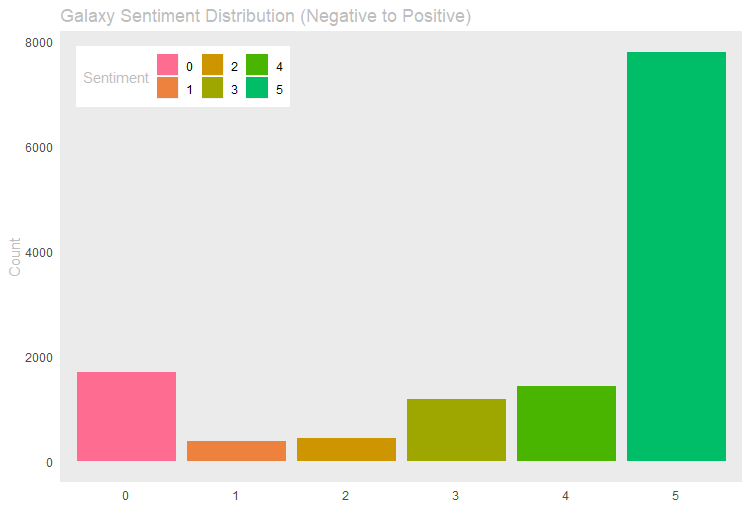
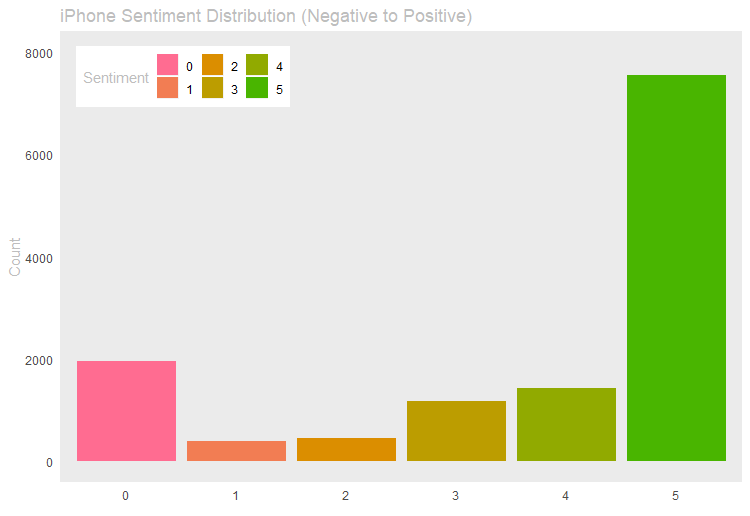
Sentiment Analysis

**Executive summary** – This project attempts to provide general sentiment score toward two specific smartphone models. Such task is required for Helio to deploy their ‘medical’ software in the best perceived model from people in order to maximize the efficiency of its software and reduce the risk of low software usage because of low smartphone performance, bad camera or poor display.

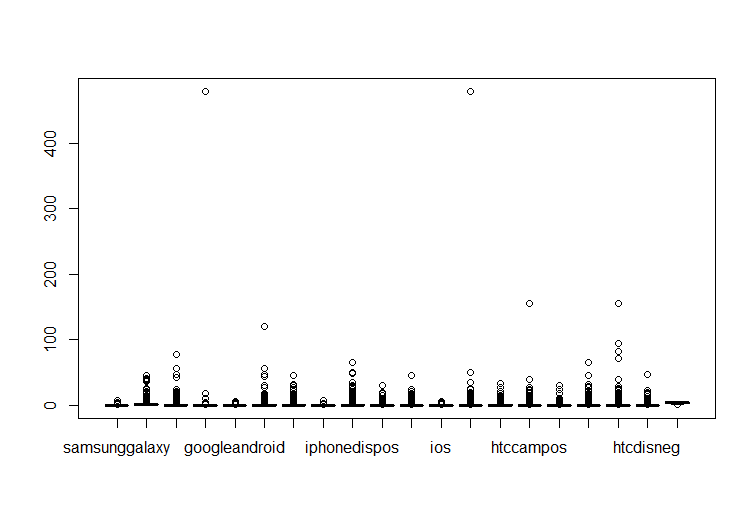
**Data exploration & cleaning –** The small matrix dataset consists of 12,933 observations spanning across 59 variables obtained manually while I was collecting the prediction set (large matrix) through Hadoop. Such dataset will be used for training the model and the large matrix for predictions.

