Analyzing the Effectiveness of Three Scikit-Learn Text Classifiers for Sentiment Analysis

Alex Pawelczyk
alex.pawelczyk@uwaterloo.ca
University of Waterloo
Waterloo, Ontario

ABSTRACT

Sentiment analysis is an important task of natural language processing (NLP) that aims to assign quantitative sentiment values to documents of text. This research uses the Python scikit-learn library to implement and analyze the effectiveness of Multinomial Naive Bayes (MNB), Support Vector Machines (SVM), and Logistic Regression (LR) text classifiers. The goal is to determine which of the three models is most effective at classifying text sentiment (positive or negative). Two experiments are conducted to test each model using two real-world datasets. Precision, recall, and F-measure is calculated for each model, and the results show that SVM had the highest effectiveness scores for both datasets. Moreover, the MNB, SVM, and LR models achieve higher scores when classifying sentiment in movie reviews, as opposed to tweets.

KEYWORDS

Text classification, Sentiment analysis, Multinomial Naive Bayes, Support Vector Machines, Logistic Regression

ACM Reference Format:
Alex Pawelczyk. 2020. Analyzing the Effectiveness of Three Scikit-Learn Text Classifiers for Sentiment Analysis. In *CS848 Final Project*. ACM, New York, NY, USA, 7 pages.

1 INTRODUCTION

Text classification is an extensively studied task in the field of natural language processing (NLP), where the aim is to assign a set of predefined categories to different documents of text. The world generates 2.5 quintillian bytes of text data per day, and with the development of new devices, sensors, and technologies, this rate is expected to accelerate even more [11]. Moreover, 80% of enterprise data is unstructured (i.e., comes from sources such as emails, social media posts, and blogs), and gathering insights from large, unstructured datasets is a challenging task [10]. Fortunately, NLP can be used to automatically analyze large volumes of text from various sources, including news outlets, blogs, and social media platforms.

Sentiment analysis aims to assign a quantitative value to a piece of text expressing an affect or mood [20]. Many businesses are customer-centric and base their decisions on customer feedback and opinions; thus, monitoring customer sentiments can provide value to a business and poses a wide range of use cases. For example, much of the content found on Twitter is opinion oriented, and sentiment analysis can be used to detect hate speech in tweets [1, 24, 26]. User moods can also be extracted from tweets and leveraged to predict stock prices [2, 19, 22]. Many online platforms provide an option for users to rate products and write reviews based on their past experiences of using them. Thus, sentiment analysis can be leveraged to help businesses gain valuable insight into the performance of their products and identify areas for improvement.

One challenge in sentiment analysis is choosing the most effective model for analyzing a given dataset. Effectiveness describes how well a model is doing the job it was designed for, where no consideration is given to the resource consumption of a model while doing that job [5]. Examples of popular effectiveness metrics include precision, recall, and F-measure. The effectiveness of a model could be dependant on the domain of a dataset, and different classifiers may generate different results. Many machine learning (ML) libraries also provide developers with powerful tools for building different classifiers, and it is important to conduct experiments that analyze the effectiveness of ML model implementations from various sources.

Using the Python scikit-learn library [21], this work analyzes the effectiveness of the Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), and Logistic Regression (LR) classifiers for conducting sentiment analysis. The aim of this work is to investigate whether there are differences in effectiveness among the MNB, SVM, and LR models, and to determine if the effectiveness of each algorithm remains consistent over two different dataset domains. The first dataset contains tweets, the second contains IMDB movie reviews, and every tweet or movie review is labeled with either a positive or negative sentiment. For each dataset, a MNB, SVM, and LR model is trained on 80 percent of the

CS848 Final Project, April 22, 2020, Waterloo, ON

data and tested on the remaining 20 percent. Precision, recall, and F1 are calculated to show the effectiveness of the models, the results are analyzed, and different ideas are presented on how to improve these results.

