# **Task Description**

Hello,

The Helio project manager called me yesterday with some new developments. The initial discussions with Apple and Samsung have progressed over the last few weeks. At this point, they would like us to prioritize—and of course speed up if we can—our sentiment analysis of the iPhone and the Galaxy over the other handsets in the short list.

While you were working on collecting the Large Matrix, an Alert! Analytics team has been has manually labeling each instance of two small matrices with sentiment toward iPhone and Samsung Galaxy. Manually labeling means that the team read through each webpage and assigned a sentiment rating based on their findings. I have attached two labelled matrices (one for each device).

Our analytic goal 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.

Our next steps are as follows:

* Set up parallel processing
* Explore the Small Matrices to understand the attributes
* Preprocessing & Feature Selection
* Model Development and Evaluation
* Feature Engineering
* Apply Model to Large Matrix and get Predictions
* Analyze results, write up findings report
* Write lessons learned report

I would like you to use the R statistical programming language and the caret package to perform this work. To get the best results, I would like you to compare the performance metrics of four different classifiers, namely C5.0, random forest, KKNN and support vector machines. This should be done for both the iPhone and Galaxy data sets.

After comparing the performance of the classifiers in "out of the box modeling, see if you can improve the performance metrics with feature selection/feature engineering. You should explore the results from several methods. This effort may or may not lead to better classifier performance, but always worth trying. After identifying your most optimal model use it to predict sentiment in the Large Matrix.

In terms of your analysis, Helio prefers short reports rather than presentations, so I would like you to prepare a document that summarizes your findings. In this summary, please lay out your interpretation of the results; your confidence in the results; and a high-level recap of what you did. In addition to your Summary of Findings for Helio, I would like you to prepare a brief Lessons Learned Report. This report will be valuable tool to improve our processes for these types of projects in the future.

Thank you,

Michael Ortiz

Senior Vice-President

Alert! Analytics

# **Task Solution**

Two data sets (iphone\_smallmatrix\_labeled\_8d.csv and galaxy\_smallmatrix\_labeled\_9d.csv) were provided by Michael Ortiz, to conduct feature engineering and model development for Apple iPhone and Samsung Galaxy phones. The best models in each case (highest kappa and accuracy values) will be used to conduct the sentiment analysis of the large matrix, this matrix was previously crawled using AWS elastic map reduced. For each small matrix dataset, 5 models will be developed to select the best model and then feature engineering will be conducted to enhance the model and get predictions with higher accuracy.

* 1. **Models Configuration**

**Models configuration:** Each model contains the following processes and parameters:

* Pre-processing
* Data Partition (70/30)
* Train Control (method = "cv", number = 5)
* Predictions
* Post Resample

**Selected Algorithms:** Decision Tree (C5.0), Random Forest (RF), Support Vector Machine (SVM), K-nearest neighbor (KKNN) and Gradient Boosting Machine (GBM).

* 1. **Data preprocessing, modelling and predictions**

The flow chart listed below (Figure 1 and 2) shows the preprocessing steps defined to subset the data, build, tune and feature engineering all the algorithms.

