HELIO PROJECT

Alert! Analytics

Abstract

Collect and develop a data matrix in the range of 20 thousand instances to analyze the sentiment toward handheld devices. Build models using a small data sample, finding the most optimal one, and using it to produce a report about the large data matrix.

Jimenez, Marcelo
mar_jim@utexas.edu

Introduction

Alert! Analytics is tasked with conducting a broad-based web sentiment analysis to gain insight into the attitudes towards suitable devices for Helio's medical apps. As technical support services are managed by a government agency that limits support to only to certain models of devices, Helios must narrow down their device support to their customer's most preferred device.

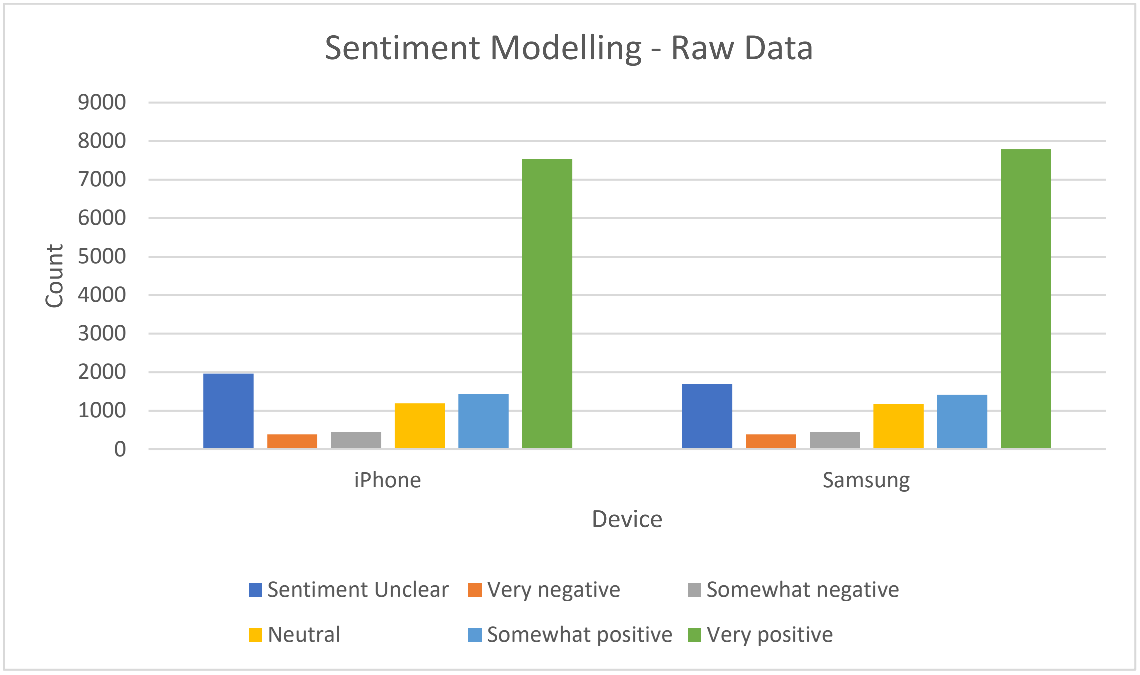
Amazon Web Services (AWS) will be used to search and gather the data. Using Elastic Map Reduce (EMR) to collect large amounts of the data from a repository of web data called Common Crawl. The aim of this is to create a Large Matrix to use for sentiment analysis of each device.