Section 2 reviews some of the related literature and relevant use cases of sentiment analysis. Section 3 outlines the research questions and hypotheses that guide this work. Section 4 provides a general overview of the MNB, SVM, and LR classification algorithms, while Section 5 details the datasets that are used in the experiments. Section 6 describes the experimental procedure that was used to test the hypotheses of this work, along with a discussion of the results. Section 7 concludes the paper by identifying future research directions.

2 RELATED WORK

Since this study is centered on the domains of Twitter and IMDB, a specific focus is given to previous studies that use sentiment analysis for classifying tweets and product reviews.

Predicting Stock Prices via Twitter Sentiment

Many people use Twitter to express their opinions about certain products or brands, and this information can be leveraged to predict stock prices. Bollen et. al [3] investigate whether measures of public mood states derived from large-scale Twitter feeds are correlated, or even predictive, of Dow Jones Industrial Average (DJIA) values. OpinionFinder was used to measure positive and negative moods from text content, and Google-Profile Mood of States (GPOMS) was used to measure mood in terms of 6 dimensions (Calm, Alert, Sure, Vital, Kind, and Happy). The resulting public mood time series were correlated to the DJIA to assess their ability to predict changes in the DJIA over time. The authors then used a combination of past DJIA values and the public mood time series to train a Self-Organizing Fuzzy Neural Network (SOFNN) for predicting DJIA closing values. Results show that the SOFNN achieved an accuracy of 87.6 percent, outperforming traditional methods. Moreover, the authors determined that the calmness and happiness of the public (measured by GPOMS) are predictive of of the DJIA, rather than the general levels of positive or negative sentiment measured by OpinionFinder.

Building on the strategy of Bollen et. al, Mittal and Goel [16] evaluate the effectiveness of the linear regression, logistic regression, SVM, and SOFNN algorithms for predicting stock prices. However, rather than using k-fold cross validation to measure the accuracy of these models, Mittal and Goel propose *k-fold sequential cross validation*, a validation technique that trains on all days up to a specific day and tests for the next k days. Experimental results suggest that when trained on the feature set of the DJIA values, a collection of Calm mood values, and the Happiness dimension over the past three days, SOFNN performs very well in predicting DJIA values and achieves an accuracy of 75.56 percent. Although this accuracy is lower than the one achieved by Bollen et. al, the authors claim that using k-fold sequential cross validation gives stronger evidence that the correlation spans over the entire range of data.

Detecting Hate Speech on Twitter

The Twitter Hateful Conduct policy aims to prevent hate speech from being used on their platform, and this poses an opportunity to use sentiment analysis techniques for the automatic detection of hate speech. Kwok and Wang [13] focused on the black constituency of Twitter and implemented a NB classifier for classifying tweets as either racist or non-racist. The classifier was evaluated using 10-fold cross validation, achieving an accuracy of 76% and a mean error rate of 24%. The authors claim that this performance is insufficient, and the low accuracy stems from only employing unigrams, which do not consider information such as text sentiments. Therefore, future implementations of their algorithm plan to include a combination of sentiment analysis and classification, bigrams, and word-sense disambiguation.

Burnap and Williams [4] aimed to develop a hate-speech classifier that could support policy makers during an evidence-based decision-making process. They first identified the nuanced features of hate speech that can be used for classification, which are based on typed dependencies that provide a representation of syntactic grammatical relationships in a tweet. The authors then used the Java WEKA ML library [9] to implement Bayesian Logistic Regression (BLR), Random Forest Decision Tree (RFDT), SVM, and *ensemble* classifiers. The ensemble method uses a voted meta-classifier based on maximum probability that combines the outputs of BLR, RFDT, and SVM to make a final classification. Their results show that BLR, RFDT, and SVM performed similarly across most feature sets. However, the ensemble classifier matched or improved on the recall of these models in all experiments, suggesting that an ensemble classification approach is must suitable for classifying hate speech.

Davidson et. al [7] address the challenge of mistakenly classifying instances of offensive language as hate speech and build a multi-class classifier to distinguish between these categories. The model was trained using a dataset of tweets containing hate speech keywords, which had been previously labeled as either being hate speech, offensive language, or neither. Their best model achieved an overall precision of 0.91, recall of 0.90, and F-measure of 0.90, but 40% of hate speech had been misclassified. This suggests that compared to human analysts, their model has a stricter set of rules for classifying a tweet as hateful.

