**Sentiment Analysis - Starbucks**

# Student: Chris Kogut

# Introduction

My primary focus will be to conduct sentiment analysis covering one coffee vendor: Starbucks.

My question will be what do customers like about the company, positive sentiment and what are the shortcomings, negative sentiment.

Proposed solution to the problem:

Analysis will be performed based on the twitter data, and I will use HashTags associated with each company (#starbucks, #timhortons). I will be searching for a specific keywords (positive, negative) that would describe coffee and associated sentiment. In my work, I will be collecting these sentiments and classify polarity of sentiments in these opinions w.r.t. Positive, Negative.

Twitter data will be collected for analysis using Twitter API.

Approach: I’ll be using Dictionary Based approach to analyze data posted by the users. Then polarity classification of this data will done.

R will be used to select and evaluate the data to answer these questions.

Classifier will be built based on the NaiveBayes algorithm – will use 10-fold cross validation.

Also, will use random forest classifier to compare performance of both models.

# Literature Review

I based my research on the following papers collected from various education institutions:

|  |  |  |
| --- | --- | --- |
| 1 | Thumbs up? Sentiment Classification using Machine Learning Techniques | **http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf** |
| 2 | Sentiment Analysis on Twitter | **http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.402.2031&rep=rep1&type=pdf** |
| 3 | Sentimentor: Sentiment Analysis of Twitter Data | http://ceur-ws.org/Vol-917/SDAD2012\_6\_Spencer.pdf |
| 4 | Sentiment Analysis of Twitter Data: A Survey of Techniques | https://arxiv.org/ftp/arxiv/papers/1601/1601.06971.pdf |
|  | Sentiment Mining of Movie Reviews using Random Forest with Tuned Hyperparameters | https://www.researchgate.net/publication/268509189\_Sentiment\_Mining\_of\_Movie\_Reviews\_using\_Random\_Forest\_with\_Tuned\_Hyperparameters |

Sentiment Analysis is sometimes called opinion mining. Based on NLP or machine learning we can characterize the sentiment of a given text (ie. Tweet, movie review, speech etc.)

**Summary of documents:**

1. **Thumbs up? Sentiment Classification using Machine Learning Technique -** Describes process of classifying text not as a topic but either as positive or negative sentiment.

Topic: Naïve Bayes, Maximum Entropy, Support Vector Machines.

Based on the analysis classifiers which were based on the presence of a given feature and not the frequency performed the best. Also unigram features alone not perform as well as unigram and bigrams. Various classifiers were described. Based on the results we concluded that all three classifiers are better at determining topic based classification rather than sentiment classification. Also, for sentiment classification we should, consider placement of the keyword determining sentiment (either beginning of the text or end of the next)

1. **Sentiment Analysis on Twitter**  -Applying sentiment analysis on Twitter is the upcoming trend with researchers recognizing the scientific trials and its potential applications. Microblogging platforms are used by different people to express their opinion about different topics, and it is a valuable source of people’s opinions used for marketing and other purposes. The researchers concluded that Naïve Bayes classifier performed much better than Max Entropy model. The Sentiment Analysis tasks can be done at several levels of granularity, namely, **word level, phrase** or **sentence level**, **document level** and **feature level.** The best results of classification are accomplished by using hybrid of NLP and Machine Learning algorithms.

Tweets have certain characteristics that can be used for classification purposes:

* Message length
* Writing technique
* Topics
* Keywords

Data can be collected and evaluated in real-time. In order to conduct data classification first text needs to go through pre-processing where unnecessary words are removed and other words are tagged as either (**emotion, adverb, verb or adjective**). Then scoring modules determines sentiment of the TEXT based on the tagged words. The overall tweet sentiment was then calculated using a linear equation which incorporated emotion intensifiers as well. Interesting approach still evolving.

