COMP30027 Report

Anonymous

1 Introduction

The fast developing world of technology has given us the opportunity to freely express our ideas with the tap of a few buttons, with any degree of anonymity to our liking. The micro-blogging site Twitter, has become one of the most popular social media sites for expressing thoughts, with hundreds of millions of tweets being generated per day. With such a large amount of data, comes an opportunity to extract information, and learn more about the inherent meaning of tweets.

These tweets often include unstructured language like slang, hashtags, account handles and misspelt words. And often they only express half an idea, making them even more challenging to comprehend.

Sentiment analysis on sentences has been a much studied field, and is mainly concerned with identifying and classifying opinions or emotions expressed through text.

It depends largely on user-generated content, whether it be product review generalisation, comment filtering etc. so the content may be difficult to automatically assign an emotion to. But with such a large amount of data, automation seems nothing short of necessary to apply.

In this paper, we aim to use machine learning models to evaluate the sentiment of tweets, and compare the efficiencies of each model against another. We hope to achieve predictive models that perform well against real world data and be able to explain each model's performance.

2 Literature review

Sentiment analysis has existed since the early 2000s(Pang et al., 2002). In recent times, many social media sites like Twitter, Instagram and Facebook, have acquired hundreds of millions of users, with millions of posts, and tweets each day.
Using computerized methods and algorithms to extract the polarity of a document (tweet in our case) is very useful in these case.

The two main methods of methodology are symbolic techniques, and machine learning techniques(Riya Suchdev et al., 2014), where symbolic techniques include training the model to learn data how a brain would learn, and machine learning trains a model by using algorithms and classifiers. We used the latter, making use of common classifiers such as Naive bayes, Logistic regression and SVM.

Sentiment Categorization is usually either binary (positive and negative) or ternary (positive, negative and neutral). Our data set(Mukherjee et al., 2014)(Rayana and Akoglu, 2015) gives each instance an ID number, a tweet, and the related sentiment.
The Evaluation method most commonly used in general is accuracy, because for multiclass labels it is fairly accurate and simple to apply.

3 Method

First we must split the data into train and test data , which has already been done, but only 0.4 of the data had been shown to test, the rest of the 0.6 will be used to test without our knowledge of their accuracy scores. The aim of Sentiment analysis is to find key elements in tweets used to identify its emotion. Extraction of this useful information included:

3.1 Data cleaning and preprocessing

With the inconsistency of a standard structure and language of tweets, data cleaning is imperative prior to letting the model learn. The NLTK library was used in this process for lemmatizing, tokenizing and stop word removal, whilst regu-

lar expressions were used to handle some of the more specific data filtering tasks.

- **Data Removal**
  Removed data which was completely irrelevant, this included hyperlinks, twitter tags, and numbers (with their proceeding strings such as 5th...). Regular expressions were mainly used for this data removal, due to its simplicity. Also removed stop words using the NLTK library, which are not classified as completely irrelevant words, but they can get in the way of classifying correct labels.

- **Data modifying**
  The current language structure of the tweets is not in its most ideal state to take into processing. Modification of data is done using the NLTK library, where data is first tokenized, and is lemmatized after. Contractions are also expanded out into their normal form using the contractions library.

3.2 Feature Selection

To extract features from the tweets,(string of characters), two text vectorization methods were thought of. Bag Of Words and TF-IDF. For bag of words, each word has its frequency as its value, while for TF-IDF, each word records its TF-IDF score. The only non stop word which was included was 'not', because of its ability to change the meaning of a sentence.

3.3 Classifiers

All the classifiers used were trained with both the BoW vectorizations and the TF-IDF vectorizations, so we have 2 models per classifier

- **Zero-R**
  The Zero R classifier is a baseline classifier, where the model does not inspect the features, it only inspects the class label, and classifies all new test instances as the class label with the maximum frequency. This model in general performs relatively badly because the model is not learning anything from the features. However it is a good baseline to start and compare against better models.

- **Naive Bayes**
  The Naive bayes classifier, is a probabilistic classifier which has an assumption of independency between features. It uses prior knowledge (Bayes Theorem) to estimate the likelihood of occurences(Shriram, 2021). Using BoW vectorization data with the multinomial naive bayes classifier aids the model due to its strength in handling discrete values(word counts). Usage of Naive Bayes can give very accurate results, because even though our data is not independent, our system is not too complicated, meaning the independence assumption can be fair. Naive bayes can also scale well to high dimensional datasets.

