Summary of Lessons learned

From this exercise I have a better understanding of:

* What big data is and the power of i-clouds.
* Common Crawl and what scraping is.
* How to navigate within aws.
* JSON (data interchange format) and what it is for. (e.g. used to execute program files between my browser terminal and aws.
* How Python coding files can be used to execute scrapes on i-cloud data.
* How to analyze unstructured data using a Sentiment Analysis.
* How to develop ML training data-sets from unstructured data.

Additionally, I got more experience with:

* Using R/R-Studio and The carat Package
* Exploring large data sets in R.
* Graph coding in R.
* Features Selection methods in carat.
* Executing/coding ML algorithms in carat.
* Interpreting performance of ML models.

Question:

Why two models, one for each phone? Could one training data-set be developed for both phones by defining the sentiment classification differently (example below)? This would also save manual work time.

0 = Sentiment Unclear (would also include galaxy iphone unclear)

1 = iPhone Positive

2 = iPhone Negative

3 = Galaxy Positive

4 = Galaxy Negative

Working with Amazon Web Services:

I am glad to have done some work aws. Even though it took me a while to set up the account I got through it. I viewed it as an opportunity to overcome a set-back typical in the Data Analyst/Scientist world. I also got a better understanding of just what “big data” is and an idea of how to interface with the data and just how accessible it can be. I find Common Crawl and the concept of scraping it fascinating. However, it does take a good understanding of JSON and coding (e.g. Python) to do this.

Learnings from Unstructured Data Analysis:

The Sentiment Analysis was eye opening. It requires a good understanding of exactly what one is looking for so that it can be properly programed into a text analysis program to scrape the data. The concept of manually classifying some of the data to develop the model was also eye opening. However, It appeared that the manual classification subsets (≈13,000 each) are quite large relative to my large dataset (27,273). In this case, perhaps it would make more since to go ahead and manually classify the entire large data set for both phones in one fell swoop (see questions above). Perhaps in reality, datasets are much larger than 27,000 something on the order of 1 million….then the sample size of the training data sets makes more sense.

Model Development and Selection:

Features Selection: In all two Feature Selection methods were explored (Near Zero Variance, and Recursive Feature Elimination in carat). While a correlation analysis of the features was done, it was found that there were many high correlations among the 58 features which made it difficult to chose which features to drop. As such this Features Selection method was not used.

Modeling: As instructed, four different ML algorithms were run across the three data sets for each phone to find the best performance. As such 24 models were run and results are charted in the table below.



All models were trained on 10-fold cross validation with 1- repeat and using “Accuracy” as the performance metric. As for tuning parameters, a tuneLength of 5 was used on all models except for kknn, as a tuneLength of 10 was needed to achieve the highest accuracy for that algorithm. To maximize computational resources, I used 6 processor parallel processing out of capacity of 8 for my machine. I also recorded the computational time for model run to see how the Features Reduction methods affected computation time vs accuracy.

Model Selection: To select which model to use for each phone, the highest Accuracy and Kappa were are considered for both training and test with stronger consideration given to test results. Since several models are very close in these performance metrics, two other metrics were taken into account: computation time shown in the table and balance accuracy taken from the confusion matrix (not shown). As such the Random Forest models of the Recursive Feature Elimination (RFE) data sets were selected for both phone models (highlighted in green).

Programing Challenges: Since many models were being run, there were a lot of object names. It was quite challenging to keep the names straight and opportunities for typos as there was a lot of copying and pasting data models which required updating the object names.