Sentiment Analysis for Online Product Reviews

Yu et. al [29] approached the problem of predicting sales performance using online reviews and identify important factors involved in generating predictions. Their primary contributions are the ARSA and ARSQA models, which factor in the effect of sentiments and past sales performance on future sales performance. In the case of ARSQA, the quality of reviews is also considered. Experimental results show that ARSQA generally outperforms ARSA, and both sentiments and review quality have significant impact on the future sales performance of products.

Panichella et. al [20] investigate whether the structure, sentiment, and text features of app reviews can be used to classify and select useful reviews that help developers identify bugs or suggestions for improving their apps. They propose a taxonomy to classify app reviews into categories that are relevant to software maintenance and evolution. The authors argue that NLP, topic analysis, and sentiment analysis each have separate advantages, and they experimented with various combinations of the three techniques to see what combination produced the best results. For each combination, five different models (NB, SVM, Logistic Regression, J48, and ADTree) were trained and tested on an Apple App Store dataset containing reviews for three different apps. They found that a using J48 with a combination of NLP, topic analysis, and sentiment analysis techniques achieves better results (precision of 75 percent and recall of 74 percent) than the results obtained from using each technique separately (precision of 70 percent and recall of 67 percent).

3 RESEARCH QUESTIONS AND HYPOTHESES

The main goal of this research is to use the scikit-learn library to implement and investigate the effectiveness of the MNB, SVM, and LR sentiment classifiers. Thus, the research questions that guide this work are:

- **RQ1:** For each dataset, will MNB, SVM, or LR achieve the highest overall F-score?
- **RQ2:** Will MNB, SVM, and LR maintain consistent effectiveness across both domains (Twitter and IMDB)?

Since [8, 23] show that SVM can be effective at text classification, the first hypothesis for this evaluation is that SVM will achieve the highest F-score for both datasets. The second hypothesis is that all models will maintain consistent effectiveness across the Twitter and IMDB domains. These hyptoheses are tested by conducting separate experiments for each dataset and the effectiveness of each model is measured by using the widely adopted performance metrics of precision, recall, and F-measure. Precision measures the the positive patterns that are correctly predicted from the total predicted patterns in a positive class, recall measures the fraction of correctly classified positive patterns, and F-measure represents the harmonic mean between precision and recall values [25].

4 MODELS

Multinomial Naive Bayes (MNB)

MNB is a widely used ML model due to its efficiency and ability to combine evidence from a large number of features [15]. MNB is based on Bayes' Theorem and has a bag of words assumption that the position of words does not matter. MNB also assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature (i.e., assumption of conditional independence) [12]. The distribution of the scikit-learn MNB model is parameterized by vectors $\theta_y = \theta_{y1}, ..., \theta_{yn}$, where θ_y represents a probability distribution over the vocabulary and how likely a feature is to appear. The parameter θ_y is estimated by the following smoothed version of maximum likelihood:

$$\theta'_{yi} = \frac{N_{yi} + \alpha}{N_{yi} + \alpha n} \tag{1}$$

In Equation 1, N_{yi} represents the number of times that feature i appears in sample class y of the training dataset, and $N_y = \sum_{i=1}^{n} N_{yi}$ represents the total count of all features for class y [21]. The MNB models implemented in this work use the scikit-learn default smoothing parameter of $\alpha = 1$, which initiates Laplace smoothing to account for features not present in the learning samples (i.e., zero probability values).

Support Vector Machines (SVM)

Support Vector Machines (SVM) are supervised learning models that aim to find the maximum marginal hyperplane that optimally divides a dataset into classes [27]. This research builds sentiment classifiers via the C-Support Vector Classification model of the scikit-learn libray. These classifiers use a linear kernel, the inverse of regularization strength C is set to 1, and *gamma* is the fraction of one over the total number of features. Given one array of training samples and one array of sentiment labels, an SVM model can be fit on the training data and used to predict positive or negative sentiments.