- **Support Vector Machines**
  These are supervised learning models used for classification, regression and outlier detection. It uses decision planes to split the data into its respective portions, and classifies an instance depending on which portion it falls into. Generally they are used because they provide exceptional performance over other classifiers, and are very efficient in high dimensional spaces. We are using LinearSVC for this classifier because it is more flexible in the choice of penalties and loss functions and should scale better to a large number of functions(Moradizadeh, 2022)

- **Logistic Regression**
  Logistic regression turns a regression model into a classification model, It models the posterior distribution directly using regression. Multiclass logistic regression is used in python due to having a ternary class label. This model makes no assumption of the underlying distribution of data, and so it is simple to train and interpret, and could be more accurate. It also gives more weight to the features that help with distinguishing between the classes, as a single word can very be very indicative of a sentence's sentiment.

3.4 Evaluation Methods and Metrics

Evaluation of the models using the testing data occurred on the Kaggle competition prediction page, where accuracy was used to rank the accuracy score against others' models. Cross validation will be performed on the training data, and we will calculate the f1score, precision and recall.Accuracy can be misinterpreted if the data is overfitted, so these may be good metrics to have. Precision is how often the model is correct when it predicts a positive case and recall is what proportion of true positive cases the model

was able to detect. F1 score is the combination between the two.

4 Results

4.1 Preprocessing and Text vectorization

The Training dataset has 21802 tweets and sentiments, here is the result of preprocessing:
Number of features using vectorized text before cleaning: 44045
Number of features using vectorized text after cleaning: 24597

4.2 Models

Did not include the Zero-r baseline model, because it contains 1 out of 3 labels, so the f1score will be 0 for labels which do not exist in the baseline model predictions.
Each graph is a different score on which evaluation method has been used:

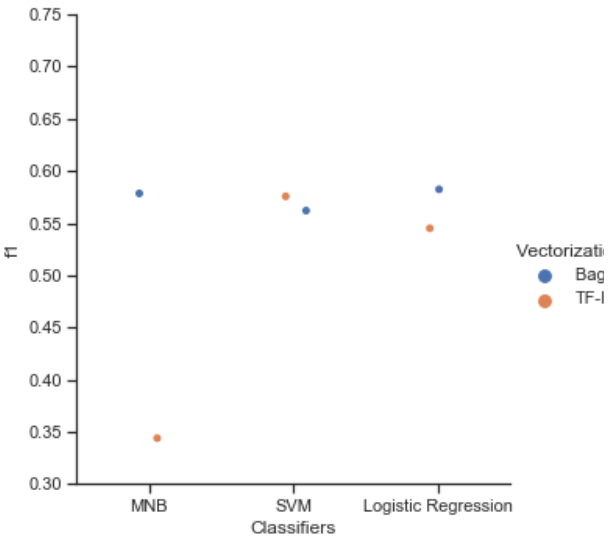

Figure 1: F1-score

4.3 Resulting data

The data for Naive bayes is a lot more volatile than the other 2 classifiers, while SVM and Logistic Regression seem to have relatively consistent scores across both text vectorization methods, and through all scores. Multinomial Naive Bayes obtains the lowest scores for accuracy and f1score when using tf-idf, otherwise, most of the scores lie between 0.55 - 0.6