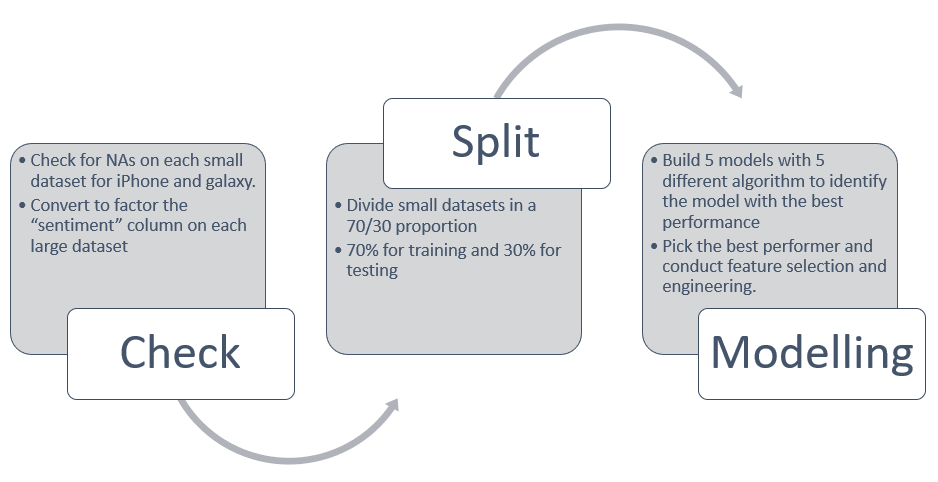


Figure 1 – Preprocessing data flow

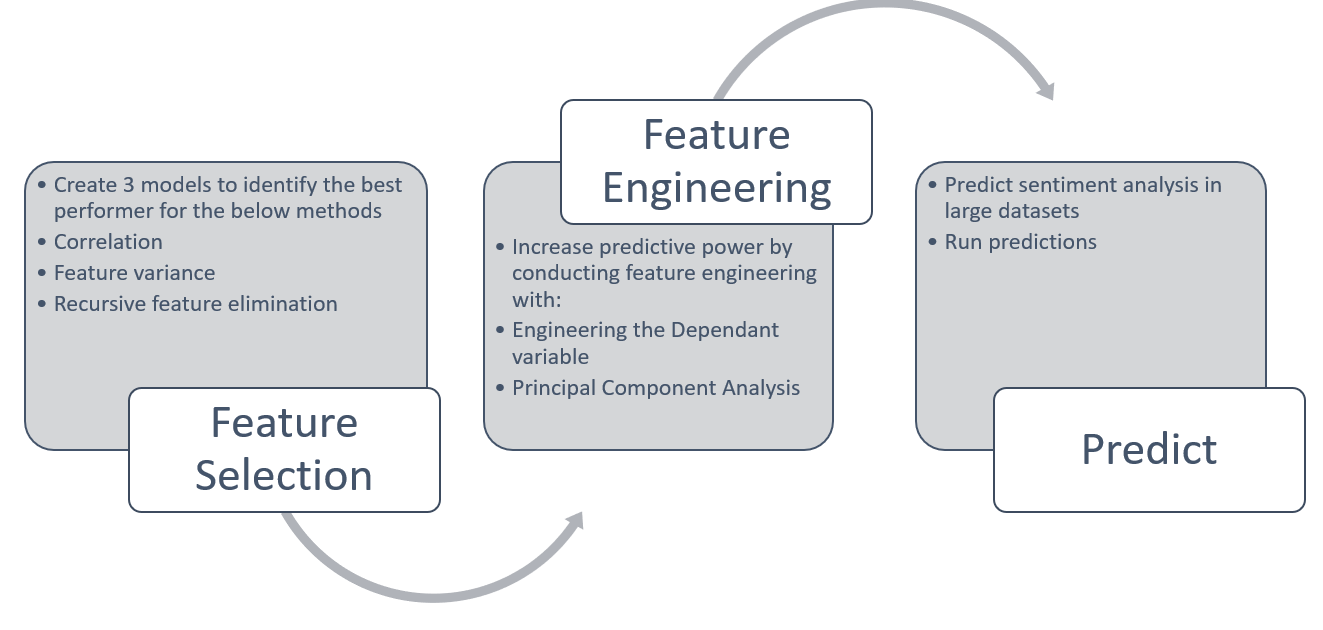


Figure 2 – Feature selection and feature engineering data flow

# **Results**

* 1. **Models Results**

Based on the proposed steps in section 2.2, the modelling step provided the following outcomes. The C5.0, RF, SVM, KKNN and GBM models were fitted under the specified conditions, the model with the best performance using the **Post Resample** values, is the Random Forest for both phones (see Table 1 and figure 3).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **iPhone** | | **Galaxy** | |
| **Model Name** | **Accuracy** | **Kappa** | **Accuracy** | **Kappa** |
| Decision Tree (C5.0) | 0.7724 | 0.5587 | 0.7675 | 0.5322 |
| Random Forest (RF) | **0.7755** | **0.5662** | **0.7703** | **0.5378** |
| Support Vector Machine (SVM) | 0.7113 | 0.4190 | 0.6982 | 0.368 |
| K-nearest neighbor (KKNN) | 0.3506 | 0.1755 | 0.698 | 0.4511 |
| Gradient Boosting Machine (GBM) | 0.7732 | 0.5581 | 0.7706 | 0.5386 |