SVM is a memory efficient algorithm because of its use of support vectors, which are a subset of training data points in the decision function [21]. However, separating support vectors from the rest of the training data is a quadratic programming (QP) problem and has expensive time complexity. Scikit-learn uses a QP solver that is based on a libsvm [6] implementation in which depending how the libsvm cache is used in practice, the algorithmic complexity scales between $O(N_{features} \times N^2_{samples})$ and $O(N_{features} \times N^3_{samples})$.

Logistic Regression (LR)

LR is a classification algorithm that uses the sigmoid activation function to create a probability distribution over the classes. A hypothesis can be represented via the function $h_\theta(x) = 1/1+e^{-\theta^T x}$ such that $0 \leq h_\theta(x) \leq 1$ and x is a feature vector [17]. In the context of sentiment analysis for tweets and movie reviews, given a document and its feature vector x that is parameterized by θ, $h_\theta(x)$ outputs the estimated probability that the sentiment of the document is positive. LR also employs a convex cost function that severely penalizes the learning algorithm whenever an incorrect classification is made [18]. The scikit-learn LR models implemented in this work use L2 regularization and set the inverse of regularization strength C equal to 1.

5 DATA

Two datasets were gathered for the purpose of this research. One consists of 50,000 IMDB movie reviews and their corresponding sentiments. Exactly half of the reviews are labeled with a positive sentiment, and the remaining half have a negative sentiment. Mass et. al [14] created this dataset by extracting movie reviews from the IMDB movie review website. Each movie review is paired with a numeric rating (scale of 1-10), and the authors used these ratings to classify the reviews as either positive or negative. The authors considered only highly polarized reviews, where a negative review has a score in the range of [1, 4], and a positive review has a score in the range of [7, 10]. Since certain movies receive substantially more reviews than others, the authors maintained balance in the dataset by including a maximum of 30 reviews per movie.

The second dataset used in this work is a Twitter dataset consisting of 1.6M tweets and a positive or negative sentiment for each tweet. Go et. el [8] used an automatic approach to generate the dataset by leveraging the presence of emoticons in tweets. The authors assumed that any tweets containing positive emoticons, like :), were positive, and tweets with negative emoticons, like :(, were negative. Moreover, any tweets containing both positive and negative emoticons were considered to be neutral and excluded from the dataset. All tweets were collected via the Twitter Search API, and each row of the source dataset consists of a tweet, positive or negative sentiment, and other metadata related to the tweet (e.g., time stamps). However, only the tweets and sentiments were extracted for this research, and the dataset was reduced to 50,000 tweets (half positive and half negative) to match the size and class balance of the IMDB dataset.

These datasets are particularly useful for investigating **RQ2** because the domain of online movie reviews varies greatly from Twitter. At the time of collecting the Twitter dataset, tweets were limited to a maximum of 140 characters, while movie reviews have a maximum limit of 10,000 characters. Moreover, tweets have the potential to refer to many different subject domains, while movie reviews are focused on only one (movies). Tweets also contain a higher frequency of mispellings and slang than text from other domains [8].

6 EVALUATION

For each dataset, an experiment was conducted to evaluate the performances of MNB, SVM, and LR. This section describes the procedure that is used by both experiments to determine the most effective models. Both experiments are conducted using Python 3, the scikit-learn library, and the Jupyter Notebook.

Experimental Procedure

Each experiment begins by loading the appropriate dataset into Jupyter Notebook, and only the text (tweets or movie reviews) and sentiments (positive or negative) are extracted. Data processing begins by removing any blank rows and converting all text to lowercase. Next, all text streams are tokenized, where each entry in the corpus is split into a set of individual words. Stop words are not removed from the text because certain stop words (e.g., negating words) are indicative of sentiment [14].

After processing the data, the dataset is split into training and testing sets, where 80 percent of the data is used for training and 20 percent is used for testing. Next, the training and testing predictors are each converted into a matrix of term frequency-inverse document frequency (TF-IDF) features via the scikit-learn TF-IDF Vectorizer [21]. All three models are then trained using the TF-IDF feature matrix from the training split. Once the models are finished training, predictions are made using the TF-IDF feature matrix from the testing split. Finally, all necessary performance metrics are calculated (e.g. precision, recall, and F1) and a confusion matrix is generated.