Initial distribution of the sentiment scores for each model considered for this project, as it can be drawn from the graphs below. Distribution for both phones seems concentrated in the edges (either positive or negative) with little amount of scores for in-between categories. This is the case for most of customer reviews which tend to leave one when they feel overwhelmed or underwhelmed by the product in question. However, this could be a problem to our models. They will tend to misclassify the in-between categories most of the time.



**Fig. 1.1 Distribution of phone sentiment for both models**

The boxplot below shows the range of each variable in the dataset. *htcphone* and *htccampos* clearly are the gross outliers with 479 counts of mentions in one observation.



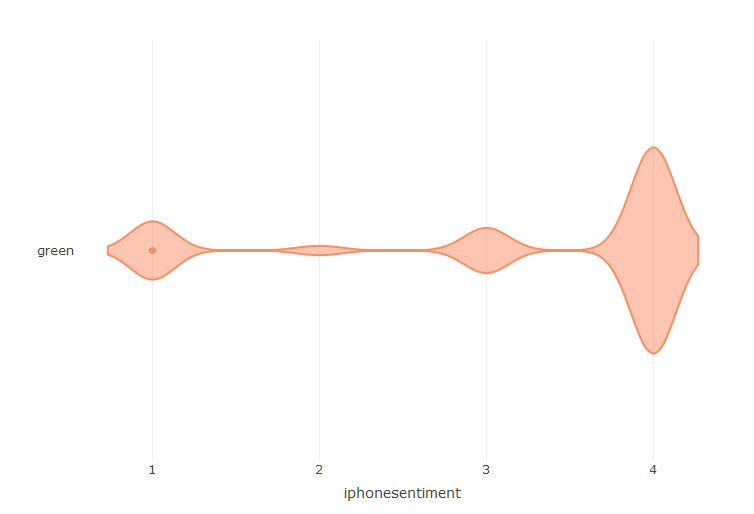
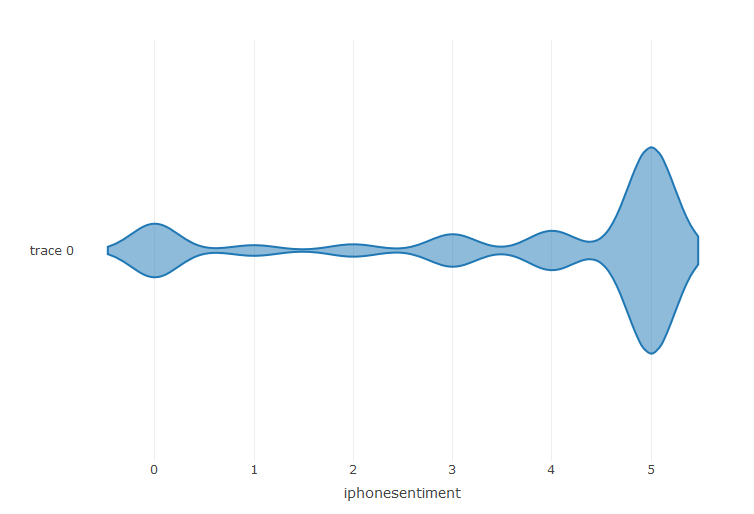
**Fig. 1.2 Range of values per each variable**

**Pre-Processing**

**RFE -** Removing a portion of the variables in the small matrices to reduce noise and only use essential features in order to capture the ‘normal’ pattern of the data. By doing so, the chances this model overfits the data are minimized. From 59 variables only 20 were used for the final model including the dependent variable. Such ‘trimming’ was done through **Recursive Feature Elimination (RFE)** which showed that the variables I decided to move forward with explained around 90% of the variation of the *phonesentiment* variable.