Table 1 – Kappa and accuracy values for iPhone and Galaxy small datasets

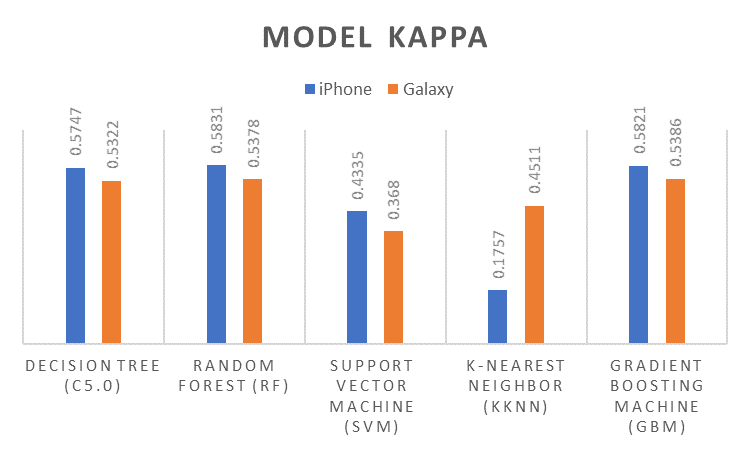
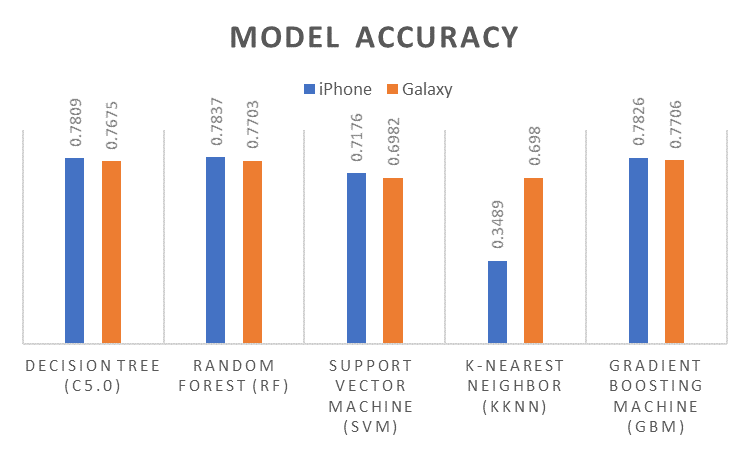


Figure 3 – Accuracy and Kappa values for out of the box models

The first round of performance improvement is based on feature selection, three methods were selected: correlation, feature variance and recursive feature elimination.

These methods were executed in the random forest model, the outcomes of this process in **Post Resample** are showed in Table 2 and figure 4.

The feature selection method with the best performance is Recursive Feature Elimination (RFE), there is no much variation in the performance metrics for iPhone and galaxy phones with the feature selection.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **iPhone** | | **Galaxy** | |
| **Model Name** | **Accuracy** | **Kappa** | **Accuracy** | **Kappa** |
| Correlation | 0.7524 | 0.5147 | 0.7509 | 0.492 |
| Feature Variance | 0.7586 | 0.5250 | 0.7548 | 0.5024 |
| Recursive Feature Elimination | **0.7753** | **0.5631** | **0.7708** | **0.5405** |

Table 2 – Kappa and accuracy values for iPhone & Galaxy after first round of improvements with RF