1. **Sentimentor: Sentiment Analysis of Twitter Data** – Tweets are broken down into negative, positive, objective. In the paper is described usage of multi nominal naive Bayes classifier to compare unigrams, bigrams and trigrams. It was concluded that bigrams give the best coverage in terms of context and expression of sentiment.
2. **Sentiment Analysis of Twitter Data: A Survey of Techniques**. Document describes in details process around system analysis including:
   * Data gathering (Twitter, Reviews, spam)
   * Pre-processing – data cleanup
   * Feature extraction
     1. Words And Their Frequencies
     2. Parts Of Speech Tags
     3. Opinion Words And Phrases
     4. Position Of Terms
     5. Negation
     6. Syntax Syntactic patter
   * Training – based on positive and negative words
   * Classification (Naïve Bayes)
     1. Supervised learning
        1. Training Set
        2. Test Set
   * Evaluation of sentiment Classification
   * Application of Sentiment
3. **Sentiment Mining of Movie Reviews using Random Forest with Tuned Hyperparameters –** Focuses on the Random Forest classification. Document describes this classification in the context of Heperparameters. Random Forest is a combination of Decision Trees and parameters which are:
4. Number of Trees to construct for the Decision Forest
5. • Number of features to select at random
6. Number of Trees to construct for the Decision Forest
7. • Number of features to select at random
8. Number of Trees to construct for the Decision Forest
9. • Number of features to select at random
10. Number of Trees to construct for the Decision Forest
11. • Number of features to select at random
    * Number of Trees to construct for the Decision Forest
    * Number of Features to select at random
    * Depth of each Tree

**Hyperparameters** are required to be set **manually** which could be time consuming and does not guarantee that it will give good results for the parameter was set manually. Each of the hyperparameters have their own importance and influence towards the output prediction.

Overall Random Forest guarantees good accuracy when it comes to sentiment analysis

# Dataset

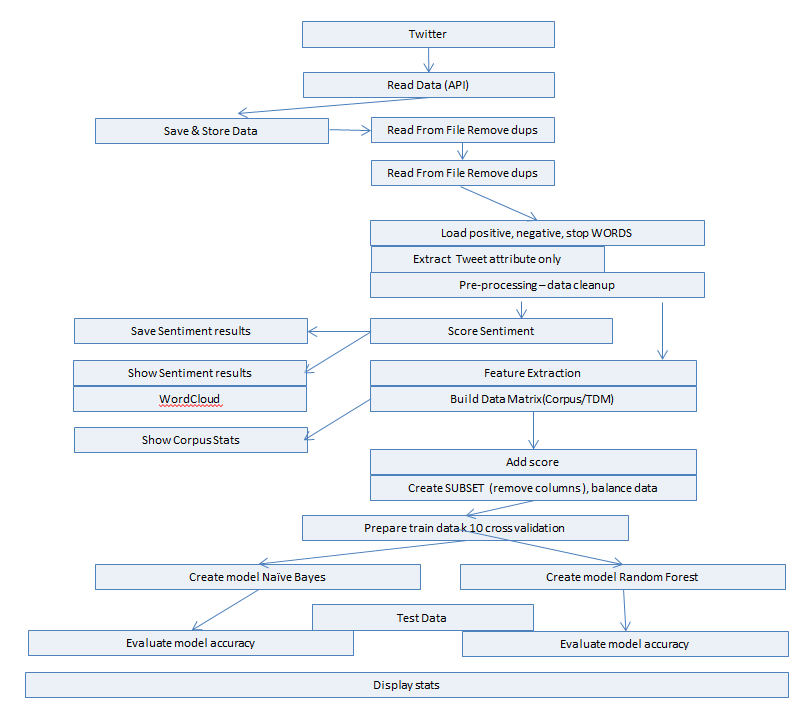
Foe the project I’ll be using standard Twitter feed retrieved real-time using Twitter API. The raw format of the feed is as follow:

|  |  |
| --- | --- |
| **Attributes** | **Used for Analysis** |
| $ created : chr "2017-11-19" "2017-11-19" "2017-11-19" "2017-11-19" ... | Yes |
|  |  |
| $ favoriteCount: num 0 0 0 0 0 0 0 0 0 0 ... |  |
| $ favorited : logi FALSE FALSE FALSE FALSE FALSE FALSE ... |  |
| $ id : chr "932344103643250689" "932343787845771264" "932343371435331586" "932343316187926528" ... |  |
| $ isRetweet : logi FALSE FALSE FALSE FALSE FALSE FALSE ... |  |
| $ latitude : chr NA NA NA NA ... |  |
| $ longitude : chr NA NA NA NA ... |  |
| $ replyToSID : chr NA NA NA NA ... |  |
| $ replyToSN : chr "Starbucks" NA NA NA ... |  |
| $ replyToUID : chr "30973" NA NA NA ... |  |
| $ retweetCount : num 0 0 0 0 0 0 0 0 0 0 ... |  |
| $ retweeted : logi FALSE FALSE FALSE FALSE FALSE FALSE ... |  |
| $ screenName : chr "passingashuman" "McSj1" "geneajourney" "Benjami31250722" ... |  |
| $ statusSource : chr "<a href=\"http://twitter.com/download/android\" |  |
| $ text : chr "@Starbucks I like your new holiday cups | Yes |
| $ truncated : logi FALSE FALSE TRUE FALSE FALSE TRUE ... |  |

**Description of attributes used:**

* Created: date when the Tweet was created
* Text: text used for sentiment analysis

# Approach



# Read Data Using Twitter API

* Open API connection
* Request Tweets based on Tags

# Save Twitter Data

* Save as Data Frame

# Read Complete Data from File

* Remove dups
* Extract only text attribute

# Read Data Using Twitter API

* Open API connection
* Request Tweets based on Tags

# Read supplementary Data

* Negative Words
* Positive Words
* Stop Words

# Pre-Process Text

* remove Http…, #.... @.. patterns
* remove special characters
* remove digits
* remove additional keywords ”starbucks”, “timhortons”….
* remove white spaces

# Perform Sentiment Analysis

* Split sentences into words for each tweet
* Score each work against: positive and negative file(word)
* Create new data frame ‘**score’** (label) corresponding to input text
* Save score to a file

# Show sentiment results

* Word count for positive, negative , neutral, words
* Show Tweet count for each day
* Show scores summary
* Histogram ‘Created’

# Build Text Data Matrix

* Convert tweets into matrix, row is a tweet column words, value (count) of words in each tweet
* Add ‘Score’ column to the matrix (label)
* SUBSET data (columns, rows) - needs calibration
* Save it for WEKA processing

# Prepare train data and use k=10 cross validation

* Caret package, create folds

# Classification – Naïve Bayes

* Run classificator using train data sets (Naïve Bayes)

# Predict result –Naïve Bayes

* Run Predictions

# Evaluate model –Naïve Bayes

* Generate confusion matrix and determine:
  + Accuracy
  + Error Rate
  + Sensitivity (Recall or True positive rate)
  + Precision (Positive predictive value)

# Classification – Random Forest

* Run classificator using train data sets

# Predict result –Random Forest

* Run Predictions

# Evaluate model –Random Forest

* Generate confusion matrix and determine
  + Accuracy
  + Error Rate
  + Sensitivity (Recall or True positive rate)
  + Precision (Positive predictive value)

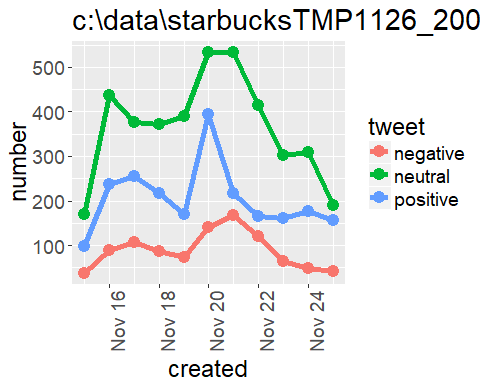
\*) Code for the project can be found here:

<https://github.com/kogutc/CAPSTONE>

# Results:

Processed : **6616** Tweets from Starbucks based on the #starbucks

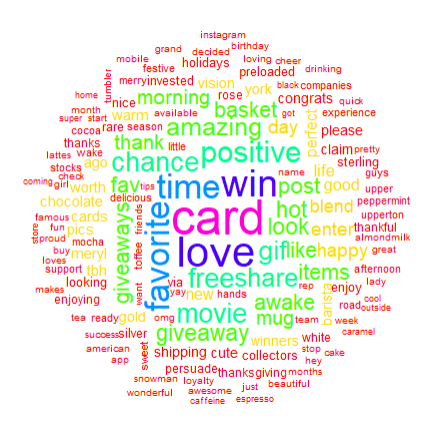
Tweets were captured over a period of 10 days.



# Neutral keywords identified in the feed:



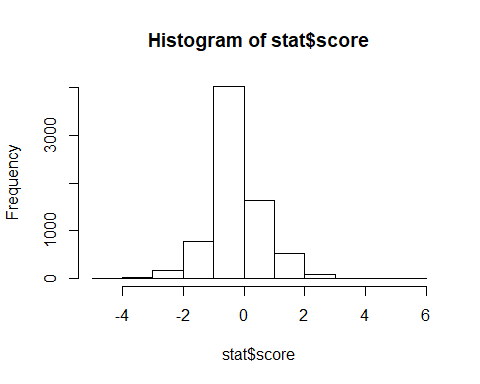
## Positive keywords identified in the feed:

****

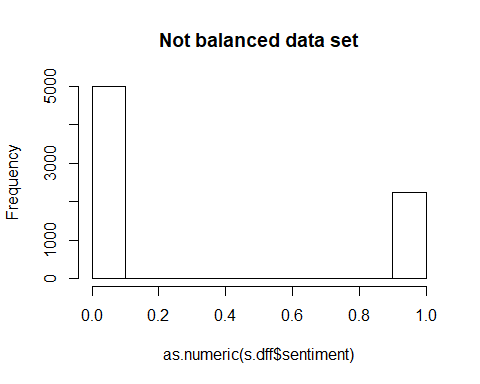
## Negative keywords identified in the feed

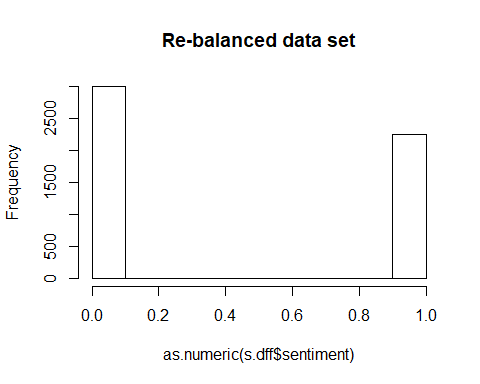


## Identified distribution of the tweets with regards to overall score. Distribution looks good(Symmetric unimodal)



In preparation for further evaluation we associated each tweet with the sentiment (negative = 0 and positive= 1) based on the content. The set is not balanced. In order to balance it we under sampled positive feedback and oversampled negative feedback





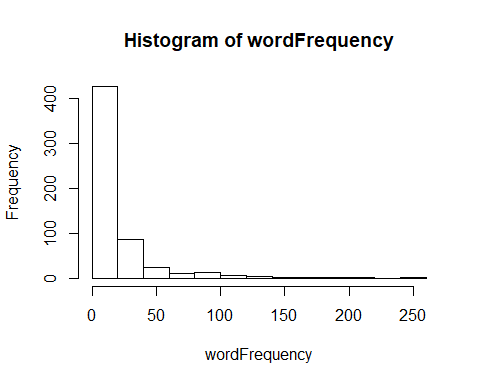
# Selected words based on the frequency of occurrence:

* Min: **10**
* Max: **290**

Mean Word Frequency: **28**

Median Word Frequency **:19**

Standard Deviation: **31.54**



## Performed classification: Naïve Bayes, based on the 70/30 split first.

## Confusion Matrix and Statistics  
##   
## pred  
## 0 1  
## 0 15 876  
## 1 14 671  
##   
## Accuracy : 0.4353   
## 95% CI : (0.4106, 0.4602)  
## No Information Rate : 0.9816   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : -0.0031   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.517241   
## Specificity : 0.433743   
## Pos Pred Value : 0.016835   
## Neg Pred Value : 0.979562   
## Prevalence : 0.018401   
## Detection Rate : 0.009518   
## Detection Prevalence : 0.565355   
## Balanced Accuracy : 0.475492   
##   
## 'Positive' Class : 0

## Then performed 10 fold cross validation Naïve Bayes

Accuracy for each fold:

## [1] 0.9676190 0.9733333 0.7371429 0.9485714 0.3409524 0.1085714 0.1333333  
## [8] 0.1009524 0.1504762 0.1368821

Mean Value of the accuracy: 0.4597835 (46%)

## Next evaluated data using Random Forest:

Params: ntree = 100, nodesize = 5

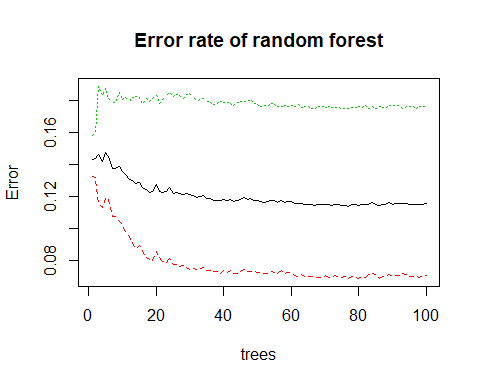
## ntree OOB 1 2  
## 50: 11.70% 7.24% 17.72%  
## 100: 11.56% 7.05% 17.66%

## Type of random forest: classification  
## Number of trees: 100  
## No. of variables tried at each split: 24  
##   
## OOB estimate of error rate: 11.56%

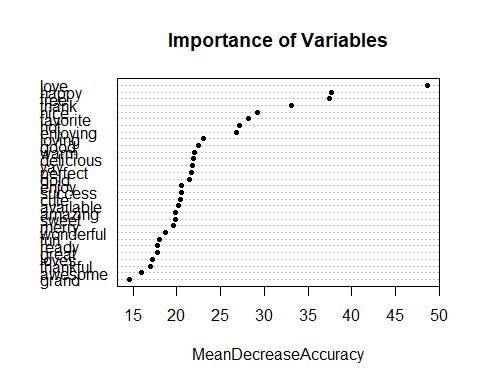
## Confusion matrix:  
## 0 1 class.error  
## 0 1963 149 0.07054924  
## 1 276 1287 0.17658349

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 832 59  
## 1 122 563  
##   
## Accuracy : 0.8852 (89%)   
## 95% CI : (0.8684, 0.9005)  
## No Information Rate : 0.6053   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7638   
## Mcnemar's Test P-Value : 4.057e-06   
##   
## Sensitivity : 0.8721   
## Specificity : 0.9051   
## Pos Pred Value : 0.9338   
## Neg Pred Value : 0.8219   
## Prevalence : 0.6053   
## Detection Rate : 0.5279   
## Detection Prevalence : 0.5654   
## Balanced Accuracy : 0.8886   
##   
## 'Positive' Class : 0   
##

## Error Rate for Random Forest:



### And we also identified Importance Variables:



### Also performed further analysis using Weka to evaluate sentiment based on sentiment class with three values (negative neutral positive) > only for Starbucks. Random forest is a winner.

