IMDB Sentiment Classification

Jonathan Pearce, Saima Tarzeen, Cristian Groza

February 22, 2019

Abstract

Expressing public opinion online has become a more common practice in recent years with the increase in personal device usage. Whether it be sharing that opinion to your friends via social media or to the public domain in the form of a review, expressing emotion and opinions online has become a standard practice for many people. Sentiment analysis is the field that utilizes this information in order to gain insights into how different topics, products and people are viewed by the public. This area has been well studied in academia and continues to be a popular area of research in natural language processing. The goal of this project was to develop a supervised machine learning model that could accurately classify IMDB movie reviews as expressing either positive or negative sentiment. The models tested and evaluated were Bernoulli Naïve Bayes, Support Vector Machines, Logistic Regression and Decision Trees. The Support Vector Machines and Logistic Regression models performed very similarly, both achieving over 90% accuracy in our validation process. Our final chosen Logistic Regression model was able to achieve 90.45% accuracy on a subset of the test data.

Introduction

In recent years access to online platforms has become more readily available and it is easier now more than ever to both share and view opinions online. Both social media websites such as Facebook and Twitter as well as websites that feature product reviews such as Amazon, Yelp and IMDB have become a common place for people's opinions. People's reviews can aid in providing the general public a greater idea of what a product is like and when performed on a mass scale can provide a very accurate assessment of a product's quality. Usually these reviews, such as the movie reviews on IMDB are smaller pieces of text, sometimes referred to as Short Texts [2]. Performing accurate sentiment analysis on these short texts can be quite a challenging task, but it does provide us with vital information about the product being considered. Specifically for IMDB movie reviews, sentiment analysis can help gauge the public's opinion of movies and whether they are liked or disliked. The goal of this project was to develop a machine learning model that could accurately classify IMDB movie reviews as either positive or negative. The training dataset was comprised of 25,000 labelled movie reviews from IMDB evenly divided into positive and negative classes, and a test set of 25,000 unlabelled movie reviews also from IMDB. The machine learning models we experimented with for performing this classification were Bernoulli Naïve Bayes, Support Vector Machines, Logistic Regression and Decision Trees. Multiple features were tested with different models

including raw text features such as unigrams, bigrams and TF-IDF (Term Frequency $\times$ Inverse Document Frequency) as well as lexicon features such as positive-negative word counts [4]. A 5 fold cross validation procedure was used to validate and compare our models' performance. In general the Support Vector Machines and Logistic Regression models performed the best in terms of average classification accuracy across all 5 data folds. These two models were able to consistently achieve over 89% accuracy on multiple different feature sets. The top SVM model had a 90.5% accuracy and the best Logistic Regression model performed very similarly with a 90.1% accuracy. In the end we decided to select the Logistic Regression model to submit for evaluation on Kaggle as it is able to train on the data significantly faster than the SVM model. The amount of time taken by the SVM model to train and predict was so large in our case that tuning hyper-parameters was not feasible. The logistic regression model had nearly the same performance and its favourable runtime makes our Kaggle submission easier to reproduce and build off of for future work in this topic. The selected Logistic Regression model achieved 90.45% accuracy on a subset of the test data.