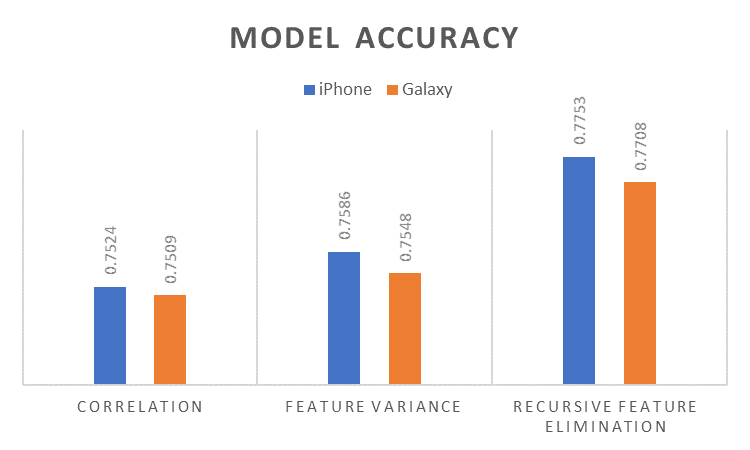
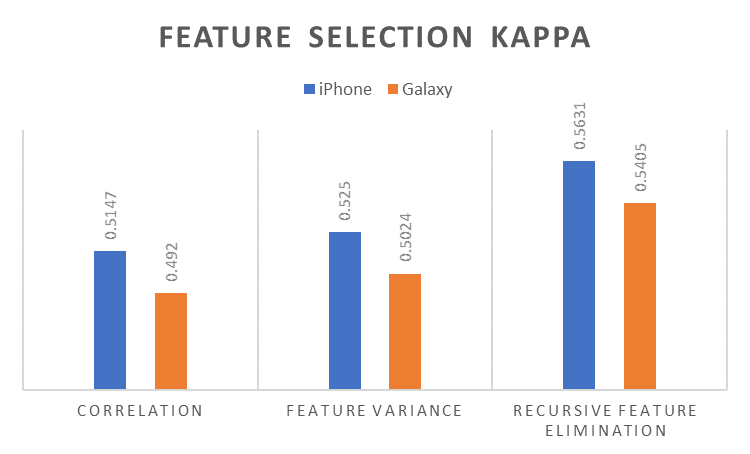
 

Figure 4 – Accuracy and Kappa values for enhanced models with feature selection

The second round of performance improvement is based on feature engineering, two methods were selected: altering the dependant variable and principal component analysis. These methods were executed in the random forest model with RFE, the outcomes of this process in **Post Resample** are showed in Table 3 & figure 5. The feature engineering method with the best performance is altering the dependant variable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **iPhone** | | **Galaxy** | |
| **Model Name** | **Accuracy** | **Kappa** | **Accuracy** | **Kappa** |
| Altering the dependant variable | **0.8496** | **0.6258** | **0.8494** | **0.6097** |
| Principal Component Analysis | 0.8439 | 0.6121 | 0.8411 | 0.5904 |

Table 3 – Kappa and accuracy values for iPhone & Galaxy after second round of improvements with RF

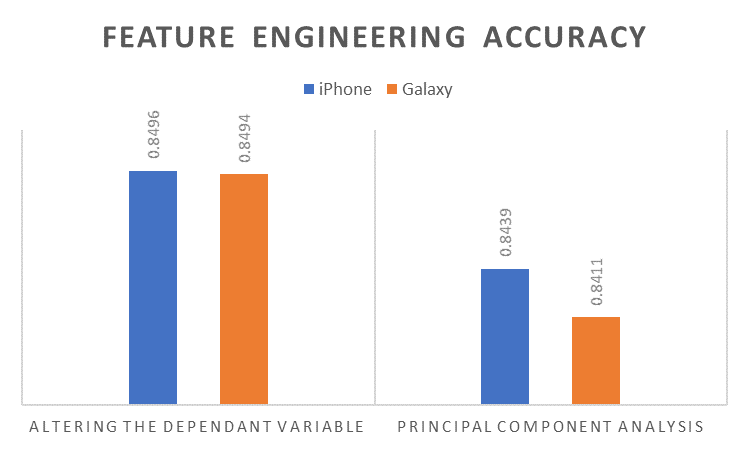
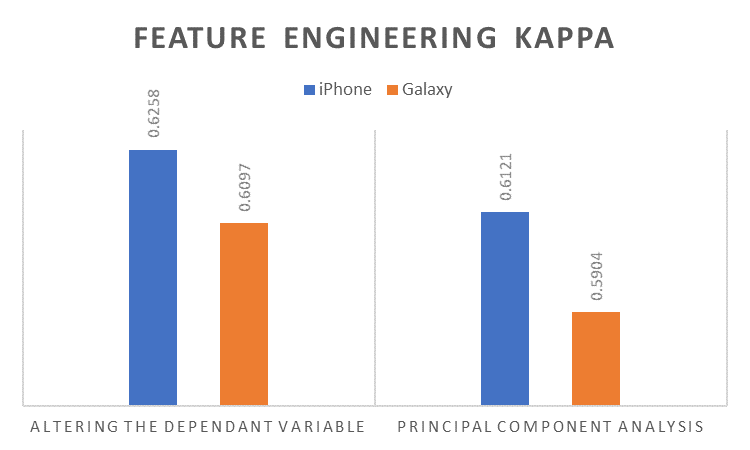
 