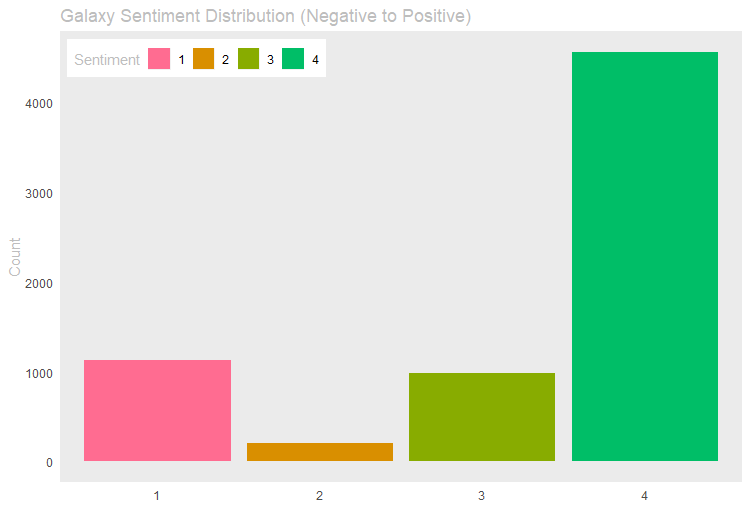
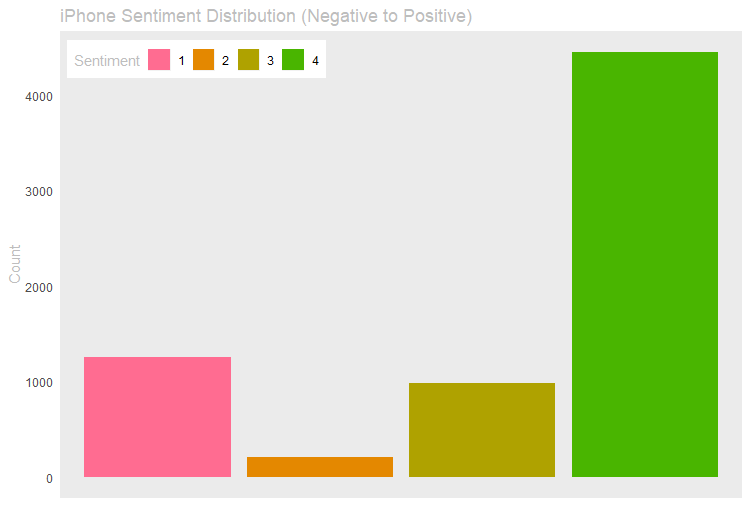
**Altering the dependent** – Having 6 different categories, it was easy for the algorithms to misclassify certain classes thus the levels of the dependent were reduced to only 4. This improved the accuracy of the final model noticeably. As it is shown on the violin plots below, the wider the shape gets the denser the population is at a certain sentiment score *(e.g. 5 - positive)*

