Twitter Sentiment Analysis

Introduction to Twitter Sentiment Analysis

Abstract

- The blast of Web 2.0 has prompted expanded action in Podcasting, Blogging, Tagging, Contributing to RSS, Social Bookmarking, and Social Networking.
- Subsequently there has been a sudden increase of enthusiasm for individuals to mine these tremendous assets of information for suppositions. Sentiment analysis or Opinion Mining is mining of sentiment polarities from online social media.
- In this project we will talk about a procedure which permits use and understanding of twitter information for sentiment analysis.
- We perform several steps of text pre-processing, and then experiment with multiple classification mechanisms.
- Using a dataset of 50000 tweets and TFIDF features, we comparison the accuracy obtained using various classifiers for this task.
- We find that linear SVMs provide us the best accuracy results among the various classifiers tried.
- Sentiment analysis classifier could be useful for many applications like market analysis of different features of a new product or public opinion for a new movie or speech by a political candidate.

Motivation for the project

- Sentiment analysis (SA), also known as opinion mining is the process of classifying the emotion conveyed by a text, for example as negative, positive or neutral.
- Also called as Opinion extraction, Opinion mining, Sentiment mining, Subjectivity analysis
- *Applications*
  - *Movie*: is this review positive or negative?
  - *Products*: what do people think about the new iPhone?
  - *Public sentiment*: how is consumer confidence? Is despair increasing?
  - *Politics*: what do people think about this candidate or issue?
  - *Prediction*: predict election outcomes or market trends from sentiment

Large data for sentiment analysis on Twitter

- A popular social medium is Twitter, a micro-blogging site that allows users to write textual entries of up to 140 characters, commonly referred to as tweets.
- As of June 2015, Twitter has over 302 million monthly active users according to their homepage, whereof approximately 88 % have their tweets freely readable.
- Data created by Twitter is made available through Twitter’s API, and represents a realtime information stream of opinionated data.

Challenges in handling Twitter data

- Tweets often contain misspellings, and the constrictive limit of 140 characters encourages slang and abbreviations.
- Unconventional linguistic means are also used, such as capitalization or elongation of words to show emphasis.
- Additionally, tweets contain special features like emoticons and hashtags that may have an analytical value.
- Hashtags are labels used for search and categorization, and are included in the text prepended by a “#”.
- Emoticons are expressions of emotion, and can either be written as a string of characters e.g., “:-)”, or as a unicode symbol.
- Finally, if a tweet is a reply or is directed to another Twitter user, mentions can be used by prepending a username with “@”.

Problem Definition

- Given: Tweet
- Predict: Sentiment polarity of the tweet – positive vs negative.

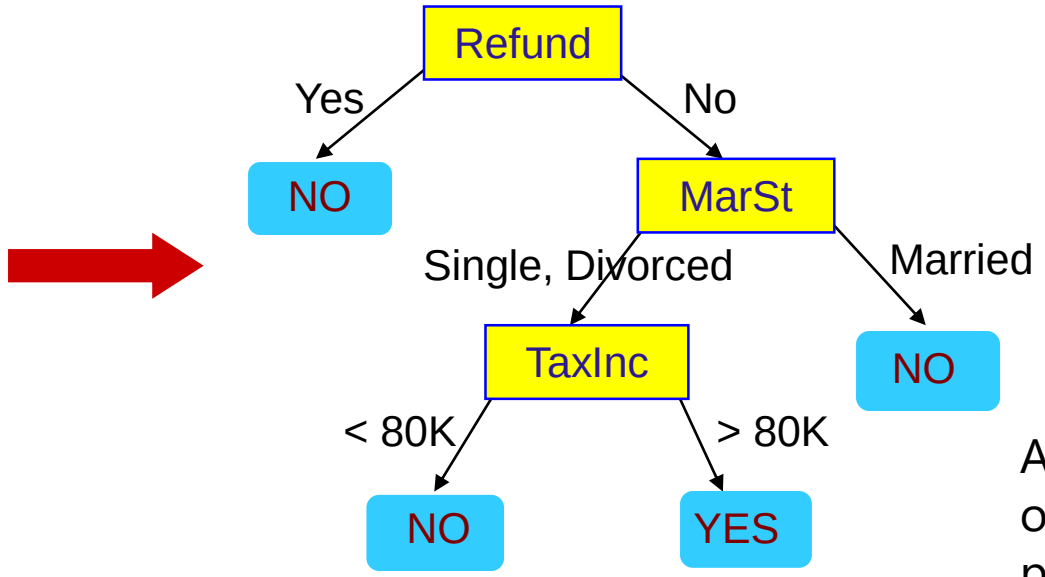
Basic ML and NLP techniques and tools needed for Twitter Sentiment Analysis

