Lessons Learned Developing Models to Predict Sentiment

Assn 4 Task 3

May 1, 2019

# Modeling approach

The following approach was conducted for both the Apple iPhone assessment and the Samsung Galaxy assessment. These assessments were *conducted separately*, as assessment each accesses a different file in order to create the prediction classifications.

* Machine learning using R’s Caret package, following the Data Science process
* Four base algorithms were trialed: C5.0, Random Forest (rf), K-Nearest Neighbors (kknn), and Support Vector Machine (svmLinear2).
* The algorithm or algorithms that rendered the model with the highest performance metrics was then trialed using the following variations. The goal was to determine if a model with higher performance metrics could be created:
  + Feature Selection with highly correlated features removed
  + Feature Selection with features with no variance removed
  + Feature Selection with recommended features removed, using Caret rfe, or recursive feature elimination process
  + Feature engineering recoding the sentiment of the target/dependent variable
  + Feature engineering using Principal Component Analysis (PCA)
  + Feature engineering combining both recoding the sentiment of the target variable and PCA

# iPhone Analysis

The iPhone training and testing models were created by accessing the iphone\_smallmatrix\_labeled\_8d.csv file and using the iphonesentiment feature as the target variable. This target variable has 6 possible outcomes, which we mapped to very negative, negative, somewhat negative, somewhat positive, positive, and very positive. The classification algorithms of C5.0 and Random Forest were both found to render the highest performing models.

**iPhone Performance Metrics of Models Created with no Feature Selection nor Feature Engineering Employed**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Accuracy** | **Kappa** | **P-Value** |
| **C5.0** | 0.7730077 | 0.5590862 | < 2.2e-16 |
| **Random Forest** | 0.7768638 | 0.5703315 | < 2.2e-16 |
| **KNN** | 0.3609 | 0.1818 | < 2.2e-16 |
| **SVM** | 0.7147 | 0.4273 | < 2.2e-16 |

Because the initial performance metrics were so similar for C5.0 and Random Forest, both were trialed more extensively using the feature selection and feature engineering approaches listed in the previous section of Modeling Approach.

* With feature engineering, Accuracy and Kappa increased for models with recoded sentiment in the target variable.
* Models created with PCA did not improve upon the performance metrics shown by other C5.0 and Random Forest models c
* As an option, the feature engineering options of recoded sentiment and PCA were combined into building a C5.0 model. It had a very good outcome but could not outperform the recoded sentiment models created previously.
* Random Forest models created with feature selection options had good Accuracy and Kappa outcomes, but they were slightly less than that of other models
* To make a final differentiation between models, a comparison per target sentiment area was conducted assessing Positive Predicted Value and Negative Predicted Value. These values were considered in light of Accuracy and Kappa per model.
* P value for all models was acceptable.

From this chart below we see that **Random Forest with feature engineering of recoded sentiment has the best performance metrics and no areas of concern, thus it will be chosen as our top model.**

**iPhone Confusion Matrix Outcomes for Top Model Test Sets (15 models generated, only displaying the top 5)**

[**Table Legend**](#TableLegend)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Kappa |  | Predicted Value Type | Sentiment Type | | | | | |
|  |  |  |  |  | Very negative | Negative | Somewhat negative | Somewhat positive | Positive | Very positive |
| C5 all features | 0.773 | 0.5591 |  | Positive Predicted Value | 0.95214 | NaN | 0.900000 | 0.94561 | 0.90683 | 0.7286 |
|  |  |  |  | Negative Predicted Value | 0.93988 | 0.96992 | 0.969509 | 0.96439 | 0.92357 | 0.9718 |
| RF all features | 0.7769 | 0.5703 |  | Positive Pred Value | 0.96500 | 0.3333333 | 0.720000 | 0.94215 | 0.86885 | 0.7343 |
|  |  |  |  | Negative Predicted Value | 0.94212 | 0.9701569 | 0.969470 | 0.96491 | 0.92663 | 0.9625 |
| C5 Recoded Sentiment | 0.8401 | 0.5979 |  | Positive Pred Value |  | 0.95039 | 1.000000 | 0.91632 | 0.8207 |  |
|  |  |  |  | Negative Predicted Value |  | 0.90277 | 0.969024 | 0.96248 | 0.9624 |  |
| RF Recoded Sentiment | 0.8452 | 0.6114 |  | Positive Pred Value |  | 0.96834 | 0.941176 | 0.93469 | 0.8236 |  |
|  |  |  |  | Negative Predicted Value |  | 0.90373 | 0.969016 | 0.96516 | 0.9735 |  |
| C5 Recoded Sentiment & PCA | 0.8355 | 0.5854 |  | Positive Pred Value |  | 0.9086 | 0.882353 | 0.95794 | 0.8181 |  |
|  |  |  |  | Negative Predicted Value |  | 0.9033 | 0.968758 | 0.95892 | 0.9513 |  |