Experimental Results and Discussion

Table 1 summarizes the results of both experiments, where SVM achieves the highest effectiveness for both datasets (F-measure of 0.91 and 0.78 under the IMDB and Twitter domains, respectively). These results answer **RQ1** and support the hypothesis that SVM would achieve the highest F-measure for both datasets. However, LR achieves scores within 2% of SVM for both datasets, and the difference in effectiveness between SVM and LR may not be significant. Table 1 also shows that the SVM, MNB, and LR models are most effective at classifying movie reviews. Although the models maintain a consistent order in scores across both datasets (i.e., SVM has the highest scores, followed by LR and MNB), the actual scores themselves are not consistent

and contradict the hypothesis that consistent effectiveness scores would be maintained across all domains.

A likely reason why the models are more effective at classifying movie reviews is that the IMDB dataset is domain specific to movies, while tweets span across a much wider range of domains. The models trained on the IMDB movie dataset are only exposed to documents of a similar nature that have the goal of rating a movie. Conversely, tweets can be related to any topic that a user is interested in, and more training examples are needed to cover the diverse nature of tweets. Moreover, the 140 character limit limit of tweets may be another reason why all the models had higher effectiveness scores when classifying movie reviews. IMDB states that a quality review is typically between 200-500 characters long, but users are allowed to submit reviews of up to 10,000 characters. Even though the Twitter and IMDB datasets consisted of an equal number of entries, it is likely that the IMDB models achieved higher scores due to the higher number of tf-idf vector features obtained from the larger supply of words in the IMDB training set.

Both experiments also highlight the poor time complexity of SVM, in which the average training time for both models was 16.93 minutes. The average training time for the LR models was 2.8 seconds, and the MNB models took an average of only 0.03 seconds to train. These results suggest that a tradeoff exists between effectiveness and efficiency, and the choice of a model should be task-dependant. For example, real-time Twitter sentiment analysis for predicting stock prices may benefit from the fast training times of LR and MNB. However, in an environment where human lives are at risk (i.e., autonomous vehicles), a business may need to endure longer training times to achieve maximum effectiveness.

Table 1: Results from both experiments showing the precision, recall, and F-measure achieved by the SVM, MNB, and LR models when trained on Twitter and IMDB datasets. Scores for both positive and negative sentiments are presented, along with a weighted average of the positive and negative scores.

Dataset	Model	Class	Prec.	Rec.	F-measure
IMDB	SVM	Neg.	0.92	0.90	0.91
		Pos.	0.90	0.92	0.91
		Avg.	0.91	0.91	0.91
IMDB	MNB	Neg.	0.84	0.88	0.86
		Pos.	0.88	0.83	0.86
		Avg.	0.86	0.86	0.86
IMDB	LR	Neg.	0.90	0.88	0.89
		Pos.	0.89	0.91	0.90
		Avg.	0.89	0.89	0.89
Twitter	SVM	Neg.	0.79	0.76	0.78
		Pos.	0.76	0.79	0.78
		Avg.	0.78	0.78	0.78
Twitter	MNB	Neg.	0.72	0.82	0.77
		Pos.	0.79	0.68	0.73
		Avg.	0.76	0.75	0.75
Twitter	LR	Neg.	0.79	0.75	0.77
		Pos.	0.76	0.79	0.78
		Avg.	0.77	0.77	0.77

7 FUTURE WORK

The results of the experiments suggest that SVM, MNB, and LR can be effective at classifying the sentiments of tweets and movie reviews, but these results can potentially be improved. First, a new model will be trained and tested using the entire 1.6M records from the Twitter source dataset. Compared to the reduced dataset used in these experiments, a larger training set of 1.28M tweets is likely to improve the effectiveness of all models. Go et. al [8] also found that for some models, accuracy was improved when using a combination of unigrams and bigrams. Future experiments will build on these ideas and test SVM, MNB, and LR models implemented with the same feature combinations. These experiments will also investigate how adding a sentiment class for neutral text will impact the performance of different models.