Approaches: Machine Learning (Decision Trees)

- Spam detection. input: documents; classes: spam/ham
- OCR. input: images; classes: characters
- Medical diagnosis. input: symptoms; classes: diseases
- Autograder. input: codes; output: grades

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Training Data

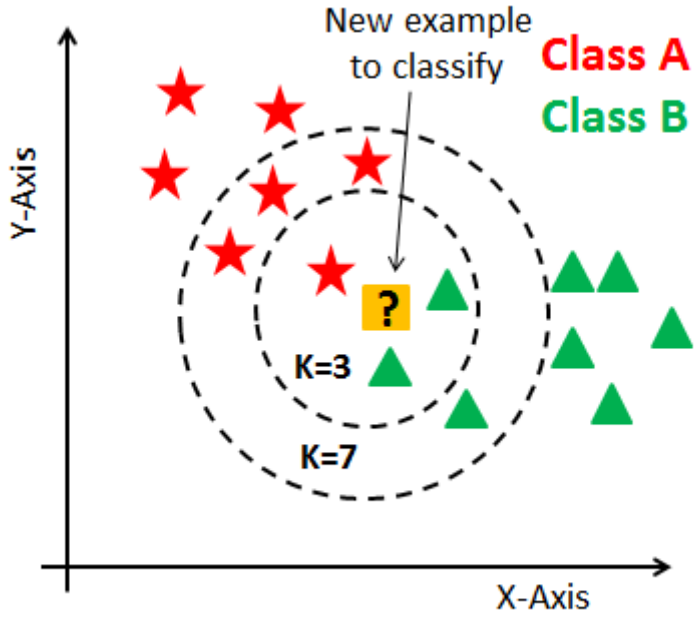


Model: Decision Tree

A tree-like graph or model of decisions and their possible consequences

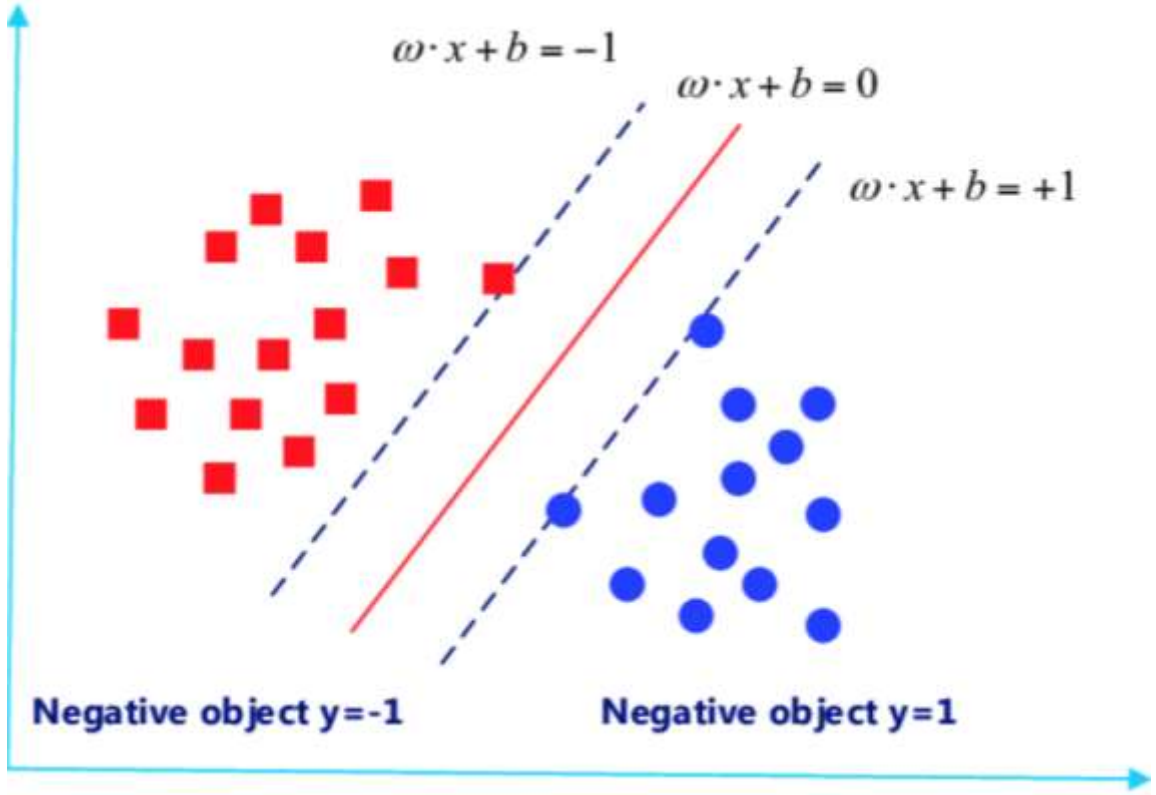
Predict whether a person will cheat or not?

Approaches: Machine Learning (KNNs)



- Pros:
  - Simple process
  - Quick to train
- Cons
  - Curse of dimensionality
  - Slow to test
  - Requires more memory
  - Missing values need to be handled separately

Approaches: Machine Learning (SVMs)

