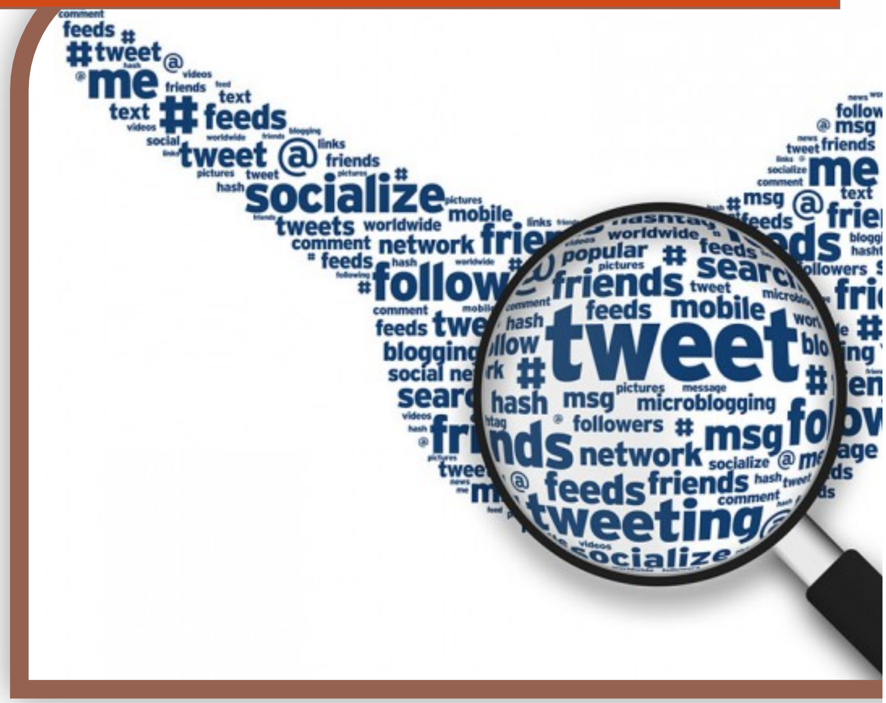
Project: Tweets Classification



Tyag Raj, Aspiring Data Scientist
Edwisor

31-03-2017

Table of Contents

Brief Introduction to Project

In this project, I will attempt to classify Tweets in to two categories; Sarcastic and Non Sarcastic by training them in two Supervised machine learning algorithms; KNN and Naïve Bayes. You will see bar chart and Word Cloud to understand the most frequently used words in Tweets in the whole data set and comparison of most common words used in sarcastic Tweets and non sarcastic Tweets. This report will also try to show before and after the preprocessing Tweets.

This Project covers understanding the data, preprocessing the Tweets, Training the models to classify the untrained or test data. Based on both models and understanding, I will try to show some Insight from the data. In the later part of project, I will try to compare both models and each algorithm's pros and cons and which algorithm is suited best for Tweets Classification along with the understanding of the both algorithms.

Data

Data consists of 3 Variables i.e. "ID" "Tweet" & "Labels" and 91298 observations. Labels are Sarcastic and Non-Sarcastic. With the help of table(Tweets$label) It can be seen that Sarcastic and Non Sarcastic Tweets are labelled as follows.

non-sarcastic	*sarcastic*
39998	*51300*

non-sarcastic	*sarcastic*
0.4381038	*0.5618962*

39998 Tweets are labelled as non-sarcastic and 51300 Tweets are labelled as sarcastic, later figure shows us that it has 44:56 ratios.

Upon checking the structure of data we find out that labels are classified as character but it should be factor so we change it to factor.

Preprocessing of data

First Tweets are selected and converted into a corpus. Tm library is used to clean the corpus such as removing numbers, punctuations, capital letters converted in to lower, stop words are removed, unnecessary white spaces are removed. Below is shown how a tweet looked before cleaning and how that same tweet looked after cleaning.

Tweet 1 in corpus.

```
docs$content[1]
[1] "b'oh yea that makes sense '"
```

Tweet 1 after cleaning process.