**Before encoding After**



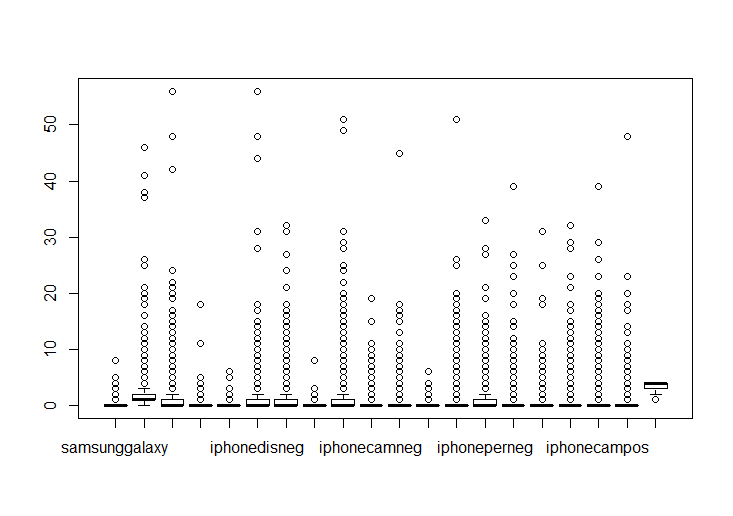
**Fig. 1.3 Distribution of phone sentiment after encoding (reducing score categories)**

**NearZero Variance** - Reducing noise was also done on a per-row basis. I counted the times when the sum of the rows of the dataset was 0 or 1 (meaning they had near-zero variance) and it made up 47% of the whole dataset. I decided to remove them because they did not provide any meaningful information and juts threw my models off. Indeed, after removing them the randomness in my classification dropped (kappa increased).



**Fig. 1.4 Distribution of phone sentiment after cleaning low variance observations**

**Outliers** – Besides the two gross outliers belonging to two of the *htc* variables, that were automatically remove after RFE, I decided to take out counts higher than 60 because they seemed out of the range considering the mean of the range of the population. 9 extra observations were removed and the final ‘normalized’ range for all variables is shown below.

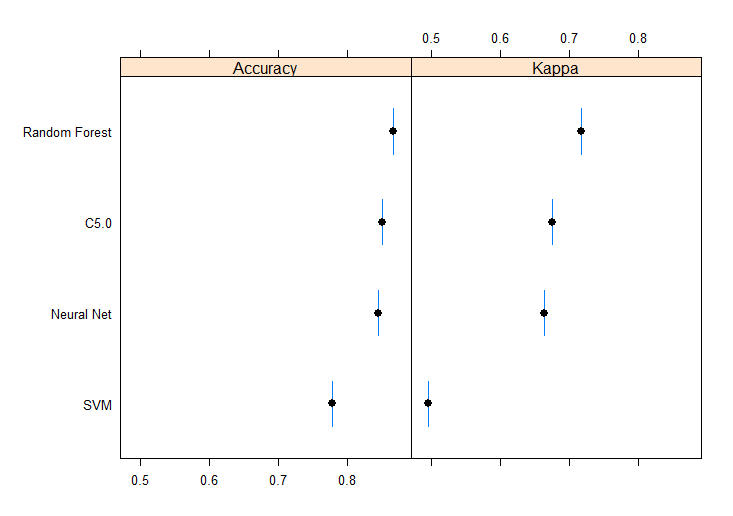


**Fig. 1.5 Normalized range after outlier removal**

**Other** - Several different pre-processing methods were used such as removing highly correlated independent variables, taking out NearZero variance columns and attempting to reduce the number of variables through Principal Component Analysis. Over and under sampling to try to balance the classes of phone’s sentiment scores worked in terms of the model classifying more correct ‘somewhat negative’ reviews although we do not care much for that category. What is important is not classifying negative reviews as positive and the opposite. Thus all of the methods mentioned above did not prove successful.