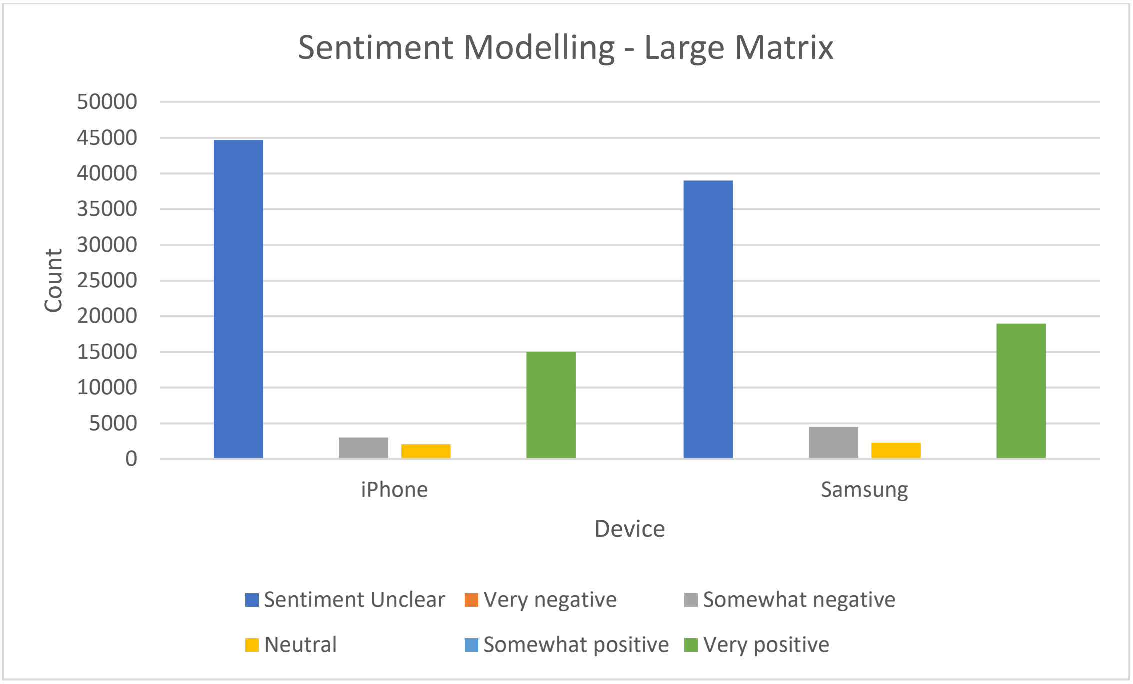
So far, python scripts have been made to filter and collect useful data from the repository. Furthermore, a small matrix was manually assessed capturing the sentiment towards iPhone and Samsung devices.

After successful negotiations with Apple and Samsung, iPhone and Galaxy take priority over the other handset devices in the sentiment analysis. So now the final task is to build models that understand the patterns in the two small matrices and then use those models with the Large Matrix to predict sentiment for iPhone and Galaxy.

Findings

After training multiple models and selecting a method for feature selection, the results show a slight preference towards Samsung Galaxy devices over iPhones. This result is only appreciated over a larger sample. The raw sentimental data used to build all the models showed they were very similar and practically impossible to tell immediately with a high degree of accuracy which one is more preferred. The results for both the raw sentimental data and the Large Matrix modelled sentimental data are shown below in the graph and table with labeled categories.





Raw Data

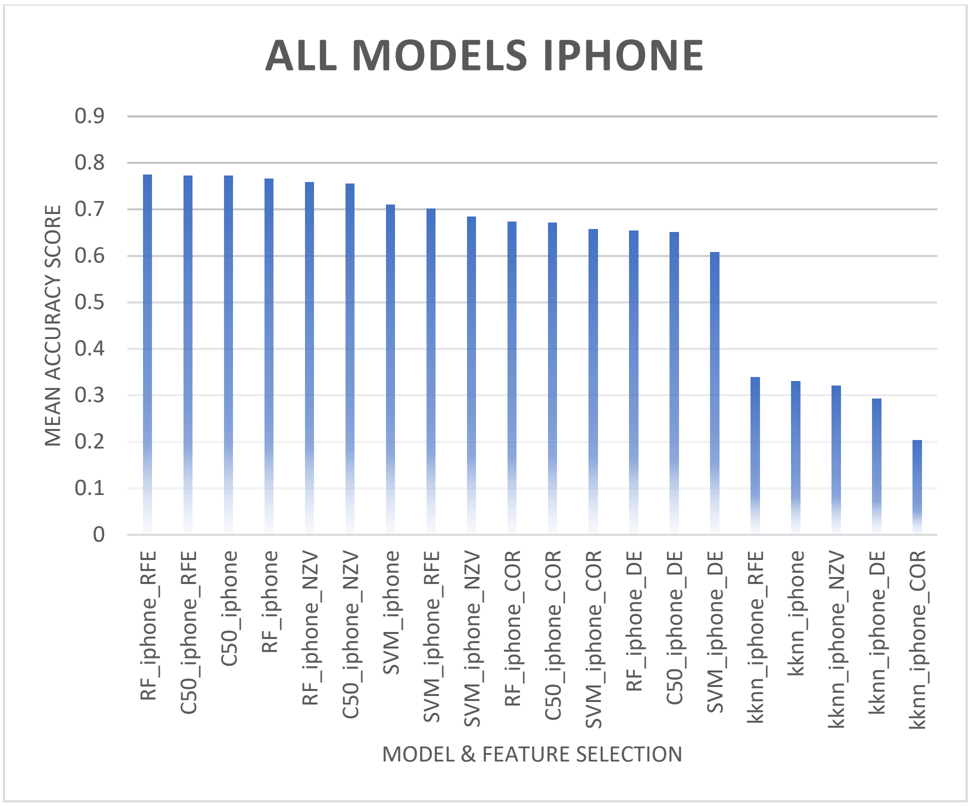
Device	Sentiment Unclear	Very negative	Somewhat negative	Neutral	Somewhat positive	Very positive
iPhone	1962	390	454	1188	1439	7540
Samsung	1696	382	450	1175	1417	7791