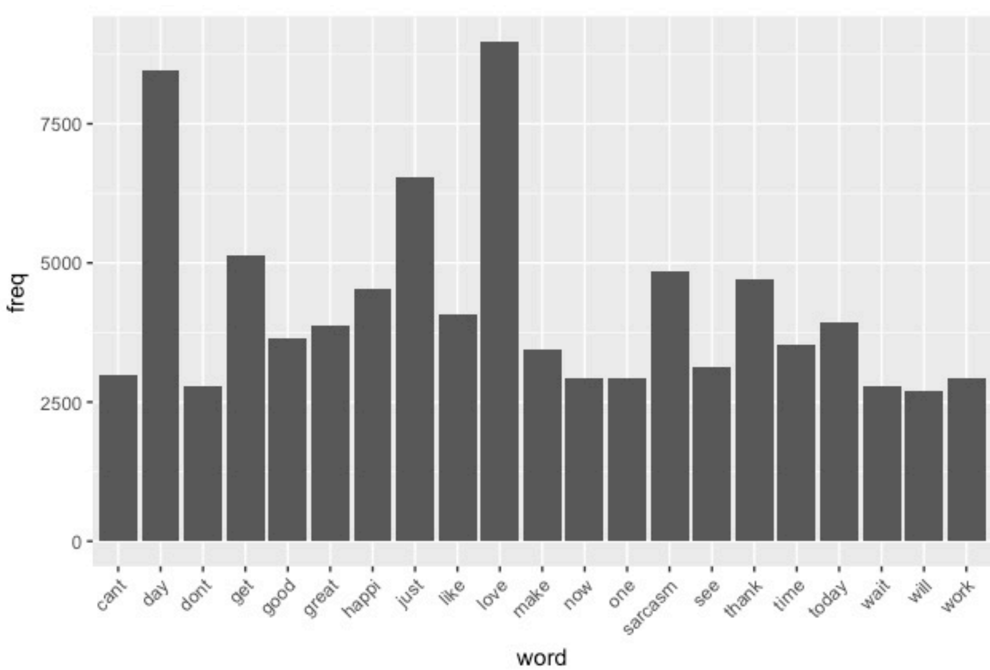
```
Docs$content[1]
[1] "oh yea make sens"
```

```
"make"  "will"  "cant"  "like"  "can"  "peopl"  "love"  "just"  "thank"

"time"  "get"  "day"  "sarcasm" "today"  "look"  "dont"  "good"  "know"

"happi"  "great"  "work"  "now"  "one"  "see"  "new"  "wait"
```

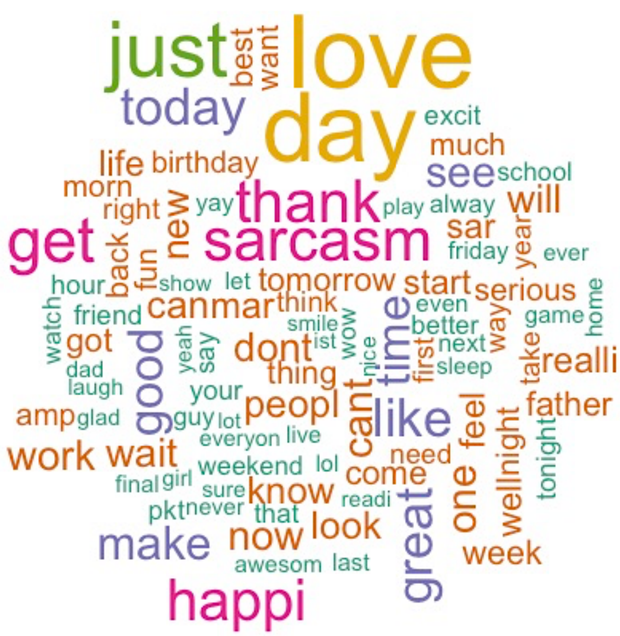
Above 26 terms were used more than 2500 times in our dataset.



The above bar chart shows Love is the most used word in our data set. It can also be seen that Can't and don't are also very frequently. It gives us a sense that when people tweet with sarcasm they use negative words often.

Following word cloud helps us in representing the Most Frequently used words in Tweets

Below word cloud shows the maximum words used in the Tweets



Insight from the data

After Comparing Sarcastic and Non sarcastic Tweets by the help of below charts, It's clear that both Tweets use the similar words more often than not.