Previous Work

Opinion mining has developed into a widely popular concept among researchers in recent years. Ample investigation has been conducted in this area based on feature extraction, feature selection, deep learning algorithms. Kumar et al. [5] performed a very similar experiment on the IMDb movie review dataset, focusing on the effects of using hybrid features and four feature selection methods to predict positive-negative reviews. Machine learning based features (TF-IDF) were concatenated with Lexicon based features (Positive/Negative word count, Positive/Negative Connotations) to create "Hybrid Features". The dataset was trained and validated with SVM, Maximum Entropy, KNN and Naïve Bayes classification models for each feature selection method. It was seen that the overall prediction accuracy of the models improved significantly with the use of hybrid features compared to simple machine learning features. Sharma and Dey [8] explores the performance of feature selection methods in sentiment analysis of movie reviews in terms of accuracy, precision and recall. Three sentiment feature lexicons along with five feature selection methods (Information Gain, Gain Ratio, Document Frequency, Relief-F and Chi-Squared) are scrutinized. It is seen that the sentiment lexicons performed poorly and Gain Ratio gave the overall best performance among the selection methods. It was also seen that the classification results depended heavily upon the number of features selected for experimentation. Pouransari and Ghili [7] investigated a wide range of Natural Language Processing (NLP) techniques to perform sentiment classification. A binary classification was performed on the same IMDb movie review dataset used in this project using Random Forest, Logistic Regression and SVM classifiers. A multi-class classification was performed on a different dataset using low-rank Recursive Neural Tensor Networks (RNTN). This method was quite time efficient; it was also able to train 10 different models to be used for ensemble-averaging which significantly improved classification accuracy.

Dataset and Setup

The training dataset consists of a total 25,000 labelled IMDB reviews, partitioned evenly into negative and positive labels. The test set is comprised of an additional 25,000 unlabelled reviews. This test set is kept reserved by Kaggle and our uploaded algorithms are initially tested on 30% of the test set.

For the given data, text pre-processing was performed with aid of the Natural Language Toolkit (NLTK) [6] package. The review texts were tokenized with *nltk.word_tokenize*. The resulting word tokens were then lemmatized using *nltk.WordNetLemmatizer*. A *CountVectorizer* from Python's scikit-learn library was fit on the training portion of the data to produce binary occurrences and TF-IDF features for unigrams and bigrams of the text.

Proposed Approach

For this project, we decided to evaluate and compare the following classifiers; Bernoulli Naive Bayes, Support Vector Machines, Logistic Regression and Decision Trees. All model evaluations were performed using a 5-fold cross validation process on the training data. The Bernoulli Naive Bayes model was implemented from scratch and its performance acted as a baseline for our other model investigations. Wang and Manning's research demonstrates the power of basic Naive Bayes models with respect to sentiment classification of movie review segments [9]. It was shown that Naive Bayes is only superior to other basic algorithms for short pieces of text and that Support Vector Machines (SVM) performed much better for sentiment classification in longer documents. Using this as motivation, we implemented the SVM algorithm via Python's Scikit-learn library using default parameters. Wang and Manning briefly mention that Logistic Regression can be used in place of SVM and provide similar results, they did not complete any formal tests with Logistic regression models. This project seemed like a good opportunity to formally check this comparison between SVM and Logistic Regression, this propelled us to choose Logistic Regression as our second scikit-learn model. Similar to SVM, we used the default settings for logistic regression with the exception of tuning the inverse of regularization strength parameter C. This value was found on a simple search over the set {0.01,0.1,1,10,100, 1000}, C=10 seemed to provide a good cross validation accuracy, regardless of features present and did not increase the risk of overfitting the model significantly (Figure 1b). Bifet and Frank acknowledge that decision trees are not typically used in sentiment classification tasks due to inferior performance relative to linear classifiers [1] however we were curious to see how significant the accuracy difference would be between methods. These models were evaluated with multiple features including TF-IDF, Unigrams and Bigrams, extracted via the Sci-Kit Learn library.

Results

Our first evaluation was testing our Bernoulli Naive Bayes (BNB) model built from scratch to establish a training performance baseline. The model utilized unigrams only and its performance was measured by a 5-fold cross validation. To ensure the model ran in reasonable time we only utilized half the training data (6250 instances of positive

reviews and 6250 of negative), from here we proceeded with cross validation normally. The Bernoulli Naive Bayes model had a 85.4% average accuracy across all 5 data folds. Although our implementation was not fully optimized and did not permit our cross validation to be performed on the entire training set, the very low variation between fold accuracies (Table 1) demonstrates that the average fold accuracy reported is reliable and can serve as a solid baseline for model comparison.