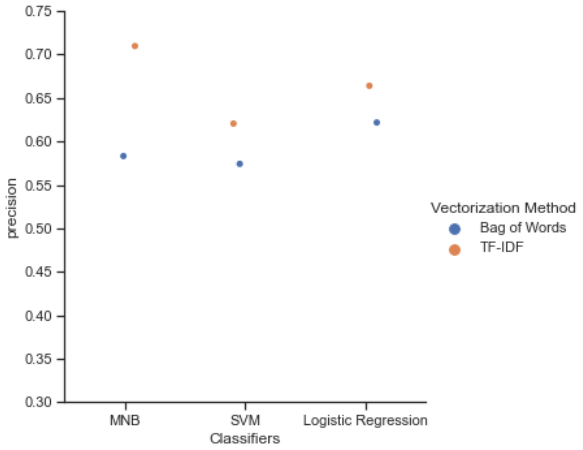

Figure 2: Precision-score

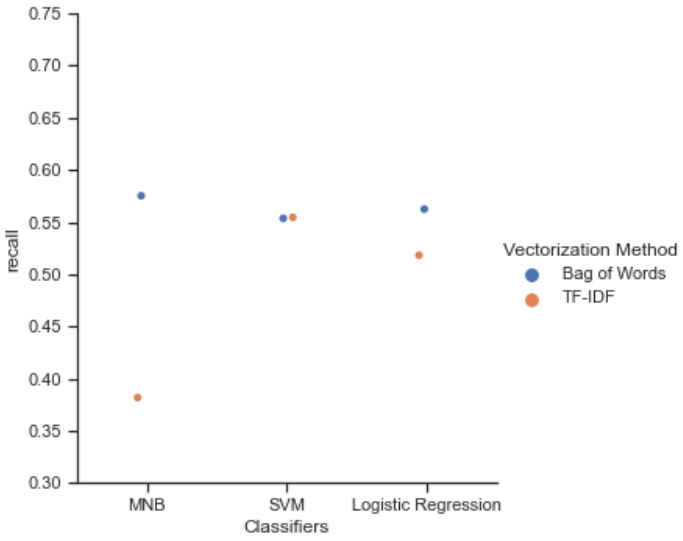

Figure 3: Recall-score

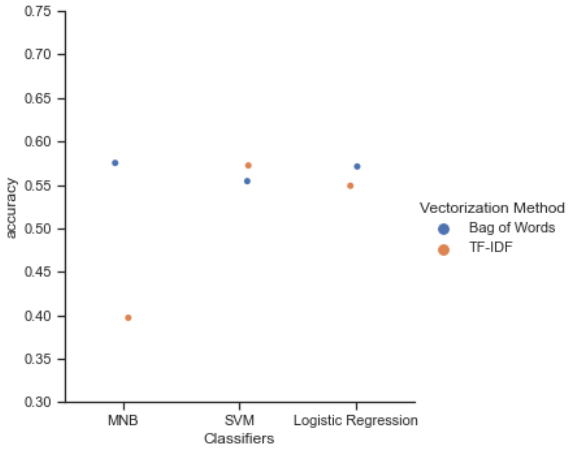

Figure 4: accuracy-score

Classifiers	Vectorization method	F1	Precision	Recall	Accuracy
MNB	Bag of Words	0.578480	0.583170	0.575303	0.57523
MNB	TF-IDF	0.344189	0.709599	0.382057	0.39729
SVM	Bag of Words	0.561988	0.574184	0.553677	0.55432
SVM	TF-IDF	0.575672	0.621722	0.554597	0.57236
Logistic Regression	Bag of Words	0.582393	0.582393	0.562459	0.57113
Logistic Regression	TF-IDF	0.545083	0.663907	0.518320	0.54899

Figure 5: table-of-scores

5 Discussion

In our research, each classifier is found to have its own behaviour. It can also depend on the type of text vectorization you apply to the data. The Accuracy of the Naive Bayes model is the highest when we use the BoW data to vectorize it. But it can be misleading to determine the Naive bayes as our best classifier, since both SVM and Logistic regression have similar accuracy scores, but a lot higher f1 scores. When we examine the F1-scores, precision and recall, we find that SVM and Logistic regression actually perform better than the Multinomial naive bayes on average, and has a lower variance(SVM has a lower variability when comparing using both BoW and TFIDF). Multinomial Naive bayes with tfidf has a very low accuracy and f1 score, because MultinomialNB uses discrete values not continous ones provided by TF-IDF(Which is why BoW works well for this model).

SVM works resiliently with tfidf(BoW is not bad either), and the high dimensionality of data does not affect the data as much as it does for other models, because there is dimensionality reduction. Linear SVMs use the kernel trick to map the data to a smaller dimension, and then find the soft margins, which should return majority of relevant words, which increases the F1 score.

Logistic Regression works similary to SVM and the scores are indicative of this. The main difference between Logistic Regression and SVM is that the logistic regression data must be more structured, something our data lacks (word vectorizations are not entirely indicative of a word value). Its f1 score is still comparable to that of SVM but SVM will scale better with unstructured data because it assumes no posterior or prior distribution(Zach, 2021).

All models perform better than the dummy classifier, which was expected.

6 Conclusion

Twitter sentiment analysis can be efficiently trained using Machine learning algorithms. It focuses on learning the importance of words attaching them to a sentiment. SVM or Logistic Regression Classifier can be well suited to this use case, if proper data cleaning and text vectorization is used. Currently some analysis methods have been able to achieve 85%-90% accuracy levels(Gupta et al., 2017). However Twitter sentiment analysis is not without its flaws. Slang words, trending twitter hashtags and misspellings are not used in this analysis. This application in other industries can also be further researched. There can still be room for development in this field, but I have no doubt it will continue to evolve.

References

Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. 2014. What Yelp fake review filter might be doing? In *7th International AAAI Conference on Weblogs and Social Media*.

Shebuti Rayana and Leman Akoglu. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 985–994.

Badhani, P., Gupta, B., and Negi, M. (2017). Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python (Review of Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python). In ResearchGate (pp. 29–34). researchgate.

Ahmad, M., Ahmad, S., Aftab, S. and Muhammad, S. (2017). Machine Learning Techniques for Sentiment Analysis: A Review. researchgate, pp.27–32.

Suchdev, R., Kotkar, P., Ravindran, R. and Swamy, S. (2014). Twitter Sentiment Analysis using Machine Learning and Knowledge-based Approach. International Journal of Computer Applications, pp.36–40.

Fueyo, E. (2020). Understanding What is Behind Sentiment Analysis – Part 1. KDNuggets, pp.1–5.

Shriram (2021). Multinomial Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2022. upGrad, pp.1–4.

Pang, B.P., Lee, L. and Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. Cornell, pp.1–8.