

Tech Mahindra - Solution 2

Business Problem Framing

The objective of this task is to detect hate speech in tweets. For the sake of simplicity, we say a tweet contains hate speech if it has a racist or sexist sentiment associated with it. So, the task is to classify racist or sexist tweets from other tweets.

Analytic problem framing

The problem involves analyzing textual information. Such problems are solved using Natural Language Programming (NLP) approach.

The problem can be solved using machine learning techniques such as Naive Bayes algorithm, Bagging or boosting method.

In this analysis we use Naive Bayes and RandomForest, a bagging technique.

Data

A series of tweets are shared in csv file.

Analysis

Create Corpus

A corpus is a large and structured set of text.

A dataset may have textual information along with some metadata such as time, user etc. A corpus is a particular format of data structure that separate source (actual text to analyze) and meta information in a structured manner.

Data Cleaning

We apply the following steps to harmonize data.

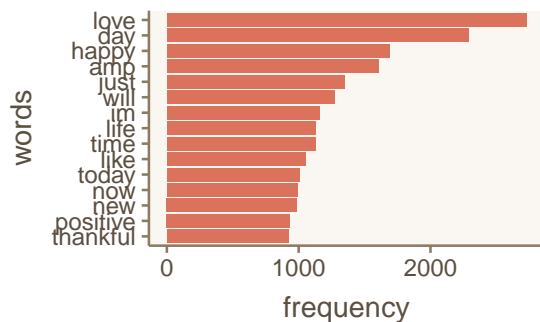
- Convert all tweets to lower case
- Remove punctuation
- Remove numbers
- Remove stopwords such as of, the etc.
- Remove the word user and u - since @user in a tweet is a stopword
- Remove extra space in between words or sentence.

Note: For this analysis we have removed number, we can use `replace_number()` from the `qdap` package to convert figures to words.

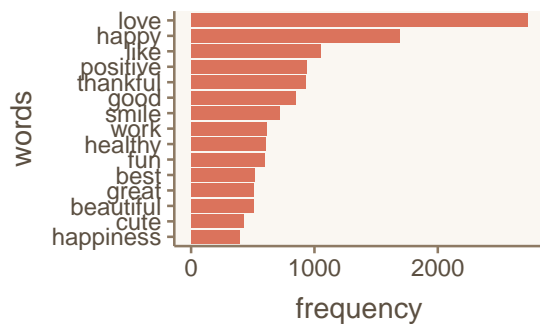
Visualization

Some exploratory textual analysis.

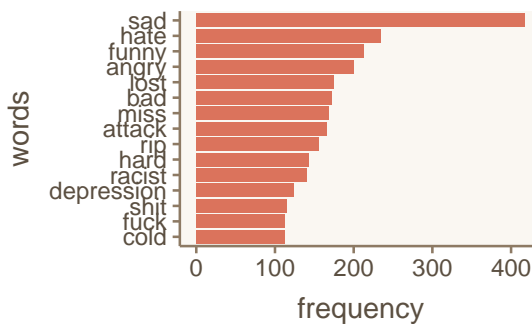
Top 15 words in the tweet



Top 15 positive words



Top 15 negative words



Create rectangular structure - DTM

Document-Term Matrix (DTM) is a matrix structure, where document text (each tweet in our analysis) are rows and terms (each word of tweets) are columns

Example of DTM:-

Text (tweets)	This	is	a	text	some	more
This is a text	1	1	1	1	0	0
Some more text	0	0	0	1	0	0

Split data

Split data in train set (75%) and test set (25%).

Feature engineering

There are a total of 24,000 words in entire corpus. Most of the are just noise. Hence will ignore words that occur less than 10 times in the entire corpus.

This model assumes for a single tweet repeated word does not make any sense. Therefore, we convert the entire structure to with boolean values. Eg:- Thank you very very very much means Thank you very much.

Modeling

Naive Bayes Classifier

Naive Bayes classifier is a family of simple **probabilistic classifiers** based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

The most common use of naive Bayes is for document classification.

Refer here to perform a Naive Bayes calculation by hand.

Confusion Matrix - Naive Bayes

		Actual	
Predicted	0	7204	238
	1	197	324

Terms	estimate
accuracy	94.53723
kappa	56.90763
sensitivity	97.33820
specificity	57.65125
pos_pred_value	96.80193
neg_pred_value	62.18810
precision	96.80193
recall	97.33820
f1	97.06933
prevalence	92.94236
detection_rate	90.46842
detection_prevalence	93.45724
balanced_accuracy	77.49472

F1 Score - Naive Bayes

.metric	.estimate
f_meas	97.37712

Randomforest model

Random Forests work by training many Decision Trees on random subsets of the features, then averaging/voting out their predictions.

Confusion Matrix - RandomForest

Predicted	Actual	
	0	1
0	6980	529
1	421	33

term	estimate
accuracy	88.06982
kappa	00.20134
sensitivity	94.31158
specificity	05.87189
pos_pred_value	92.95512
neg_pred_value	07.26872
precision	92.95512
recall	94.31158
f1	93.62844
prevalence	92.94236
detection_rate	87.65541
detection_prevalence	94.29863
balanced_accuracy	50.09173

F1 Score - RandomForest

.metric	.estimate
f_meas	93.62844

Deployment and Maintenance

Deployment

The model can be deployed as an API.

The advantage of using a API is that the model can be used by any software - python's flask app, html webpage, Java application to name a few. By creating an API, we empower our model to be leveraged by other services.

With the help of `plumber` and `docker` we can easily convert our model into a REST API.

Maintenance

Once deployed, the whole system needs proper monitoring and maintenance.