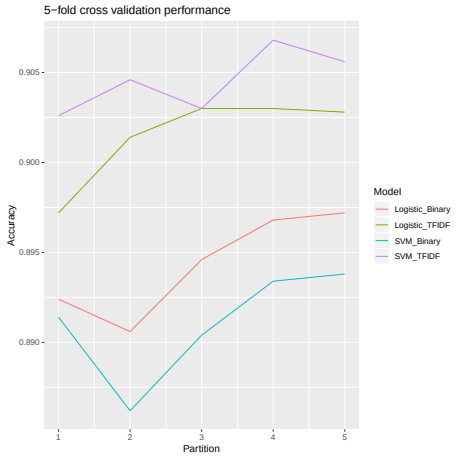
Our next step was to try and improve upon our baseline performance with Logistic Regression, Support Vector Machines and Decision Trees. We experimented with multiple feature sets for each model type. All of these models utilized both unigrams and bigrams. We compared TF-IDF weighting and binary data for each model. Finally, we also attempted to leverage lexicon information to create additional features to improve model accuracy (Table 1). One such set of features are the VAD features. VAD represents 4 additional features consisting of the mean valence, mean arousal and mean dominance of words in a review [10]. Another attempt was made with the CONOP features. CONOP represents 4 features consisting of negative/positive opinion word counts and negative/positive connoted word counts in a review [3]. Unfortunately, the addition of such features had a negligible effect on the performance of our models. All models were evaluated using a 5-fold cross validation process, the same as for the Bernoulli Naive Bayes model.

Model	Features	Mean Acc.	Acc. Var.	Runtime(s)
BNB	Unigrams	0.854	0.00125	217
Logistic Reg.	Uni+Bigrams+TFIDF	0.901	0.00248	16
Logistic Reg.	Uni+Bigrams+TFIDF+VAD	0.894	0.00290	15
Logistic Reg.	Uni+Bigrams+TFIDF+CONOP	0.894	0.00235	17
Logistic Reg.	Uni+Bigrams+Binary	0.894	0.00283	21
SVM	Uni+Bigrams+TFIDF	0.905	0.00176	> 3600
SVM	Uni+Bigrams+Binary	0.891	0.00305	> 3600
Decision Tree	Uni+Bigrams+TFIDF	0.704	0.00513	261
Decision Tree	Uni+Bigrams+Binary	0.713	0.00677	290

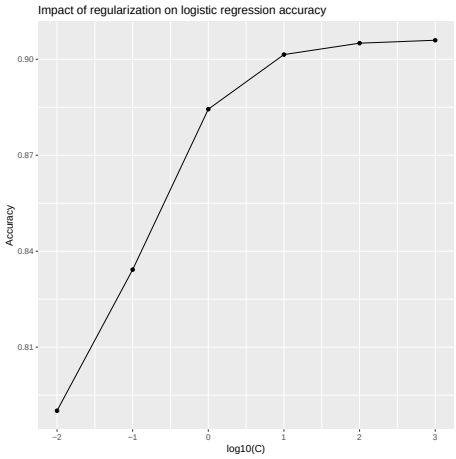
Table 1: Comparison of average validation accuracy for each model and feature set. Also shown, variance of accuracy across the 5 validation folds and runtime of model.

The Decision Tree models performed poorly compared to the Bernoulli Naive Bayes baseline. SVM and Logistic Regression were our best performing models. When used with TF-IDF weighting these models were more accurate at classification then when using binary data. This improvement is consistent across all 5 data folds (Figure 1a). The SVM TF-IDF model had the highest accuracy during the validation process, however the long runtime required to train the model was a deterrent from selecting this as our final model. Due to this significant runtime it was not feasible to tune any hyper-parameters for the SVM models (as opposed to the Logistic Regression models), this made us less confident in how well our top SVM model would generalize to the test set. Tuning hyper-parameters is very important in optimizing model performance, varying the C parameter for the Logistic Regression model has a significant impact on model accuracy (Figure 1b).