**Random Forest 10 – fold cross validation - - using WEKA 3 states , negative, neutral and positive**

**Correctly Classified Instances 5991 90.5532 %**

**Incorrectly Classified Instances 625 9.4468 %**

**Kappa statistic 0.8577**

**Mean absolute error 0.1237**

**Root mean squared error 0.2302**

**Relative absolute error 27.9289 %**

**Root relative squared error 48.9261 %**

**Total Number of Instances 6616**

**TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class**

**0.946 0.038 0.913 0.946 0.929 0.899 0.985 0.972 -1**

**0.907 0.085 0.860 0.907 0.883 0.814 0.959 0.933 0**

**0.869 0.021 0.955 0.869 0.910 0.869 0.978 0.969 1**

**---------------------------------------------------------------------------------**

**0.906 0.049 0.908 0.906 0.906 0.857 0.973 0.957**

**=== Confusion Matrix ===**

**a b c <-- classified as**

**1844 100 6 | a = -1**

**138 2194 86 | b = 0**

**38 257 1953 | c = 1**

**Sample Variables used in tree construction:**

**"love" "favorite" "amazing" "hot" "giveaways" "blend" "free" "thank" "winners" "happy" "worth"**

**[12] "positive" "song" "perfect"**

**Naïve Bayes – fold cross validation - using WEKA 3 states , negative, neutral and positive**

=== Stratified cross-validation ===

Correctly Classified Instances 4272 **64.5707 %**

Incorrectly Classified Instances 2344 **35.4293 %**

Kappa statistic 0.4615

Mean absolute error 0.3284

Root mean squared error 0.411

Relative absolute error 74.1765 %

Root relative squared error 87.365 %

Total Number of Instances 6616

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.580 0.087 0.736 0.580 0.649 0.533 0.858 0.713 -1

0.757 0.348 0.556 0.757 0.641 0.394 0.786 0.666 0

0.583 0.109 0.733 0.583 0.649 0.505 0.862 0.786 1

Weighted Avg. 0.646 0.190 0.669 0.646 0.646 0.473 0.833 0.721

=== Confusion Matrix ===

a b c <-- classified as

1131 679 140 | a = -1

250 1830 338 | b = 0

155 782 1311 | c = 1

# Conclusions

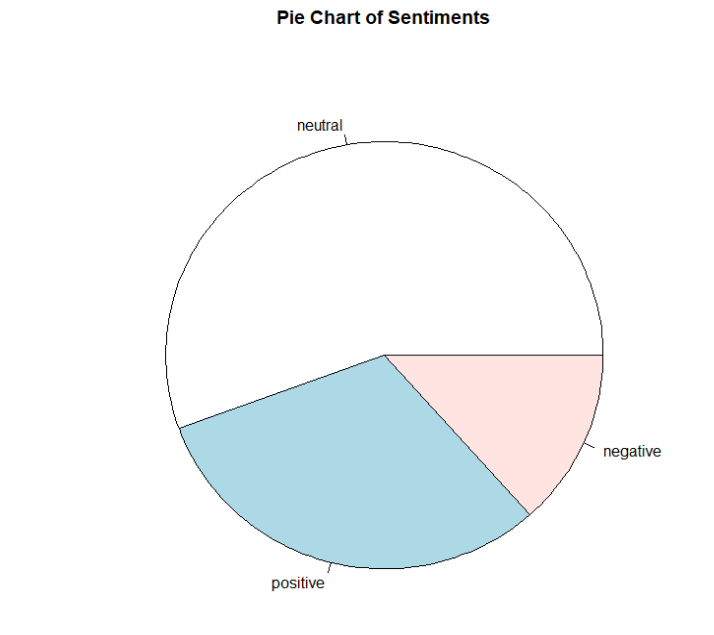
Usage of Twitter data can be a good tool for evaluation of customer sentiment. In this project I used number of classifiers to see which one works best for the classification of the Twitter data.

For this type of data Random Forest classification performed much better than Naïve Bayes.

Best accuracy achieved for Class (Negative Positive) was 90% using Random Forest.

### Feedback about data and recommendation to the management

Overall sentiment is: **positive**



To improve customer satisfaction the following changes are recommended:

* Decrease wait time
* Ensure right temperature of the beverages
* Professional customer service

Customers currently like the following services:

* Gift cards
* Free giveaways
* Positive atmosphere