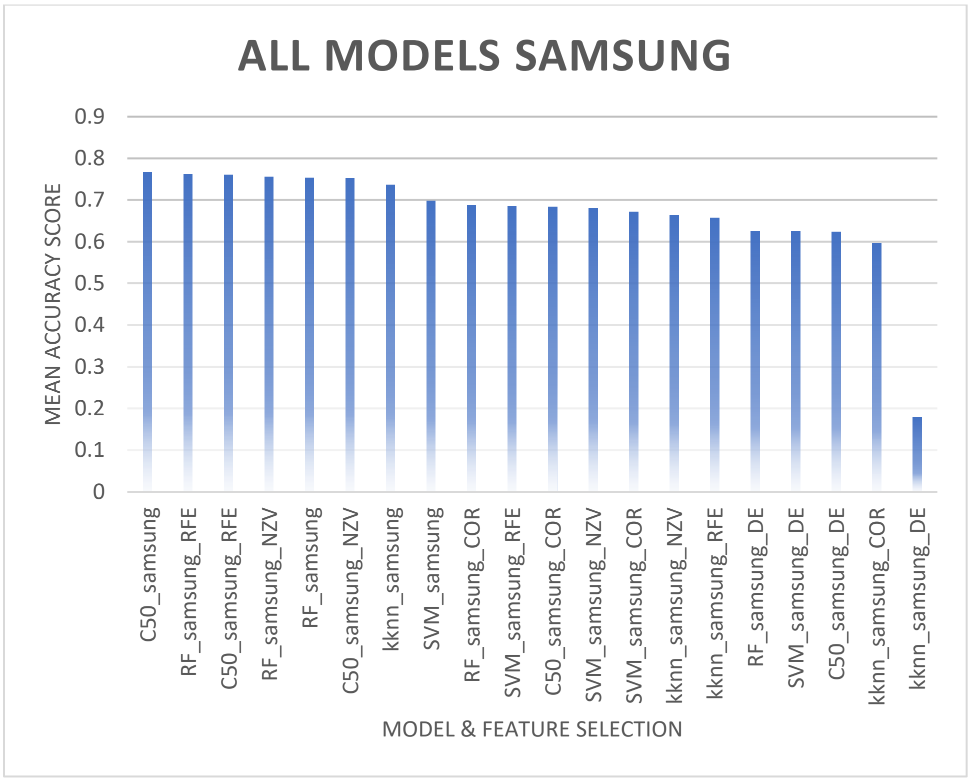
Large Matrix Modelled Data

Device	Sentiment Unclear	Very negative	Somewhat negative	Neutral	Somewhat positive	Very positive
iPhone	44725	0	3029	2071	2	15021
Samsung	39022	0	4482	2301	84	18959

Confidence

In the simplest terms, over 40 models for iPhone and Samsung were made to compare accuracy scores of each model and select the best out of them all. The iPhone model selected was Random Forest modelling using Recursive Feature Elimination for its feature selection, this yielded an accuracy score of 77.5% when testing it against with the raw sentimental data matrix. Similarly, for Samsung the model selected was C50 using the original data with no feature selection, this yielded an accuracy score of 76.6%. This is an acceptable number as it is much higher than just leaving the device selection to chance (50%). Below are some graphs that show the accuracy of all the models made for iPhone and Samsung.