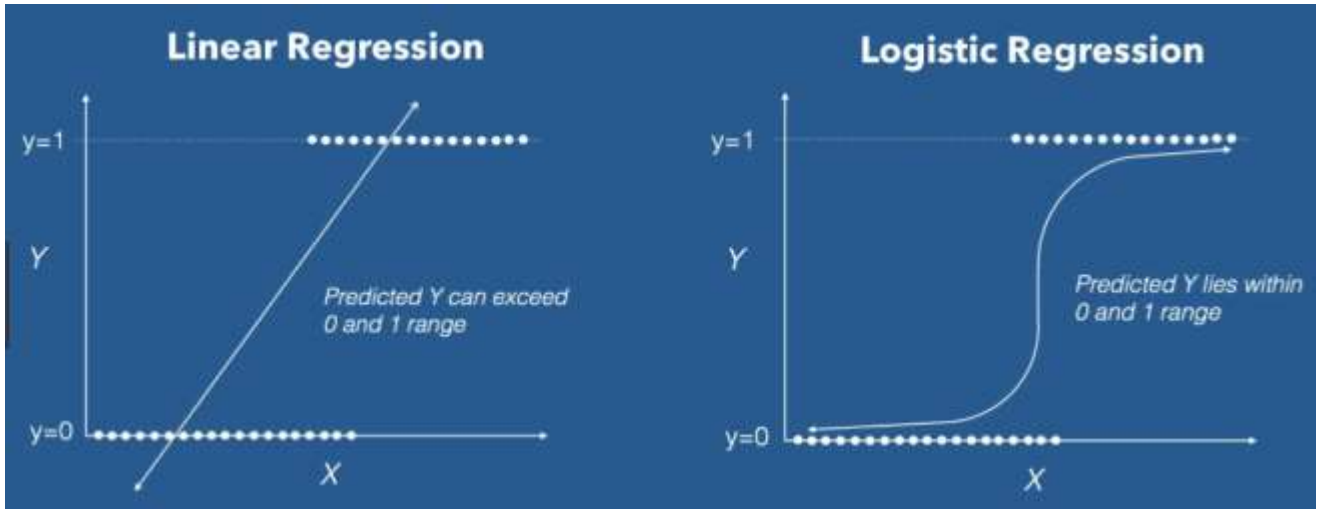


- *SVM searches for the hyperplane with the largest margin*, i.e., maximum marginal hyperplane (MMH) using constrained convex quadratic optimization
- Pros: 1. High accuracy 2. Nice theoretical guarantees regarding overfitting 3. With an appropriate kernel they can work well even if your data is not linearly separable in the base feature space 4. Good for high-dimensional data (like text)
- Cons: 1. Memory-intensive 2. Hard to interpret 3. Annoying to run and tune (long training time) 4. Not easy to incorporate domain knowledge (priors)
- Kernel Trick

Approaches: Machine Learning (Ensemble Learning)

- Use a combination of models to increase accuracy
- Popular ensemble methods
    - Bagging: averaging the prediction over a collection of classifiers
    - Boosting: weighted vote with a collection of iteratively learned classifiers
        - AdaBoost
        - Gradient Boosting
    - Random Forest: Each classifier in the ensemble is a decision tree classifier and is generated using a random selection of attributes at each node to determine the split
- Pros: Highly effective in presence of large amount of training data
- Cons: Take long time to train

Approaches: Machine Learning (Logistic Regression)



For multi-class categorization

$$P(y = k|X) = \frac{\exp(w_k^T X)}{\sum_{j=1}^{K} \exp(w_j^T X)}$$

$$P(y = k|X) \propto \exp(w_k^T X)$$

$$p(y|x) = \sigma(w^T X) = \frac{1}{1 + \exp(-w^T X)}$$

Approaches: Machine Learning (Naïve Bayes)

Likelihood

Prior

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

Normalization Constant

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}}\ P(c_j)\prod_i P(x_i \mid c_j)$$

$$= \underset{c_j \in C}{\operatorname{argmax}}\ P(c_j)P(x_1 = \text{"our"} \mid c_j)\cdots P(x_n = \text{"text"} \mid c_j)$$

