Course 3 Task 3 Project Instructions

**[2. Optional Task - Feature Selection](https://ut.daacertificate.com/mc/poa?productID=2654&taskID=3346" \l "collapsepoa4324)**

Although optional, it is highly recommended that you work through this step. The matrix files contain 57 attributes. Taking the time to understand how the features are related can save you time later on.

While there are some models that thrive on correlated attributes, other models may benefit from reducing the level of correlation between the attributes.

Before exploring the methods below, take some time to visualize the relationships between the attributes.

1. Use library with the corrplot package. (install this package if you do not have it.)
2. Create a heat map: **corrplot(cor(df), order ="hclust")**
3. Examine the resulting plot.

**Method One**

A popular method of feature selection is using a decision tree algorithm to determine the attributes that are most useful to the dataset. We can use the caret pipeline to do this.

1. Create fitControl object using the caret trainControl command
2. Build a random forest model using the caret train command
3. Use the caret “predictors” command. This command returns a list of attributes that were used to understand the y-variable.
4. Consider removing any attribute not on this list. This can help reduce noise in your data set.

**Method Two**

The caret package has a function called findCorrelation which evaluates the relationships within the dataset and creates a list of highly correlated predictors for elimination. Visit the [caret preprocessing page](http://topepo.github.io/caret/pre-processing.html) for more information.

1. Create an object containing correlation results: **descrCor <- cor(df)**
2. Investigate the summary statistics of your new object. What is the range of correlation values? Is there anything that might cause overfit? **summary(descrCor[upper.tri(descrCor)])**
3. Use findCorrelation to create a list of highly correlated predictors. Specify a cut off point for your definition of "highly". Feel free to experiment, but we'll use .80 for this example: **highlyCorDescr <- findCorrelation(descrCor, cutoff = .80)**
4. **Summary(highlyCorDescr)** will give you the list.
5. To create a new dataset the excludes the list: **newdf <- df[, -highlyCorDescr]**

**[3. Preprocess the Large Matrices](https://ut.daacertificate.com/mc/poa?productID=2654&taskID=3346" \l "collapsepoa3271)**

Note: This step must be done for both the iPhone and Galaxy - some of the variable names will differ.

Upon import the y-variable is an integer class with a negative to positive range. Ideally you should convert the y-variable to a factor by discretizing this range into several levels that could convey sentiment to a business audience. For example, 7 levels (very negative, negative, somewhat negative, neutral (neither negative nor positive), somewhat positive, positive, very positive).

There are many packages available for discretization, but we'll focus on the [arules package](https://cran.r-project.org/web/packages/arules/index.html) and its [discretize function](http://finzi.psych.upenn.edu/library/arules/html/discretize.html).

1. Install the arules package and use the library function.
2. Think about the range of the y-variable. Negative numbers represent negative sentiment. Positive numbers, positive sentiment. Zero represents no sentiment or neutral. We will used "fixed" as an argument in our command and specify the range for each factor level by naming the "breaks".
3. From y-variable iphoneSentiment data, create a vector containing discretized levels. **disfixed7 <- discretize(df$iphoneSentiment, "fixed", categories= c(-Inf, -50, -10, -1, 1, 10, 50, Inf))**
4. Inspect the vector by using summary() and str(). You should find counts for each of the 7 factor levels.
5. If you are satisfied with the distribution of counts, insert the vector into your data set. **df$iphoneSentiment <- disfixed7**
6. Optional: Explore other numbers of levels. It is not recommended that you choose less than 7 levels. Doing so will result in the loss of useful sentiment information.

#### [4. Collect Sentiment Counts](https://ut.daacertificate.com/mc/poa?productID=2654&taskID=3346" \l "collapsepoa4325)

Note: This step must be done for both the iPhone and Galaxy - some of the variable names will differ.

1. Now that you have changed the y-variable to a factor with x-levels, you can get a useful information for your report. Use the summary command with iphoneSentiment and note the number of instances represented in each level. Recall the 7 level example in step 3: very negative, negative, somewhat negative, neutral (neither negative nor positive), somewhat positive, positive, very positive.   
     
   Note all of these counts for your report.

**[5. Model Development and Optimization](https://ut.daacertificate.com/mc/poa?productID=2654&taskID=3346" \l "collapsepoa4326)**

The sentiment count distribution collected in the last step is a huge component of our client's requirements for this project. Now you need to develop models that understand the data and can be pointed at new data should the client revisit this project.