Figure 5 – Accuracy and Kappa values for enhanced models with feature engineering

The round of enhancements shows an increase in terms of accuracy and kappa values from 0.78 to 0.84 in iPhone and 0.77 to 0.84 in galaxy, this represents nearly a 7% increase for both models.

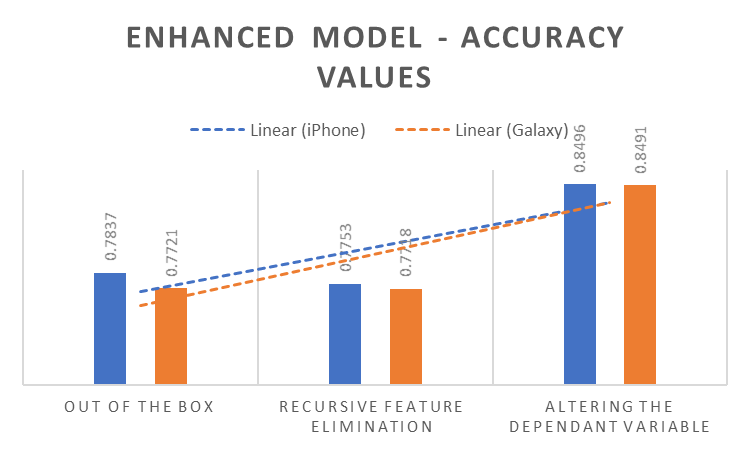
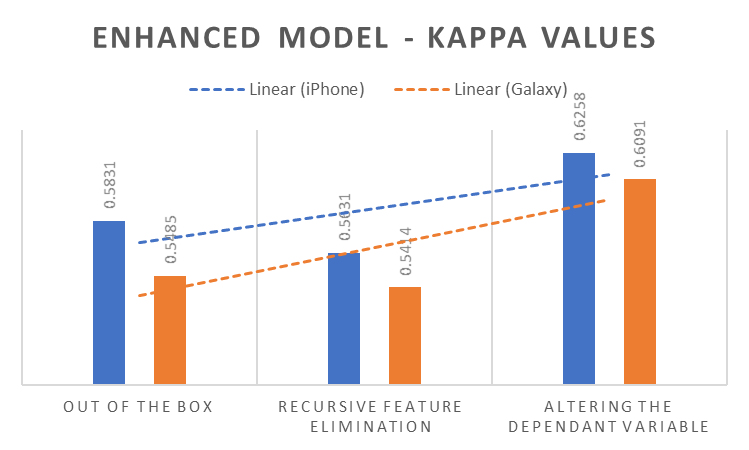
 

Figure 6 – Model enhancement from out of the box model until feature engineering

* 1. **Sentiment Analysis**

The original sentiment categories include very negative, negative, somewhat negative, somewhat positive, positive, very positive for iPhone and Galaxy in small matrix.

The sentiment analysis was conducted using a random forest model with RFE, the dependent variable was engineered, and some of these levels of this variable were combined, which would help increased the accuracy. After that the sentiment categories include negative, somewhat negative, somewhat positive, and positive for iPhone and Galaxy in large matrix.

The large matrix we collected from Common Crawl has 32226 observations, the distribution across the categories are below (Figure 7)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Galaxy** | **iPhone** | **% Galaxy** | **% iPhone** |
| 1: negative | 13107 | 13086 | 40.7% | 40.6% |
| 2: somewhat negative | 1011 | 892 | 3.1% | 2.8% |
| 3: somewhat positive | 2058 | 2470 | 6.4% | 7.7% |
| 4: positive | 16050 | 15778 | 49.8% | 49.0% |
| **Grand Total** | **32226** | **32226** | **100.0%** | **100.0%** |

Figure 7 – Sentiment categories distribution

A graphical representation of the category distribution can be appreciated in figure 8 – iPhone and Galaxy Sentiment.