Approaches: NLP (TF-IDF)

- Term frequency-inverse document frequency (TF-IDF) is a common term weighting scheme for the bag-of-words model, which lets us identify words in a collection of documents that can guide in deciding a document's topic.

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}, \qquad |\{d \in D : t \in d\}| \neq 0$$

- Here D is the entire corpus of documents, and N is the total number of documents in the corpus. As the number of documents where a term appears increases, the ratio inside the logarithm will decrease towards 1, making the idf approach 0.

Approaches: NLP (TF-IDF example)

- Consider a document containing 100 words wherein the word 'Cauvery' appears 3 times.
- The term frequency (tf) for 'Cauvery' is then TF = (3 / 100) = 0.03.
- Now, assume we have 10 million documents and the word 'Cauvery' appears in 1000 of these. Then, the inverse document frequency (idf) is calculated as IDF = log(10,000,000 / 1,000) = 4.
- Thus, the Tf-idf weight is the product of these quantities TF-IDF = 0.03 * 4 = 0.12.

Tools: Scikit Learn

- Scikit-learn [Pedregosa et al., 2011] is a framework for the Python programming language that offers machine learning models as well as tools for performing preprocessing and data analysis.
- Transformer objects are an essential part of performing machine learning with Scikit-learn, and are objects that may “clean, reduce, expand, or generate feature representations”.
- Scikit-learn provides a framework for pipelining machine learning tools, making it easy to chain tasks such as preprocessing and feature extraction together with a machine learning algorithm in a tidy manner.
- One of the benefits of using the Pipeline framework is that it allows performing a parameter grid search across all the estimators in the Pipeline.