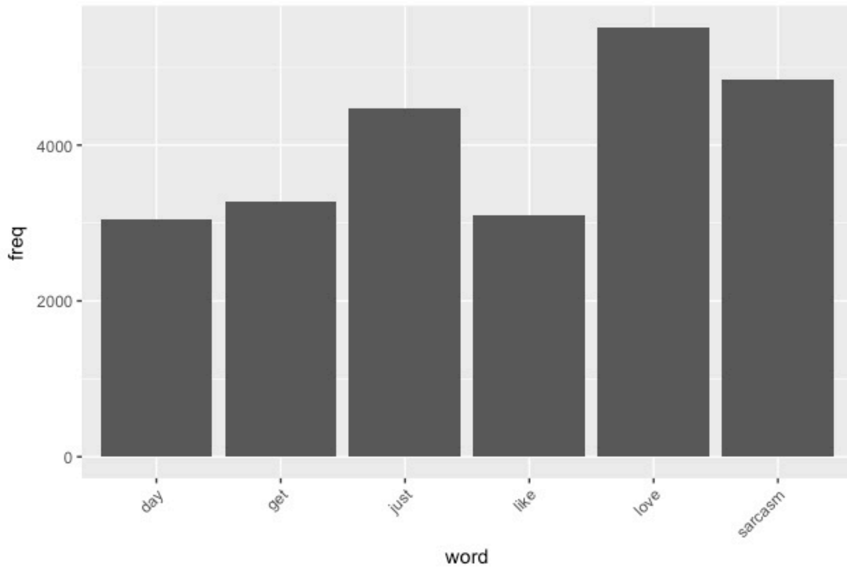


Figure 1 Non Sarcasm Label Tweets

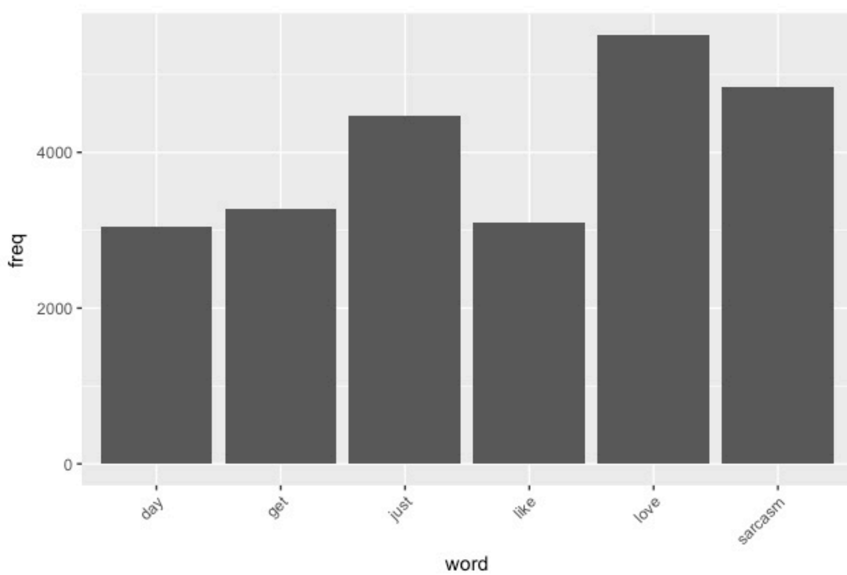


Figure 2 Sarcasm Tweets Label

Both Non Sarcastic and Sarcastic most used words are same. It's very difficult to say which words are more often used in sarcastic vs non-sarcastic Tweets.

Figure 3 Non Sarcastic Word cloud

Figure 4 Sarcastic Word Cloud

As the bar Charts showed. Word cloud shows the same picture. Most frequently used words are similar and hard to differentiate.

Model 1. KNN

For KNN model only 10% sample of Original data used to build a classification model. Proportionate ratio of sarcastic and non sarcastic labels was measured to make sure sample data is not biased and doesn't not affect the performance of the Knn Model.

Confusion Matrix and Statistics

Prediction	Reference: non-sarcastic	Reference: sarcastic
non-sarcastic	751	593
sarcastic	232	706

Accuracy: 0.6385
95% CI: (0.6184, 0.6582)
No Information Rate: 0.5692
P-Value [Acc > NIR]: 9.582e-12

Kappa: 0.2943
Mcnemar's Test P-Value: < 2.2e-16

Sensitivity: 0.7640
Specificity: 0.5435
Pos Pred Value: 0.5588
Neg Pred Value: 0.7527
Prevalence: 0.4308
Detection Rate: 0.3291
Detection Prevalence: 0.5890
Balanced Accuracy: 0.6537

'Positive' Class: non-sarcastic

Model 2. Naive Bayes

20% sample data is used for this model to classify tweets in to sarcasm and non-sarcasm. Proportionate ratio of sarcastic and non sarcastic labels was measured to make sure sample data is not biased and doesn't not affect the performance of the Naive Bayes Model.

Confusion Matrix and Statistics

Prediction	Reference: non-sarcastic	Reference: sarcastic
non-sarcastic	1032	1466
sarcastic	1240	1521

Accuracy: 0.4855
95% CI: (0.4719, 0.4991)
No Information Rate: 0.568
P-Value [Acc > NIR] : 1

Kappa: -0.0361
Mcnemar's Test P-Value: 1.523e-05

Sensitivity: 0.4542
Specificity: 0.5092
Pos Pred Value: 0.4131
Neg Pred Value: 0.5509
Prevalence: 0.4320
Detection Rate: 0.1962
Detection Prevalence: 0.4750
Balanced Accuracy: 0.4817

'Positive' Class: non-sarcastic

Naive Bayes Model achieved 49% Accuracy with performance enhancement using Laplace = 1.

Comparison of Naïve Bayes And Knn algorithms		
	Naïve Bayes	**KNN**
Pros	It is Simple, Fast and Highly Effective.	It is also simple and effective.
	It require small data to train.	It trains pretty quickly.
	It deals easily with missing data.	
Cons	Relies on often faulty assumptions.	It doesn't build a model.
	Estimated probabilities are less reliable than predicted classes.	It requires large memory.
		Slow Classification phase.

Based on both of my Models, KNN Model got better accuracy than Naïve Bayes Model with less sample of data. Knn proved to be a better model for Tweet Classification.

Recommendation

Based on this Project, my recommendation is to use Knn Model to classify Tweets in Sarcastic and Non-Sarcastic. Most important words used in Sarcastic Tweets are Love, Just and Sarcasm. Although When most common words used in Sarcastic Tweets Vs Non-Sarcastic Tweets, it seems both tweets use similar words often.