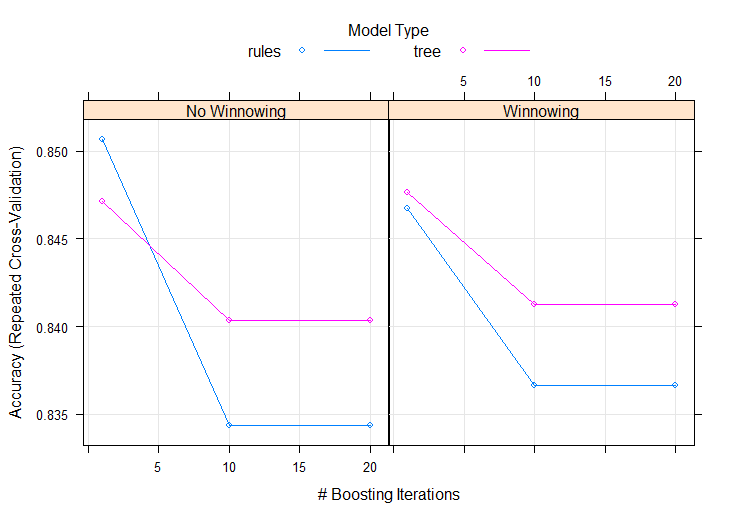
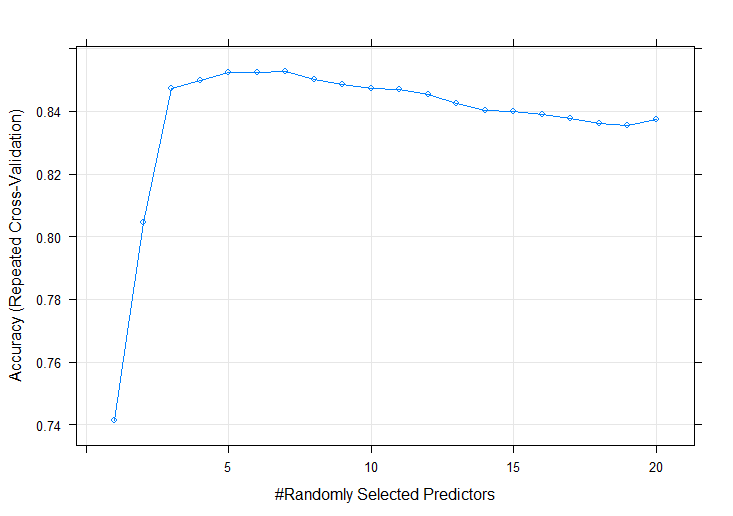
**Model selection**

The following models were tested against the training set: a simple Neural Net, C5.0, Random Forest and Linear Support Vector Machine. Below is a boxplot showing the accuracy and kappa of each model tested and their variance in the chosen metrics (small variance, no change if there is hyperparameter tuning).



**Fig. 1.6 Boxplot comparison of various models against training set**

As it can be inferred, random forest was slightly better than C5.0 in both metrics. Below there is a more detailed graph about how such models performed with different tuning parameters.

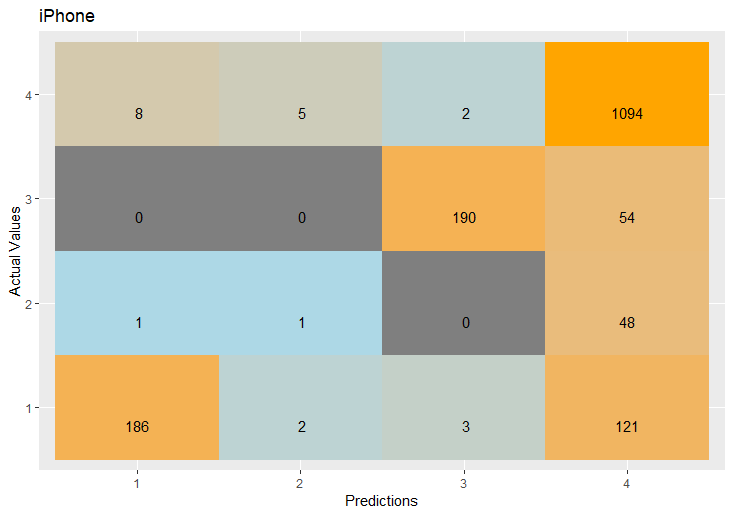
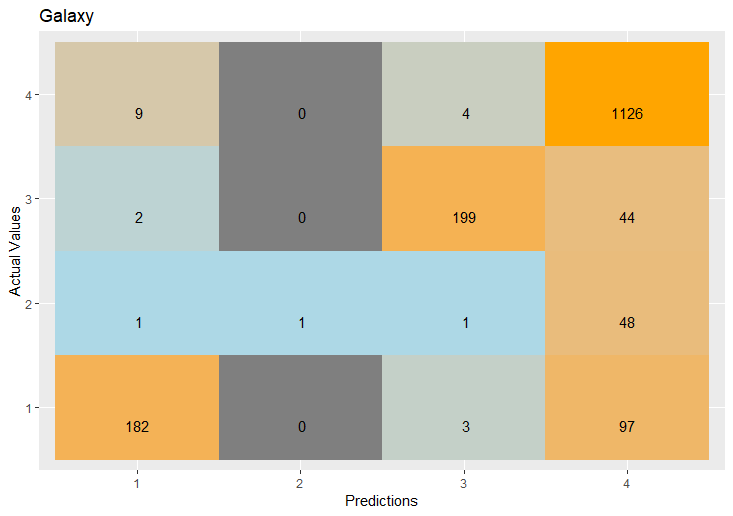