ALL MODELS IPHONE
MEAN ACCURACY SCORE
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
RF_iphone_RFE
C50_iphone_RFE
C50_iphone
RF_iphone
RF_iphone_NZV
C50_iphone_NZV
SVM_iphone
SVM_iphone_RFE
SVM_iphone_NZV
RF_iphone_COR
C50_iphone_COR
SVM_iphone_COR
RF_iphone_DE
C50_iphone_DE
SVM_iphone_DE
kknn_iphone_RFE
kknn_iphone
kknn_iphone_NZV
kknn_iphone_DE
kknn_iphone_COR
MODEL & FEATURE SELECTION


ALL MODELS SAMSUNG
MEAN ACCURACY SCORE
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
C50_samsung
RF_samsung_RFE
C50_samsung_RFE
RF_samsung_NZV
RF_samsung
C50_samsung_NZV
kknn_samsung
SVM_samsung
RF_samsung_COR
SVM_samsung_RFE
C50_samsung_COR
SVM_samsung_NZV
SVM_samsung_COR
kknn_samsung_NZV
kknn_samsung_RFE
RF_samsung_DE
SVM_samsung_DE
C50_samsung_DE
kknn_samsung_COR
kknn_samsung_DE
MODEL & FEATURE SELECTION

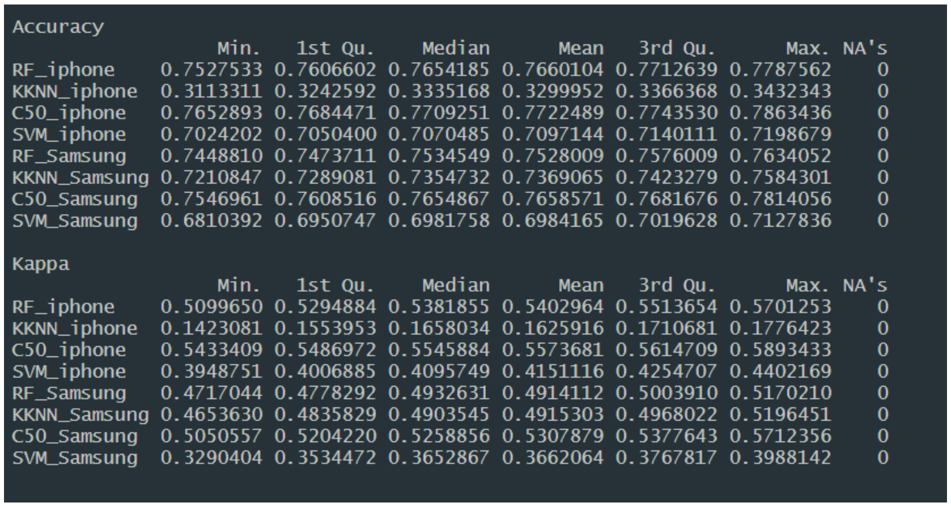
Implications

Knowing that there's a slight preference towards Samsung devices helps to narrow down even further which device Helio's app should support and allows for negotiations to proceed with a better understanding of the customer's preference. However, an argument could be made that the difference in preference is very small and therefore Helios might want to consider supporting both devices as they would cover a wider range of customer demographics. Furthermore, it is also worth noting that the model did not assign any "Very Negative" sentiment score towards either phone. This could be due to the very few scores given in the small sample data, so it could be in Helios best interest to obtain a larger sample set with which to build the models.

Methodology

After importing the data into RStudio and performing a quick EDA, it is clear most of the time taken for this project will be in the modelling. First, I built 10 different data frames containing both iPhone and Samsung sentimental data to find the best feature selection method. These include the original data, Domain Expertise (DE), Feature Correlation (COR), Near zero variance (NZV), and Recursive Feature Elimination (RFE). They each have their own merit towards selecting the best features but the most useful was RFE as it performs the feature selection using random forest and decision tree algorithms to iterate through all the features and eliminate the unnecessary ones, the goal being to achieve the best accuracy with the smallest number for features.