Tools: Pandas

- Pandas [McKinney, 2012] is an open-source library providing high-performance data structures and data analysis tools for the Python programming language.
- It also includes tools for efficiently reading and writing data between in-memory data structures and different textual file formats, such as comma-separated value files.

Exploratory data analysis and Pre-processing for Twitter Sentiment Analysis

Related Work

- Sentiment analysis has been studied for a long time in the research community.
- While the simplest version of the problem is to classify a document as positive or negative, a more difficult version is to assign a rating, say on a scale of 1-10. Further, in some settings, finding the source and target of the sentiment is useful. In some other settings, fine grained moods like happy, peaceful, calm, etc. are more useful to discover from the documents.
- Typically researchers have used classification based approaches for sentiment analysis. There has been quite some work on extracting interesting feature values for the task, including simple frequency counts to TFIDF to scaled likelihood.
- There have also been efforts to build sentiment lexicons. Quite a few manually generated sentiment lexicons are publicly available. SentiWordNet is an automatically generated lexicon which is quite popular as well.
- There have also been semi-supervised approaches proposed for the sentiment analysis task to increase sentiment lexicon size. These include using simple heuristics like (a) words separated by “and” have same sentiment polarity while words separated by “but” have different sentiment polarity. (b) dictionaries and thesaurus can be used to include synonyms of positive words in positive set and antonyms of positive words in negative set, and (c) words frequently co-occurring with extremely positive words are positive, while words frequently cooccurring with extremely negative words are negative.
- Recently there has also been work on extracting sentiment lexicons per domain, extracting sentiment carrying phrases, and also performing aspect based sentiment analysis.
- [Pang and Lee, 2008] and [Liu 2012] are good books on research done in the field of sentiment analysis. Further, we redirect the reader interested in reading more about previous work on sentiment analysis for Twitter data to [Alexander and Paroubek, 2010], [Kouloumpis et al., 2011], [Agarwal et al., 2011], and [Jørgen and Reitan, 2011].

Dataset Details

- Some people build a program to collect automatically a corpus of tweets based on two classes, “positive” and “negative”, by querying Twitter with two type of emoticons:
  - Happy emoticons, such as “:)”, “:P”, “:)” etc.
  - Sad emoticons, such as “:(“, “:’(”, “=(“.
- Others make their own dataset of tweets my collecting and annotating them manually which very long and fastidious.
- Our dataset consists of 1600000 tweets in English coming from two sources: Kaggle and Sentiment140.

	sentiment	id	date	user	text
0	4	1880323927	Fri May 22 00:50:28 PDT 2009	IHCF	i love you
1	4	2059478439	Sat Jun 06 17:00:58 PDT 2009	cryst0clearre	yaY! my phone is back on!
2	4	1693379887	Sun May 03 22:22:17 PDT 2009	b2therooke	@espisc thanks!
3	4	1835593196	Mon May 18 06:20:19 PDT 2009	AmandaGierusz	I watched the best workout show this morning i...
4	0	2228331816	Thu Jun 18 14:18:26 PDT 2009	!shoemark_smith	Last day of college today and I already miss ...

Challenges with Twitter data

- The presence of acronyms "bf" or more complicated "APL". Does it means apple ? Apple (the company) ?
- The presence of sequences of repeated characterssuch as "Juuuuuuuuuuuuuuuussssst", "hmmmm". In general when we repeat several characters in a word, it is to emphasize it, to increase its impact.
- The presence of emoticons, ":O", "T_T", ":¬|" and much more, give insights about user's moods.
- Spelling mistakes and “urban grammar” like "im gunna" or "mi".
- The presence of nouns such as "TV", "New Moon".
- Furthermore, we can also add,
- People also indicate their moods, emotions, states, between two such as, *\cries*, *hummin*, *sigh*.
- The negation, “can't”, “cannot”, “don't”, “haven't” that we need to handle like: “I don’t like chocolate”, “like” in this case is negative.