**Fig. 1.7 Tuning differences in the two leading models**

Next, when I compared them to the testing set random forest outperformed C5.0 by a thin margin of 0.01 for accuracy and 0.03 for *Kappa*. What influenced the decision for the final model was that our major challenge of not classifying positive reviews as negative and the opposite was done much better with the random forest than with C5.0 for both phone models.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Model* | **iPhone** | | **Galaxy** | |
|  | Accuracy | | Kappa | |
| **Random Forest** | 0.8577 | 0.695 | 0.8783 | 0.7332 |
| **C5.0** | 0.8426 | 0.6685 | 0.8655 | 0.7032 |

Moving on with the random forest, the confusion matrix below shows exactly how the model classified the data and where it went wrong.

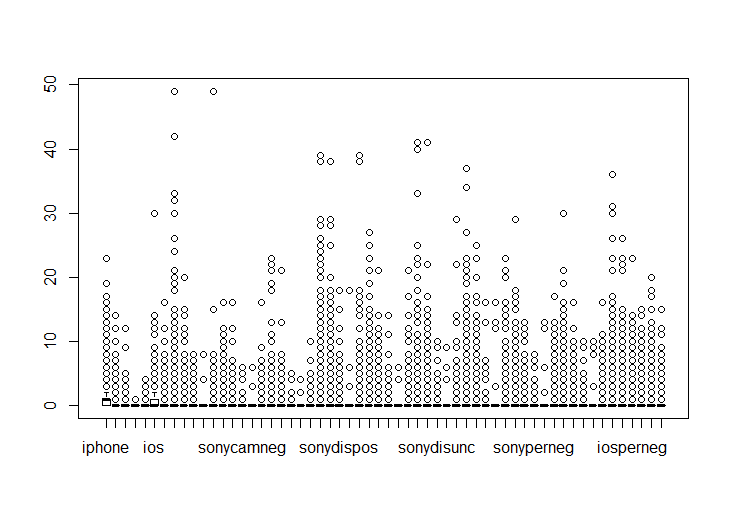
**Fig. 1.8 Confusion matrix for iPhone (left) and Galaxy (right)**

As it can be seen from above. Most errors are made because the ‘best’ category dominates the other classes in terms of portions of the dataset and the model classifies around 200 times sentiment scores that should have been for category 1, 2 or 3 as the 4th one (positive score). More data could potentially solve such issue.

Nonetheless, we see that the model for *Galaxy* is more accurate and makes less mistakes in mixing up negative and positive reviews (the two edges). This is a signal that the sentiment scores for our predictions will be more accurate for *Galaxy* phones rather than *iPhone*. This could come from ambiguous reviews left from *iPhone* customers or that people express more clearly their feelings toward *Galaxy* models. On this last note, such differences could come from the way the scripting for collecting such sentiment scores was created. All of this observations need further attention in the future, in order to improve things even more.

