Introduction:

Social networking sites has evolved as a major platform for people to express their opinions and sentiments on variety of topics like movies, politics issues, daily chores etc. This has made Social networking sites e.g. Facebook, Twitter rich with data, which can be used for opinion and sentiment mining. But mining of such unstructured text can be cumbersome and challenging. Hence I am motivated in this assignment to develop a model for sentiment and opinion analysis for twitter data.

Classification:

Classification lies in heart of both human and Machine intelligence [1] and it has been used in supervised learning to predict the outcomes, given one or more inputs. The outcomes are often labels or categories. For example in our case a tweet is classified based on its sentiment i.e. “positive” or “negative”. I am using K-nearest neighbour (kNN) for classification. kNN is a typical example based classifier that does not build an explicit, declarative representation of the category, but relies on the category labels attached to the training documents similar to the test document. Given a test document d, the system finds the k nearest neighbours among training documents. The similarity score of each nearest neighbour document to the test document is used as the weight of the classes of the neighbour document[2].

Implementation & Experimental results

The data for experiment was retrieved using function searchTwitter from R package twitteR. Twitter Standard search API searches against a sampling of recent Tweets published in the past 7 days only. Hence I have choose topics like “star war”, “assignment”, “best movie”, “The Greatest Showman”, “the post”, “Stranger Things” and some few which are released currently in conjunction with emoticons :), :(to retrieve tweets as shown in below excerpt.

```r
#### Collect data with positive sentiment
starwar <- searchTwitter(’star+wars+:)’, lang="en", n=1000, resultType="recent")
#### Collect data with negative sentiment
showman <- searchTwitter(’The Greatest Showman+:(’, lang="en", n=200, resultType="recent")
```

Cleaning and processing data

The retrieved tweets consists of various attributes e.g. created, URLS, texts etc. and we are considering message text only. The tweets were then processed and cleaned with below steps:

- Remove duplicate tweets
- Label tweets as positive or negative with emoticons :), :-), :)), :D, :-)) or :(, :((, :-(, :-((respectively.
- Merging all tweets to a single list.
- Clean tweets removing numbers, punctuation, urls, unwanted symbols, emoticons and lowering case.
- Create corpus and generate term document matrix.

Model building and evaluation

The final data is divided into training set covering 75% of data for training the model and testing set covering 25% of data for evaluation of the model. Predictive model for classification was build using R package “caret” with 10-fold cross validation repeated 3 times. The best value for k, model used is 1.

The performance of model is evaluated by calculating various metrics like precision (P), recall(R), F1 Score. Precision is the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved. The two measures are sometimes used together in the F1 score, a measure of a test’s accuracy.

The precision (P), recall(R), F1 score and accuracy (A) are evaluated as:

- $P = TP / (TP+FP)$
- $R = TP / (TP+FN)$
- $A = (TP + TN) / (TP + TN + FP + FN)$
- $F1 = 2 * ((P * R)/(P + R))$

where TP, TN, FP, FN implies true positive, true negative, false positive and false negative respectively

Table 1: Precision/Recall/F1-Score values

S.N	Total Tweets	Execution Time(m)	Precision	Recall	F1-Score	Accuracy
1	2888	9.58	0.833	0.929	0.878	0.864

The precision for our model is 0.833, means that 83.3 percent of the tweets are classified as positive sentiment, while 16.7 percent of those tweets have been misidentified as positive sentiment and recall is 0.929, means that 92.9 percent of the tweets are identified positive sentiment which were in fact positive sentiment.

Discussion

Overall our model performance was good with each performance measure greater than 0.8. Whenever using machine learning to solve real world problems we should be clear on what performance metric defines success for us i.e. either recall or precision or both. And I have not found de facto standard for precision or recall around the web, but in general it is better to have high recall and precision.

The use of emoticons for analyzing sentiment of tweets are good but for best analysis we need to integrate natural language processing methods and symbol analysis[3]. Currently we are using Unigrams, but considering bigrams or trigrams might increase our performance. In addition we are ignoring the context in tweets, so use of more sophisticated algorithm like SVM with POS-tagging(Parts of Speech) may help us to gain more performance.

Increasing our sample size also helps us to gain performance but computation cost is quite high with kNN especially when the size of the training set grows, because we need to compute distance of each query instance to all training samples and some indexing (e.g K-D tree) or PCA may reduce this computation cost.

Moreover, most classification techniques used so far are for texts with long lengths, but tweets are shot, summarized and precise due to length limitation i.e 140 characters and to make a significant progress in sentiment analysis using tweets, we still need novel ideas.

Time Distribution:

S.N	Topic	Time in hours
1	Literature review & State of the arts	5
2	Coding and Testing	6
3	Report Preparation & Miscellenous	4

Note: *Most time in Coding and Testing section went on testing and improving performance of model.*

References

1. James H. Martin, Daniel Jurafsky &. 2017. "Speech and Language Processing" Book.
2. Songbo Tan, Jin Zhang 2008. "An Empirical Study of Sentiment Analysis for Chinese Documents" Article.
3. Wolny, Wieslaw 2016. "Sentiment Analysis of Twitter Data Using Emoticons and Emoji Ideograms" Article.
4. http://rmarkdown.rstudio.com/authoring_bibliographies_and_citations.html#citation_styles
5. http://www.scholarpedia.org/article/K-nearest_neighbor
6. https://en.wikipedia.org/wiki/Confusion_matrix
7. https://en.wikipedia.org/wiki/Precision_and_recall
8. https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
9. [A Detailed Introduction to K-Nearest Neighbor KNN Algorithm] (https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/), Web Article.
10. http://www.cs.uvm.edu/~xwu/kdd/kNN-11.ppt

Source Code of the assignment can be found on github