Exploratory data analysis

```
'   Cannot get chatroom feature to work.   Updated Java to 10, checked ports, etc. I can see video, but in the &quot;chat,&quo
t; only a spinning circle.   '
```

Data contains HTML entities

	ItemID	Sentiment	SentimentSource	SentimentText
45	46	1	Sentiment140	@ginaaa <3 go to the show tonight
46	47	0	Sentiment140	@spiral_galaxy @ymptweet it really makes me sad when i look at muslims reality now
47	48	0	Sentiment140	- all time low shall be my motivation for the rest of the week.
48	49	0	Sentiment140	and the entertainment is over, someone complained properly.. @ruptureapture experimental you say? he should experiment with a me...
49	50	0	Sentiment140	another year of lakers .. that's neither magic nor fun ...
50	51	0	Sentiment140	baddest day eveer.
51	52	1	Sentiment140	bathroom is clean..... now on to more enjoyable tasks.....
52	53	1	Sentiment140	boom boom pow
53	54	0	Sentiment140	but i'm proud.
54	55	0	Sentiment140	congrats to helio though

Data contains @mentions

	ItemID	Sentiment	SentimentSource	SentimentText
50	51	0	Sentiment140	baddest day eveer.
51	52	1	Sentiment140	bathroom is clean..... now on to more enjoyable tasks.....
52	53	1	Sentiment140	boom boom pow
53	54	0	Sentiment140	but i'm proud.
54	55	0	Sentiment140	congrats to helio though
55	56	0	Sentiment140	David must be hospitalized for five days end of July (palatine tonsils). I will probably never see Katie in concert.
56	57	0	Sentiment140	friends are leaving me 'cause of this stupid love http://bit.ly/ZoxZC
57	58	1	Sentiment140	go give ur mom a hug right now. http://bit.ly/azFwv
58	59	1	Sentiment140	Going To See Harry Sunday Happiness
59	60	0	Sentiment140	Hand quilting it is then...

Data contains URLs

	ItemID	Sentiment	SentimentSource	SentimentText
1578592	1578608	1	Sentiment140	'Zu SpÃœt' by Die Ã„rzte. One of the best bands ever
1578593	1578609	1	Sentiment140	Zuma bitch tomorrow. Have a wonderful night everyone goodnight.
1578594	1578610	0	Sentiment140	zummie's couch tour was amazing....to bad i had to leave early
1578595	1578611	0	Sentiment140	ZuneHD looks great! OLED screen @720p, HDMI, only issue is that I have an iPhone and 2 iPods . MAKE IT A PHONE and ill buy it @micro...
1578596	1578612	1	Sentiment140	zup there ! learning a new magic trick
1578597	1578613	1	Sentiment140	zyklonic showers *evil*
1578598	1578614	1	Sentiment140	ZZ Top â€" I Thank You ...@hawaiibuzz .....Thanks for your music and for your ear(s) ...ALL !!!! Have a fab... â™« ||url||
1578599	1578615	0	Sentiment140	zzz time. Just wish my love could B nxt 2 me
1578600	1578616	1	Sentiment140	zzz twitter. good day today. got a lot accomplished. imstorm. got into it w yet another girl. dress shopping tmrw
1578601	1578617	1	Sentiment140	zzz's time, goodnight. ||url||

Data contains UTF8 encoded symbols

Pre-processing the dataset

- Cleaning up the HTML entities, and any other HTML tags
- Remove @mentions
- Remove URLs from tweets
- Convert short forms of negation words to their full forms.
  - "isn't":"is not", "aren't":"are not", "wasn't":"was not", "weren't":"were not", "haven't":"have not", "hasn't":"has not", "hadn't":"had not", "won't":"will not", "wouldn't":"would not", "don't":"do not", "doesn't":"does not", "didn't":"did not", "can't":"can not", "couldn't":"could not", "shouldn't":"should not", "mightn't":"might not", "mustn't":"must not”
- Decode UTF8 encoded symbols
- Replace non-alphabetic characters to spaces.
- Ignore words of size 1.
- Remove punctuations
- Perform word lemmatization