1. Revisit the Caret Work Flow posted on Piazza and available in C2/T3.
2. Use your preprocessed (and potentially feature selected) iPhoneLargeMatrix dataset
3. Set your seed.
4. Consider sampling your data for computational efficiency. Try a sample of 4000 to start with.
5. Create training and testing sets with a 70/30 split using caret's createDataPartition function. CreateDataPartition helps ensure an equal distribution of factor levels between train and test sets.
6. Train four algorithms and choose the best performing model. Model codes for caret can be found here: <http://topepo.github.io/caret/available-models.html>
   * C5.0
   * Random Forest
   * KNN
   * SVM
   * Feel free to experiment with other algorithms
7. Use your best performing model to make predictions with your test set.
8. Use postResample to confirm accuracy. postResample can quickly check your predicted values against the actual values in your test set. Note the accuracy for your report.
9. **Repeat the process starting with step one for the GalaxyLargeMatrix.**

**Optional Step:** Ask your mentor for additional data and make predictions using your optimized model. This will allow you to embellish your report by expanding the sentiment counts.

#### [6. Analyze Results and Write Up Findings Report](https://ut.daacertificate.com/mc/poa?productID=2654&taskID=3346" \l "collapsepoa4327)

1. Based on your classifier performance and postResample, what is the expected result if you used your most optimized model on unseen data?
2. **Explore and note observations.**Consider the comparative frequency and distribution of the ratings for iPhone and Galaxy. Ask yourself the following questions to help you get started on your interpretation of the results:
   1. Are the distributions of the ratings clustered at one end of the scale or the other, or are they evenly distributed? What can this tell you about the polarity of the attitudes toward the given handset (e.g., do reviewers either hate it or love it? Or are most reviewers neutral?)?
   2. How does the frequency and distribution of the ratings of one handset compare to the other?
   3. Are there any anomalies that strike you? How would you describe them?
3. What is the best way to compare iPhone and Galaxy sentiment results if your data sets have different numbers of instances? How can this be done visually?

Tip: Visual representations of the data may help you draw conclusions more quickly. The visualization process is about taking numeric data and turning it into something we can recognize intuitively. Try plotting your data in Excel in a variety of ways until you begin to see patterns and trends. Don't be afraid to experiment!

1. Construct the story of the data. Review your observations and draw conclusions, and then organize and prioritize your conclusions into a coherent narrative that compares the handsets. Consider the following questions as you go through this process:
   1. Consider how these conclusions relate to your client’s goals. Which of the conclusions will have the most value to your client? For example, how would strong preferences impact your client’s decision about which handset to bundle their software with? What if no strong preferences exist?
   2. How confident do you feel about your conclusions? Are some of the conclusions more strongly supported by the data than others?
   3. Do any of your conclusions conflict with each other? How can that be explained?

Trap: Beware of playing up small differences in the results. They may disappear in a larger or different sample of data.

1. Illustrate your narrative. Consider using Excel and other appropriate software to construct and include charts or graphs to emphasize and support your conclusions. Your audience should be able to interpret your chart without help from you. Make sure it is relevant and designed to get across your points.
2. Write up your Summary of Findings. Start by outlining your document or presentation. You can organize the document in the way you feel is the most appropriate for a client audience. Be sure to include:
   1. Your narrative of the data supported by the results.
   2. How confident you are in the results . This should contain three parts: 1) The reported error metrics from R 2) Your personal sense of how well the attributes we are measuring will actually capture pages that have relevant sentiment. Don’t be afraid to express your opinion here, but you must justify your statements with reason and examples. 3) Caveats of where you think this analysis process might not be capturing the sentiment accurately and suggestions for how to do better in the next round of analysis.
   3. What implications your narrative has to the client’s goals.
   4. High-level explanation of what you did.
3. Submit your Summary of Findings

#### [7. Write Lessons Learned Report](https://ut.daacertificate.com/mc/poa?productID=2654&taskID=3346" \l "collapsepoa4328)

Your lessons learned report should be no more than five pages in Word and include:

1. For both iPhone and Galaxy:
   * The classifier you selected and the features (attributes) you used to train the classifier.
   * Your rationale for selecting the classifier that you did.
   * The features you eliminated from the data matrix and your rationale for doing so.
   * Comparative performance of the classifiers you tried (you can explain in text or with a chart).
2. What worked well. What didn’t work. What was difficult.
3. How the process to execute similar projects should be changed for the future.
4. **Submit Your Lessons Learned Report**