Our final selected model was the Logistic Regression model (C=10) that utilized Unigrams, Bigrams and TF-IDF weighting. On the 30% subset of the test set made



(a) Comparing TF-IDF weighting and Binary data for Logistic Regression and SVM models across all 5 data folds.



(b) Examining the effect on Logistic Regression accuracy when hyper-parameter C is varied.

Figure 1

available via Kaggle, this model had a 90.45% accuracy.

Discussion and Conclusion

We have performed sentiment classification on this set of IMDB reviews with the aid of four classification models. Logistic Regression and Support Vector Machines were seen to have the best classification accuracy in general. The favourable runtime of Logistic Regression allowed for proper hyper-parameter tuning. This influenced us in selecting a Logistic Regression model with Unigrams, Bigrams and TF-IDF weighting as our final model, even though the equivalent SVM model performed slightly better in terms of validation accuracy. Experimentation with lexicon based features did not result in a fruitful outcome for this dataset. As a future investigation, it could be worthwhile to learn the models over a short, selected vocabulary from this large database. The poor performance of the Decision Tree models could be improved with ensembling techniques such as Random Forests or AdaBoost. Finally, even more complex models such as neural networks may prove to be the best avenue for a sentiment classification task such as this one.

Statement of Contributions

The workload of the project was almost equally divided among the three members. The communication between the team members was excellent. Even though Jonathan Pearce and Cristian Groza worked more on the coding and Saima Tazreen contributed more towards the write-up, we all sat together to discuss the ideas and complete the requirements.

References

[1] Albert Bifet and Eibe Frank. "Sentiment Knowledge Discovery in Twitter Streaming Data". In: *Proceedings of the 13th International Conference on Discovery Science*. DS'10. Canberra, Australia: Springer-Verlag, 2010, pp. 1–15. ISBN: 3-642-16183-9, 978-3-642-16183-4. URL: http://dl.acm.org/citation.cfm?id=1927300.1927301.

[2] Cicero Dos Santos and Maira Gatti de Bayser. "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts". In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics. Technical Papers, Aug. 2014, pp. 69–78.

[3] Song Feng et al. "Connotation Lexicon: A Dash of Sentiment Beneath the Surface Meaning". In: *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, 2013.

[4] Minqing Hu and Bing Liu. "Mining and Summarizing Customer Reviews". In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '04. event-place: Seattle, WA, USA. New York, NY, USA: ACM, 2004, pp. 168–177. ISBN: 1-58113-888-1. DOI: 10.1145/1014052.1014073. URL: http://doi.acm.org/10.1145/1014052.1014073.

[5] H. M. Keerthi Kumar, B. S. Harish, and H. K. Darshan. "Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method". In: *International Journal of Interactive Multimedia and Artificial Intelligence* In Press.In Press (2018), pp. 1–7. ISSN: 1989-1660. DOI: 10.9781/ijimai.2018.12.005. URL: http://www.ijimai.org/journal/sites/default/files/files/2018/12/ip18_12_05_pdf_37064.pdf.

[6] Edward Loper and Steven Bird. "NLTK: The Natural Language Toolkit". In: *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*. ETMTNLP '02. event-place: Philadelphia, Pennsylvania. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 63–70. DOI: 10.3115/1118108.1118117. URL: https://doi.org/10.3115/1118108.1118117.

[7] Hadi Pouransari. "Deep learning for sentiment analysis of movie reviews". In: 2015.

[8] Anuj sharma and Shubhamoy Dey. "Performance Investigation of Feature Selection Methods". In: *arXiv e-prints* (Sept. 2013), arXiv:1309.3949.

[9] Sida Wang and Christopher Manning. "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2012, pp. 90–94. URL: http://aclweb.org/anthology/P12-2018.

[10] Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. "Norms of valence, arousal, and dominance for 13,915 English lemmas". In: *Behavior Research Methods* 45.4 (Dec. 2013), pp. 1191–1207. ISSN: 1554-3528. DOI: 10.3758/s13428-012-0314-x. URL: https://doi.org/10.3758/s13428-012-0314-x.