TF-IDF vectorization, train/test split, cross validation

- We start by splitting the data randomly into two parts: train and test. We use 80% data for training and the remaining for testing.
- We extract TFIDF features from tweets using TFIDF vectorizer. It converts a collection of raw documents to a matrix of TF-IDF features.
- Further, we use these features to learn multiple classifiers. For each type of classifier, we first train the classifier on train data and report the accuracy on test data. We also report cross validation accuracy with 10 folds.
- We report confusion matrix and accuracy values for each classifier.

Twitter Sentiment Analysis Results

Results

Classifier	Test Acc.	Cross Validation Acc.
Naïve Bayes	0.7623	0.7524
Logistic Regression	0.7776	0.7757
Linear SVM	0.7867	0.7842
Decision Trees	0.5663	0.5758
Boosted Trees	0.7284	0.7161
Random Forests	0.7637	0.7609
Nearest Neighbors	0.5347	0.5163

- We obtain the maximum accuracy using Linear SVM. However, also notice that logistic regression and random forests also lead to similar accuracy values.
- However, simple classifiers like decision trees and nearest neighbors somehow cannot provide good results at all.
- We used default hyper parameter settings for all the classifiers as provided by scikit learn. Tuning those hyper parameters can lead to further accuracy boost and we plan to explore that as part of future work.

Results: Confusion Matrices

Naive Bayes:

```
array([[4441,  621],
       [1756, 3182]])
```

Logistic Regression:

```
array([[4077,  985],
       [1239, 3699]])
```

Linear SVM

```
array([[4100,  962],
       [1171, 3767]])
```

Decision Trees:

```
array([[1194, 3868],
       [ 469, 4469]])
```

Boosted Trees:

```
array([[3403, 1659],
       [1057, 3881]])
```

Random Forests:

```
array([[3814, 1248],
       [1115, 3823]])
```

Code and data

- The code is contained in the Jupyter notebook: TwitterSentimentAnalysis.ipynb
  - To be able to run this, you must have Jupyter notebook installed on your machine.
  - It can be run on any platform: Windows, Linux, Mac-OS
- The data is supplied as a csv file: TwitterSentimentAnalysis.csv
- Installations needed
  - Anaconda prompt, Jupyter notebook, numpy, pandas, sklearn, NLTK

Model Deployment

- Hardware requirement: Any operating system that supports Python. Less than 2 GB RAM.
  - Training the model takes a few hours on a machine with 8 processors.
- Software requirement: Python
- All the generated models can be saved using Python pickle library and can be easily loaded on machines with small RAM.

Concluding the Twitter Sentiment Analysis Project

Conclusion

- In this project, we discussed the problem of classifying whether a tweet is positive or negative.
- We performed classification using TF-IDF features and experimented with multiple classifiers.
- We used a Python library, Scikit Learn for building classifiers.
- The results show that Linear SVMs work well on our dataset.
- The solution can be useful for multiple stake-holders.

Future Work

- We could further improve our classifier
  - By trying to extract more features from the tweets,
  - By trying different kinds of features,
  - By tuning the parameters of the classifiers, or
    - We could try logistic regression without any regularization or try say L1 regularization.
    - For SVMs, we could try SVMs with various kinds of kernels or also vary the complexity parameter C and see the impact.
    - For decision trees, we could try different levels of pruning.
    - For random forests, we experimented with 100 trees, but we could vary the number of trees and check its impact on accuracy.
    - For nearest neighbors, we used k=5. We could try varying the value of k.
  - By trying more classifiers like deep learning architectures.