The primary limitation of this work is a lack of statistical analysis for measuring significance in the evaluation results. Precision, recall, and F-measure are useful metrics for evaluating the effectiveness of models, but these metrics alone do not suffice for determining if the results are reliable. Yeh [28] found that many commonly used tests for measuring model effectiveness underestimate the significance between different models. This underestimation is often due to a violation of independence assumptions, and a critical component of future work will be to use randomized tests for measuring precision and F-measure. Randomization is a type of stratified shuffling that gathers all scores made by the model of interest, shuffles them, and reassigns them to one of n models that are being evaluated [28]. Randomization requires no previous assumptions of independence, and using randomization in future experiments will help determine if one model achieves significantly better results than another.

Since this work only uses the scikit-learn implementations of MNB, SVM, and LR, future research will recreate the Twitter and IMDB sentiment analysis experiments using other ML libraries. The primary goal of these experiments is to investigate if consistent results can be achieved by the same algorithms that are implemented with different frameworks (e.g., scikit-learn, WEKA [9] and Apache Spark [30]). These

experiments are important because they can help developers make informed decisions when evaluating the overall usefulness of different models implemented by different sources. However, Büttcher et. al [5] argue that efficiency is an equally important metric to consider when evaluating how well a model is doing the task it was designed for. Efficiency measures the resource consumption of a model while conducting a task, and these experiments will analyze the training and testing time complexities of the MNB, SVM, and LR models.

Hate speech detection on Twitter is an interesting use case of sentiment analysis, and a long-term goal is to use ideas from [4, 7, 13] to build a hate-speech classifier. Models will be trained to classify tweets as either being hate speech, offensive, or neither, and a key challenge will be to reduce the number of times that offensive speech gets classified as hate speech. Training a model to define a concrete border between offensive and hateful tweets could perhaps be accomplished by implementing a penalty system that would severely punish the learning algorithm whenever an offensive tweet is misclassified as hate speech, but the incorrect classification of hateful tweets would be penalized less. The intuition behind this idea is that if Twitter wanted to one day launch an automatic hate speech detection and removal system, classifying non-hateful tweets as hateful would be very problematic and inhibit the usability of the platform (e.g., people could become frustrated and leave the platform). Conversely, incorrectly classifying a hateful tweet is not the ideal outcome of a model, but the consequences are much less severe (e.g., a small number of people see an offensive tweet before it is manually taken down by Twitter).

8 CONCLUSION

This research shows that scikit-learn implementations of MNB, SVM, and LR can achieve high effectiveness when classifying sentiment in text. SVM achieves the highest effectiveness scores for both Twitter and IMDB domains, but the quadratic time complexity of this algorithm results in much higher training times than MNB or LR. Experimental results also show that these algorithms are more effective when trained on domain-specific IMDB movie reviews, rather than multi-domain tweets. When combined with other ML techniques, sentiment analysis can play a valuable role in helping businesses gain intelligence from massive amounts of data. As data continues to grow at a massive scale, these techniques are likely to become increasingly valuable, and it is important for the NLP community to continue research towards improving the effectiveness and efficiency of sentiment classifiers.

REFERENCES

[1] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekanata. 2017. Hate speech detection in the Indonesian language: A dataset and preliminary study. In *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. 233–238.

[2] L. Bing, K. C. C. Chan, and C. Ou. 2014. Public Sentiment Analysis in Twitter Data for Prediction of a Company's Stock Price Movements. In *2014 IEEE 11th International Conference on e-Business Engineering*. 232–239.

[3] Johan Bollen, Huina Mao, and Xiao-Jun Zeng. 2010. Twitter mood predicts the stock market. *CoRR* abs/1010.3003 (2010). arXiv:1010.3003 http://arxiv.org/abs/1010.3003

[4] Pete Burnap and Matthew Williams. 2015. Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making: Machine Classification of Cyber Hate Speech. *Policy Internet* 7 (04 2015). https://doi.org/10.1002/poi3.85

[5] Stefan Büttcher, Charles Clarke, and Gordon V. Cormack. 2010. *Information Retrieval: Implementing and Evaluating Search Engines*. The MIT Press.