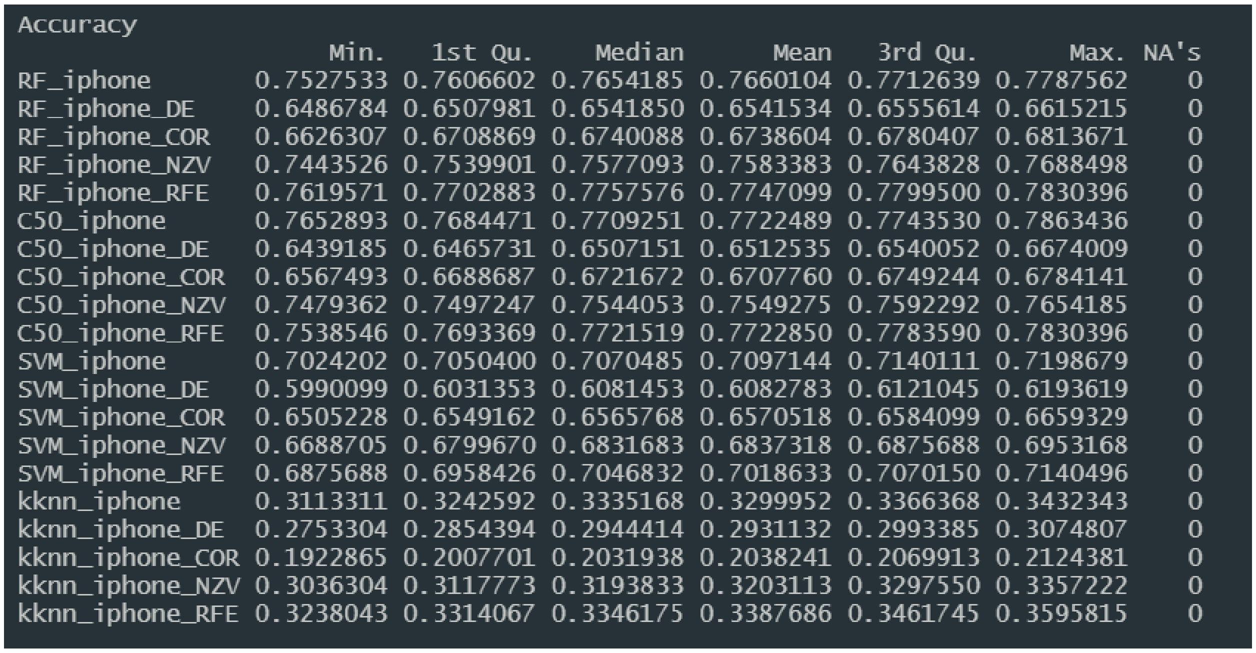
Then, all the data frames where split into "training" and "testing" data to make all the different models and test their accuracy. Also, due to the amount of data the computer would have to process, I used parallel processing in R to accelerate the model creation. The models used in all the data frames were Random Forest, C50 (Decision Tree algorithm), SVM (Support Vector Machine), and KKNN (same as K-nearest neighbor algorithm but optimized for a higher dimensional data frame). All their respective Accuracy and Kappa scores where recorded and stored through the "resample" function in R. Here is a sample table of all the raw results using the original data with no feature selection:



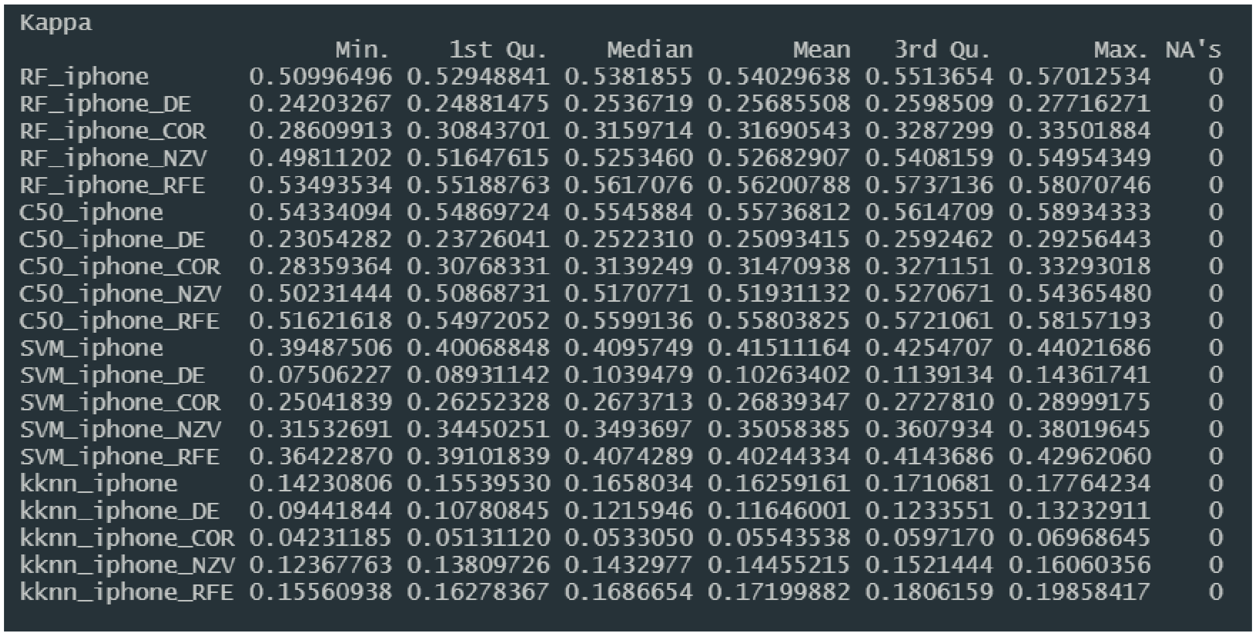
```
Accuracy
                 Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
RF_iphone    0.7527533 0.7606602 0.7654185 0.7660104 0.7712639 0.7787562    0
KKNN_iphone  0.3113311 0.3242592 0.3335168 0.3299952 0.3366368 0.3432343    0
C50_iphone   0.7652893 0.7684471 0.7709251 0.7722489 0.7743530 0.7863436    0
SVM_iphone   0.7024202 0.7050400 0.7070485 0.7097144 0.7140111 0.7198679    0
RF_Samsung   0.7448810 0.7473711 0.7534549 0.7528009 0.7576009 0.7634052    0
KKNN_Samsung 0.7210847 0.7289081 0.7354732 0.7369065 0.7423279 0.7584301    0
C50_Samsung  0.7546961 0.7608516 0.7654867 0.7658571 0.7681676 0.7814056    0
SVM_Samsung  0.6810392 0.6950747 0.6981758 0.6984165 0.7019628 0.7127836    0

Kappa
                 Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
RF_iphone    0.5099650 0.5294884 0.5381855 0.5402964 0.5513654 0.5701253    0
KKNN_iphone  0.1423081 0.1553953 0.1658034 0.1625916 0.1710681 0.1776423    0
C50_iphone   0.5433409 0.5486972 0.5545884 0.5573681 0.5614709 0.5893433    0
SVM_iphone   0.3948751 0.4006885 0.4095749 0.4151116 0.4254707 0.4402169    0
RF_Samsung   0.4717044 0.4778292 0.4932631 0.4914112 0.5003910 0.5170210    0
KKNN_Samsung 0.4653630 0.4835829 0.4903545 0.4915303 0.4968022 0.5196451    0
C50_Samsung  0.5050557 0.5204220 0.5258856 0.5307879 0.5377643 0.5712356    0
SVM_Samsung  0.3290404 0.3534472 0.3652867 0.3662064 0.3767817 0.3988142    0
```