In the appendix, there are the variables used in both models ranked by importance (in explaining the sentiment score for each phone). The top 5 which is almost identical, is made up of other phone models. A possible explanation could be that when customers review they tend to compared different phone models and features. Another anomaly is that *iPhone* and *iOS* mentions tend to be more important when the sentiment is negative. Still, more information on this is needed to make more reasonable deductions.

**Predictions** – Identical transformations applied to the Small matrix were made to the Large Matrix obtained through Hadoop and hosted on Amazon Web services (EMR) console. Below is shown the normalized range of all the variables.



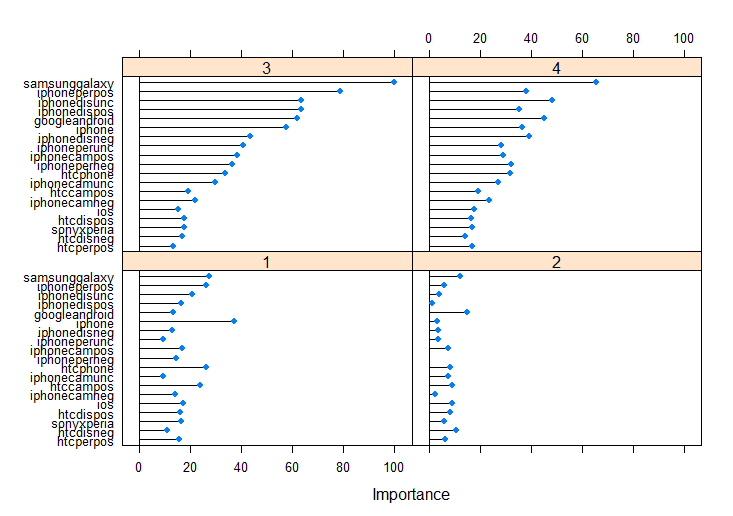
**Fig. 1.9 Cleaning of Large Matrix**

The range looks like the training set which means our model should perform similarly, not having huge variation. To conclude, below are the final predictions.

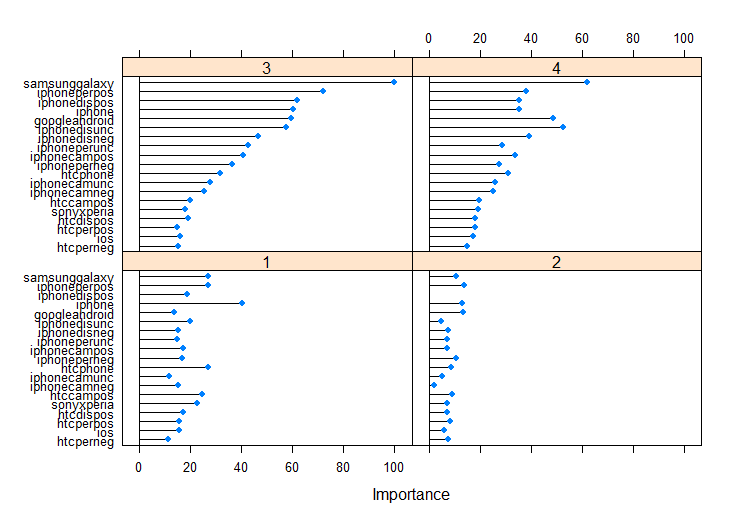
**Fig. 1.10 Predictions for both model compared for each sentiment score**

For ‘in-between’ categories *iPhone* and *Galaxy* are pretty much equal, while for the edges, *iPhone* has less negative reviews and more positive ones. Thus, the best model for the project you have in hand would be the *iPhone* model.

**Appendix:**



**Fig. 1.11 Importance of variables, random forest (iPhone)**



**Fig. 1.12 Importance of variables, random forest (Galaxy)**