Model Training Metrics from RStudio

A picture containing building

Description automatically generatedA picture containing outdoor

Description automatically generated

# Galaxy Analysis

The galaxy\_smallmatrix\_labeled\_9d.csv was accessed in order to create Galaxy training and testing models. The target variable was the galaxysentiment feature. This target variable also had 6 possible outcomes: very negative, negative, somewhat negative, somewhat positive, positive, and very positive. The classification algorithms of C5.0 and Random Forest were both found to render the highest performing models. KNN and SVM were both close contenders however they were not evaluated further.

**Galaxy Performance Metrics of Models Created with no Feature Selection nor Feature Engineering Employed**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Accuracy** | **Kappa** | **P-Value** |
| **C5.0** | 0.7577 | 0.5092 | < 2.2e-16 |
| **Random Forest** | 0.7577 | 0.5118 | < 2.2e-16 |
| **KNN** | 0.7404 | 0.4881 | < 2.2e-16 |
| **SVM** | 0.7102 | 0.387 | < 2.2e-16 |

The initial performance metrics for C5.0 and Random Forest were even closer for Galaxy performance than they were for iPhone, therefore both were trialed more extensively using the feature selection and feature engineering approaches listed in the previous section of Modeling Approach.

* When assessing feature engineering, Accuracy and Kappa increased for models with recoded sentiment. Models created with PCA did not improve upon the performance metrics shown by other C5.0 and Random Forest models created unless they were coupled with the feature engineering option of recoded sentiment; this combination was tested for both C5.0 and Random Forest.
* Random Forest and C5.0 models created with correlated features removed outperformed other models created with either no features removed or created with feature selection using RFE or removing near zero variance features.
* Just as in the previous iPhone assessment, we conducted an analysis per target sentiment area assessing Positive Predicted Value and Negative Predicted Value. These values were considered in light of Accuracy and Kappa per model.
* P value for all models was acceptable.

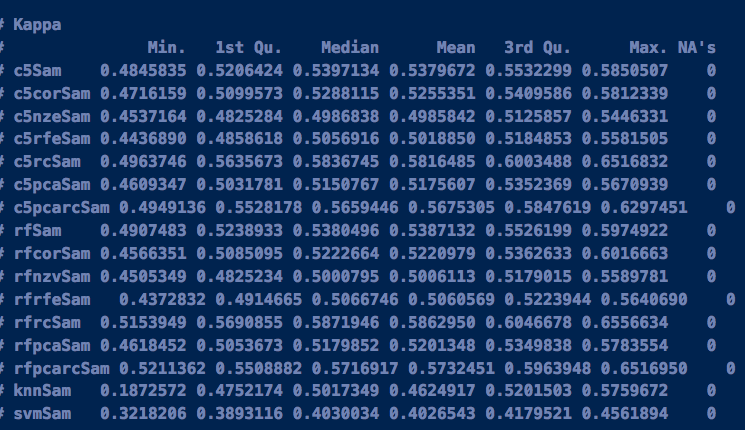
From the chart below we see that both Random Forest and C5.0 with the target variable having recoded sentiment had the highest accuracy and Kappa. **The model that had the highest values, across the board, for both Positive Predicted Value, and Negative Predictive Value was C5.0 with the target variable recoded.** While the Random Forest model had a few higher values, it also had a low Positive Predictive Value for one of the sentiment areas, and thus came in second place.