Finally, these are the raw results for iPhone with all the feature selections and models made:



```
Accuracy
                     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
RF_iphone       0.7527533 0.7606602 0.7654185 0.7660104 0.7712639 0.7787562    0
RF_iphone_DE    0.6486784 0.6507981 0.6541850 0.6541534 0.6555614 0.6615215    0
RF_iphone_COR   0.6626307 0.6708869 0.6740088 0.6738604 0.6780407 0.6813671    0
RF_iphone_NZV   0.7443526 0.7539901 0.7577093 0.7583383 0.7643828 0.7688498    0
RF_iphone_RFE   0.7619571 0.7702883 0.7757576 0.7747099 0.7799500 0.7830396    0
C50_iphone      0.7652893 0.7684471 0.7709251 0.7722489 0.7743530 0.7863436    0
C50_iphone_DE   0.6439185 0.6465731 0.6507151 0.6512535 0.6540052 0.6674009    0
C50_iphone_COR  0.6567493 0.6688687 0.6721672 0.6707760 0.6749244 0.6784141    0
C50_iphone_NZV  0.7479362 0.7497247 0.7544053 0.7549275 0.7592292 0.7654185    0
C50_iphone_RFE  0.7538546 0.7693369 0.7721519 0.7722850 0.7783590 0.7830396    0
SVM_iphone      0.7024202 0.7050400 0.7070485 0.7097144 0.7140111 0.7198679    0
SVM_iphone_DE   0.5990099 0.6031353 0.6081453 0.6082783 0.6121045 0.6193619    0
SVM_iphone_COR  0.6505228 0.6549162 0.6565768 0.6570518 0.6584099 0.6659329    0
SVM_iphone_NZV  0.6688705 0.6799670 0.6831683 0.6837318 0.6875688 0.6953168    0
SVM_iphone_RFE  0.6875688 0.6958426 0.7046832 0.7018633 0.7070150 0.7140496    0
kknn_iphone     0.3113311 0.3242592 0.3335168 0.3299952 0.3366368 0.3432343    0
kknn_iphone_DE  0.2753304 0.2854394 0.2944414 0.2931132 0.2993385 0.3074807    0
kknn_iphone_COR 0.1922865 0.2007701 0.2031938 0.2038241 0.2069913 0.2124381    0
kknn_iphone_NZV 0.3036304 0.3117773 0.3193833 0.3203113 0.3297550 0.3357222    0
kknn_iphone_RFE 0.3238043 0.3314067 0.3346175 0.3387686 0.3461745 0.3595815    0
```



```
Kappa
                      Min.    1st Qu.    Median       Mean   3rd Qu.       Max. NA's
RF_iphone       0.50996496 0.52948841 0.5381855 0.54029638 0.5513654 0.57012534    0
RF_iphone_DE    0.24203267 0.24881475 0.2536719 0.25685508 0.2598509 0.27716271    0
RF_iphone_COR   0.28609913 0.30843701 0.3159714 0.31690543 0.3287299 0.33501884    0
RF_iphone_NZV   0.49811202 0.51647615 0.5253460 0.52682907 0.5408159 0.54954349    0
RF_iphone_RFE   0.53493534 0.55188763 0.5617076 0.56200788 0.5737136 0.58070746    0
C50_iphone      0.54334094 0.54869724 0.5545884 0.55736812 0.5614709 0.58934333    0
C50_iphone_DE   0.23054282 0.23726041 0.2522310 0.25093415 0.2592462 0.29256443    0
C50_iphone_COR  0.28359364 0.30768331 0.3139249 0.31470938 0.3271151 0.33293018    0
C50_iphone_NZV  0.50231444 0.50868731 0.5170771 0.51931132 0.5270671 0.54365480    0
C50_iphone_RFE  0.51621618 0.54972052 0.5599136 0.55803825 0.5721061 0.58157193    0
SVM_iphone      0.39487506 0.40068848 0.4095749 0.41511164 0.4254707 0.44021686    0
SVM_iphone_DE   0.07506227 0.08931142 0.1039479 0.10263402 0.1139134 0.14361741    0
SVM_iphone_COR  0.25041839 0.26252328 0.2673713 0.26839347 0.2727810 0.28999175    0
SVM_iphone_NZV  0.31532691 0.34450251 0.3493697 0.35058385 0.3607934 0.38019645    0
SVM_iphone_RFE  0.36422870 0.39101839 0.4074289 0.40244334 0.4143686 0.42962060    0
kknn_iphone     0.14230806 0.15539530 0.1658034 0.16259161 0.1710681 0.17764234    0
kknn_iphone_DE  0.09441844 0.10780845 0.1215946 0.11646001 0.1233551 0.13232911    0
kknn_iphone_COR 0.04231185 0.05131120 0.0533050 0.05543538 0.0597170 0.06968645    0
kknn_iphone_NZV 0.12367763 0.13809726 0.1432977 0.14455215 0.1521444 0.16060356    0
kknn_iphone_RFE 0.15560938 0.16278367 0.1686654 0.17199882 0.1806159 0.19858417    0
```

For more information about the methodology and the exact commands used to build the models please refer to the code attached to this project.