References

- Beevolve. (2012, October 10). An exhaustive study of Twitter users across the world. Retrieved December 18, 2014, from http://www.beevolve.com/twitter-statistics/#c1
- Councill, I. G., McDonald, R. & Velikovich, L. (2010). What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In Proceedings of the workshop on negation and speculation in natural language processing (pp. 51–59). NeSp-NLP '10. Uppsala, Sweden: Association for Computational Linguistics.
- Kiritchenko, S., Zhu, X. & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. Journal of Artificial Intelligence Research, 50, 723–762.
- Cortes, C. & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20 (3), 273–297.
- Fletcher, T. (2009). Support vector machines explained, tutorial paper. http://www.tristanfletcher.co.uk/SVM%20Explained.pdf. Retrieved June 11, 2015.
- Hsu, C.-W. & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks, 13 (2), 415–425.
- Mitchell, T. M. (1997). Machine learning (1st ed.). New York, NY, USA: McGraw-Hill, Inc.
- Vryniotis, V. (2013, November 20). Machine learning tutorial: the max entropy text classifier. Retrieved Dece r 16, 2014, from http: //blog.datumbox.com/machine-learning-tutorial-the-maxentropy- text-classifier/
- Givón, T. (1993). English grammar: a function-based introduction. John Benjamins Publishing.
- Morante, R. & Sporleder, C. (2012). Modality and negation: an introduction to the special issue. Computational Linguistics, 38 (2), 223–260.
- Tottie, G. (1991). Negation in English speech and writing: a study in variation. Academic Press.
- Polanyi, L. & Zaenen, A. (2006). Contextual valence shifters (J. G. Shanahan, Y. Qu & J. Wiebe, Eds.). Springer.
- Manning, C. D., Raghavan, P. & Schütze, H. (2008). Introduction to information retrieval. Cambridge, UK: Cambridge University Press.
- Marcus, M. P., Marcinkiewicz, M. A. & Santorini, B. (1993). Building a large annotated corpus of English: the Penn Treebank. Computational Linguistics, 19 (2), 313–330.

References

- Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J. & Smith, N. A. (2011). Part-of-speech tagging for Twitter: annotation, features, and experiments. In Proceedings of the 49th annual meeting of the Association for Computational Linguistics: human language technologies: short papers (Vol. 2, pp. 42–47). Association for Computational Linguistics. Portland, Oregon.
- Collins, M. (2003). Head-driven statistical models for natural language parsing. Computational Linguistics, 29 (4), 589–637.
- Hudson, R. (2010). An encyclopedia of word grammar and English grammar. Retrieved December 16, 2014, from http://tinyurl.com/wgencyc
- McKinney, W. (2012). Pandas: a Python data analysis library. Retrieved May 13, 2015, from http://pandas.pydata.org
- Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C. & Smith, N. (2014). A dependency parser for tweets. In Proceedings of the conference on empirical methods in natural language processing (pp. 1001– 1012). EMNLP '14. Doha, Qatar.
- Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N. & Smith, N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In Proceedings of the 2013 conference of the North American Chapter of the Association for Computational Linguistics: human language technologies (pp. 380–390). Atlanta, GA, USA.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). Scikit-learn: machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
- Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." *Foundations and Trends® in Information Retrieval* 2, no. 1–2 (2008): 1-135.
- Liu, Bing. "Sentiment analysis and opinion mining." *Synthesis lectures on human language technologies* 5, no. 1 (2012): 1-167.
- Pak, Alexander, and Patrick Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining." In *LREc*, vol. 10, no. 2010, pp. 1320-1326. 2010.
- Kouloumpis, Efthymios, Theresa Wilson, and Johanna D. Moore. "Twitter sentiment analysis: The good the bad and the omg!." *Icwsm* 11, no. 538-541 (2011): 164.
- Agarwal, Apoorv, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. "Sentiment analysis of twitter data." In *Proceedings of the workshop on languages in social media*, pp. 30-38. Association for Computational Linguistics, 2011.
- Faret, Jørgen, and Johan Reitan. "Twitter sentiment analysis." (2011).

Thanks!