[6] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2 (2011), 27:1–27:27. Issue 3. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[7] Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. *CoRR* abs/1703.04009 (2017). arXiv:1703.04009 http://arxiv.org/abs/1703.04009

[8] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *Processing* 150 (01 2009).

[9] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* 11, 1 (Nov. 2009), 10–18. https://doi.org/10.1145/1656274.1656278

[10] IBM 2016. *Big Data: Big Challenge or Big Opportunity*. IBM. https://www.ibm.com/watson/infographic/discovery/big-data-challenge-opportunity/.

[11] IBM 2017. *10 Key Marketing Trends for 2017 and Ideas for Exceeding Customer Expectations*. IBM. ://paulwriter.com/wp-content/uploads/2017/10/10-Key-Marketing-Trends-for-2017.pdf.

[12] Dan Jurafsky. 2000. *Text Classification and Naive Bayes*. Stanford University. https://web.stanford.edu/~jurafsky/NLPCourseraSlides.html.

[13] Irene Kwok and Yuzhou Wang. 2013. Locate the Hate: Detecting Tweets against Blacks. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence* (Bellevue, Washington) *(AAAI'13)*. AAAI Press, 1621–1622.

[14] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1* (Portland, Oregon) *(HLT '11)*. Association for Computational Linguistics, USA, 142–150.

[15] Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

[16] Anshul Mittal and Arpit Goel. 2012. *Stock Prediction using Twitter Sentiment Analysis*. Stanford University. http://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf.

[17] Andrew Ng. 2016. *Lecture 6.2 — Logistic Regression | Hypothesis Representation*. https://www.youtube.com/watch?v=t1IT5hZfS48.

[18] Andrew Ng. 2016. *Lecture 6.4 — Logistic Regression | Cost Function*. https://www.youtube.com/watch?v=HIQlmHxI6-0.

[19] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi. 2016. Sentiment analysis of Twitter data for predicting stock market movements. In

2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES). 1345–1350.

[20] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall. 2015. How can i improve my app? Classifying user reviews for software maintenance and evolution. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 281–290.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[22] N. Rajesh and L. Gandy. 2016. CashTagNN: Using sentiment of tweets with CashTags to predict stock market prices. In *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*. 1–4.

[23] D. Sculley, Wachman, and Gabriel M. 2007. Relaxed online SVMS for spam filtering. *Proceedings of the 30th Annual International ACM SIGIR Conference*. https://doi.org/10.1145/1277741.1277813

[24] N. A. Setyadi, M. Nasrun, and C. Setianingsih. 2018. Text Analysis For Hate Speech Detection Using Backpropagation Neural Network. In *2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*. 159–165.

[25] E VALUATIONS. 2015. A REVIEW ON EVALUATION METRICS FOR DATA CLASSIFICATION EVALUATIONS.

[26] H. Watanabe, M. Bouazizi, and T. Ohtsuki. 2018. Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection. *IEEE Access* 6 (2018), 13825–13835.

[27] Patrick Winston. 2010. *16. Learning: Support Vector Machines*. MIT OpenCourseWare. https://www.youtube.com/watch?v=_PwhiWxHK8o.

[28] Alexander Yeh. 2000. More Accurate Tests for the Statistical Significance of Result Differences. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 2* (Saarbrücken, Germany) *(COLING '00)*. Association for Computational Linguistics, USA, 947–953. https://doi.org/10.3115/992730.992783

[29] X. Yu, Y. Liu, X. Huang, and A. An. 2012. Mining Online Reviews for Predicting Sales Performance: A Case Study in the Movie Domain. *IEEE Transactions on Knowledge and Data Engineering* 24, 4 (2012), 720–734.

[30] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. 2016. Apache Spark: A Unified Engine for Big Data Processing. *Commun. ACM* 59, 11 (Oct. 2016), 56–65. https://doi.org/10.1145/2934664