**­­Galaxy Confusion Matrix Outcomes for Top Model Test Sets (16 models generated, only displaying the top 6)**

[**Table Legend\*\***](#TableLegend)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Kappa |  | Predicted Value Type | Sentiment Type | | | | | |
|  |  |  |  |  | Very negative | Negative | Somewhat negative | Somewhat positive | Positive | Very positive |
| C5 remove  correlated features | 0.7709 | 0.5428 |  | Positive Pred Value | 0.87681 | NaN | 0.916667 | 0.85772 | 0.85315 | 0.7444 |
|  |  |  |  | Negative Predicted Value | 0.95806 | 0.97055 | 0.970626 | 0.96110 | 0.91872 | 0.9141 |
| RF remove  correlated features | 0.7685 | 0.545 |  | Positive Pred Value | 0.88834 | 0.000000 | 0.814815 | 0.84151 | 0.79310 | 0.7474 |
|  |  |  |  | Negative Predicted Value | 0.95675 | 0.970451 | 0.970604 | 0.96423 | 0.92237 | 0.8832 |
| C5 Recoded Sentiment | 0.8479 | 0.6065 |  | Positive Pred Value |  | 0.88941 | 0.785714 | 0.85590 | 0.8421 |  |
|  |  |  |  | Negative Predicted Value |  | 0.92892 | 0.967859 | 0.95718 | 0.9042 |  |
| RF Recoded Sentiment | 0.8502 | 0.6137 |  | Positive Pred Value |  | 0.90488 | 0.550000 | 0.87654 | 0.8431 |  |
|  |  |  |  | Negative Predicted Value |  | 0.92721 | 0.967809 | 0.96170 | 0.9034 |  |
| C5 Recoded Sentiment & PCA | 0.843 | 0.5913 |  | Positive Pred Value |  | 0.88783 | 0.750000 | 0.86036 | 0.8363 |  |
|  |  |  |  | Negative Predicted Value |  | 0.92731 | 0.967358 | 0.95589 | 0.8928 |  |
| RF Recoded Sentiment & PCA | 0.8443 | 0.5987 |  | Positive Pred Value |  | 0.88729 | 0.444444 | 0.87446 | 0.8398 |  |
|  |  |  |  | Negative Predicted Value |  | 0.92677 | 0.968010 | 0.95880 | 0.8859 |  |

Model Training Metrics from RStudio

A picture containing bottle, text

Description automatically generated

# What Went Well & Suggestions for Improvement

This machine learning task went fairly smoothly. There are a few things that could be done to improve the process, however.

I did have RStudio get into a bad state, and I lost a few model objects. I then learned to save off objects frequently using the call of save.image(), and how to restore those objects into RStudio by loading .rPart. This was a valuable lesson, as the last thing that is needed to lose output or data. Time is valuable.

Gather sentiment from web crawls spanning a longer time period. This can be a sample of data spanning a few months or many quarters. Sentiment gathered from one month only may reflect blips in public opinion.

Some algorithms take more time to run, such as Random Forest. The project could be speeded up if more computing power was allocated to this task –whether it be a virtual machine or physical hardware.

Verify that the linguistic corpus we have for analyzing sentiment is adequate. It is possibly skewed to assess negative sentiment more readily than positive or neutral sentiment.

I was surprised that with the exercise we did not combine some of the feature selection techniques. I will research that in the future more as a possible option. However, I did successfully combine two Feature Engineering techniques, and that seemed valid.

**Table Legend\*\***

* Ignore empty cells. In those rows there are only 4 values for sentiment.
* A figure highlighted in yellow denotes an area of concern.
* Values highlighted in green denote superior values (note that many values in this chart are very good).