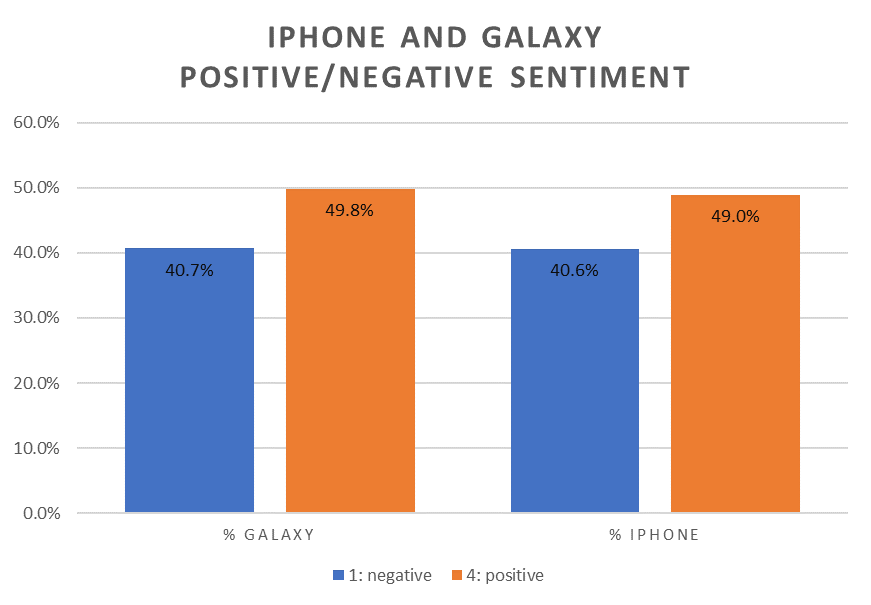
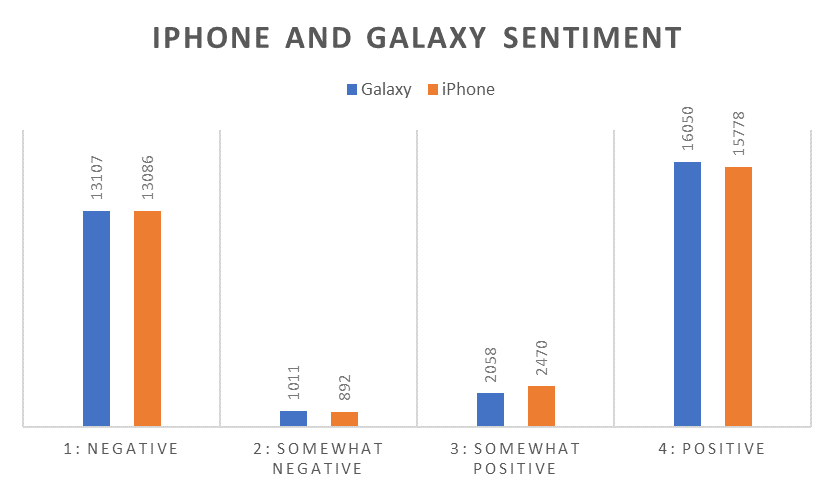


Figure 8 – iPhone and Galaxy Sentiment

Based in these outcomes the recommendation is for the Galaxy phone, since the positive sentiments are slightly higher than the iPhone positive sentiments (49,8% against 49,0%)

# **Results Discussion**

The goal of this project was accomplished, the feasibility to determine / predict the location of a user based on WAPs signals, was successfully achieved.

The predictions of the models are very accurate and quite representative. However, we have to take into account that we combined both datasets (Training and Validation) to create the models, since the training data had a lack of representativity. Hence, it is difficult to gauge if our model can have a problem of overfitting.

This model it’s an accurate and inexpensive way to determine an user location in indoor spaces, several use cases can be developed to take advantage of the model.

# **Recommendations**

Use the model RF to conduct the predictions since the kappa and accuracy values are the best.

# **R scripts**

Provided in a zip file