper attribution for all such sources. You may not use any human help, except whatever help is provided by me.

Return your solutions by Thursday, 5/11/17, 11:59PM

- by e-mail **as a single .RMD file together with the resulting .pdf file**
- or hand in printed copies of both files
- or fax both files to 202.687.6067.

The .Rmd file should load all data and all packages, make all plots, and contain all comments and explanation. Set the seed to the year in which you maternal grandmother was born.

Problems 1 - 4 use the **Vehicle** data that are available in the `mlbench` package. Problems 5 - 7 use the 'mlbench.smiley' function in the same package which makes artificial data set. Be sure to read the package documentation and the data set descriptions.

Vehicle data

Load the package and make the data available:

```r
require(mlbench)
data(Vehicle)
```

1. (20) Use a decision tree to predict the vehicle class. Assess your prediction accuracy using a confusion matrix, once from the full data set and also with 10-fold cross validation.
2. (40) Now use PCA to find the principal components of the matrix of predictors, scaling these predictors so that all have unit variance before performing the PCA. It is possible to use the first k principal components to predict the vehicle class with a tree. Use 10-fold cross validation to select the best k. Assess the error using the misclassification rate.
3. (40) Now use a random forest with the original dataset (not principal components). Predict the vehicle class with a random forest model where trees with up to m terminal nodes are used. The parameter m can be set with the *maxnodes* parameter. Use 10-fold cross validation to find the best m. Assess the error using the misclassification rate.
4. (40) It turns out that the two classes **Opel** and **Saab** are difficult to separate. Make a subset of vehicle data of Saabs and Opels only and predict the vehicle class using logistic regression and five predictors. You have to find a way to choose five predictors and explain your reasoning. *Possible approaches for choosing predictors: run logistic regression for the whole set of predictors and discard the non-significant ones; or use regsubsets (choosing one of the possible methods); or compute correlations between all predictors and eliminate those that are highly correlated; or use regularized regression such as LASSO; or use a pruned tree.* Evaluate the final model of your choice using 10-fold cross validation. Summarize the results of this exploration in a paragraph, using tables or graphs as you see fit.

Smiley data

Read the instructions for the `mlbench.smiley()` function.

5. (20) Generate four different smiley data sets with $n = 500$ points each that have different values of standard deviations for the eyes and the mouth (these are the *sd1* and *sd2* arguments). Plot them with colors given by the target class labels.
6. (20) Next, create a series of Smiley data sets in the following way: use $sd1 = .1$ and let $sd2$ range from 0.05 to 0.5. For each of these, use k-means clustering with $k = 4$ to cluster smiley data. Explore for which values of $sd2$ the clusters coincide (more or less) with the target class labels. Summarize the result of this exploration in a paragraph or two, using tables or graphs as you see fit.
7. (20) With the same series of Smiley data sets, use hierarchical clustering, followed by cutting the tree to $k = 4$ clusters for the smiley data. Use a linkage method of your choice. Explore for which values of $sd2$ the clusters coincide (more or less) with the target class labels. Summarize the result of this exploration in a paragraph or two, using tables or graphs as you see fit.