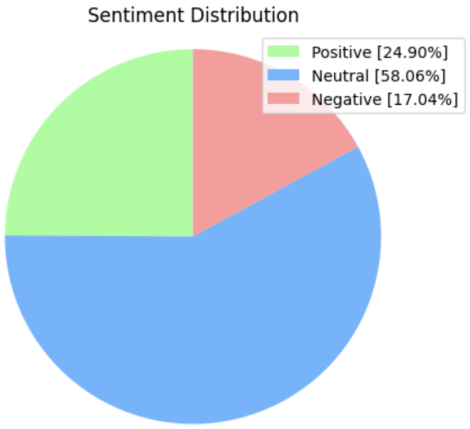
COMP30027 Report

1. Introduction

As the rapid growth of social media in modern days, more researchers are getting interested in information behind texts. Sentiment analysis is a popular method that is based on text information to classify the emotion revealed by the textual content (Habernal et al., 2013). This project aims to predict the sentiment of tweets measured from three point scales: “neutral”, “positive” and “negative”.

This dataset consists of more than 20k original tweets, unevenly distributed with more than half data with neutral sentiments. Unlike traditional reviews, tweets are more casual and not necessarily theme based (Rosenthal, 2017).

Figure 1: sentiment distribution of training dataset



This paper is organized as follows. Method section focuses on two aspects: Data preprocessing including feature selection and machine learning classifier’s development. Finally, results will be presented and evaluated.

2. Method

Statistical models including Multinomial Naive Bayes, SVM, Multinomial Logistic Regression will be applied as individual learners. Some models’ performance can be greatly affected by hyperparameters. The process of parameter tuning is assisted with grid search and cross validation technique, which will be introduced in detail and explained. In addition to individual learning models, ensemble systems are proven to be highly efficient(Polikar, R., 2012). In order to further increase the performance of individual classifiers, ensemble stacking is applied.

2.1 Preprocessing

Replacing common patterns

In order to reduce the complexity of the input data some patterns were either removed completely or replaced with a less complex alternative, these are as follows in Table 1.

As URLs and usernames do not convey any meaning on their own and contribute noise to the dataset they are replaced with the replacements specified in table

description	pattern	replacement
urls	http\S+	URL
username @’s	@[A-Za-z0-9]+	USERNAME

Table 1: Patterns that are filtered out

Initial instincts point to removing punctuation and special characters, but as seen in table 2 there were uneven distributions of certain special characters across the labels. For this reason no punctuations were filtered out.

	% of labels with symbol		
symbol	positive	neutral	negative
!	28%	9%	10%
<	0.22%	0.11%	0.08%
(	4.2%	4.9%	2.9%
*	0.36%	0.54%	1.13%

Table 2: Patterns not filtered out

All together the preprocessing described above reduced the amount of unique values to 76% of the non-filtered set of words.

Vectorisation

Two vectorization methods were supplied; Bag of Words (BoW) and term frequency–inverse document frequency (TFIDF).

Across various untuned and tuned models Bag of Words consistently outperformed TFIDF, and feature reduction only helped to increase this difference. Along with this higher accuracy also came a risk of overfitting as can be seen in table 2. Although this was the case, the issue seemed to be greatly influenced by the overfitting of only the Linear SVM classifiers that had an accuracy on the training data of 98%. Because no other classifiers had this same issue, it was decided that the BoW method was the preferred vectorisation method as it has a higher accuracy than TFIDF.

Method	Average Train Accuracy	Average Test Accuracy
Bag of words	0.8041	0.6111
Tf-IDF	0.6374	0.5788

Table 3: Average accuracy vectorization methods

2.2 Feature selection

Initially we used a variety of unoptimised classifiers in order to calculate the number of features in our BoW vectorisation. In order to accomplish this, a 70/30 testing split was made and unoptimised classifiers were run on the following.

- MultinomialNB
- LogisticRegression
- LinearSVC
- RandomForestClassifier

After this step the average accuracy was calculated in order to select the optimal amount of features.

To select the best amount of features, different models were chosen as seen in table 4, and the different accuracy scores.

Because a value of 500 maximised the average accuracy any further models that are trained and evaluated against Bag of Words will have a feature value of 500.

Number of attributes k	Average accuracy
10	0.6083
50	0.6385
100	0.6501
150	0.6552
500	0.6559
1000	0.6533
2000	0.6487

Table 4: Feature selection Accuracy

2.3 Classifiers

0R

Chosen for simplicity and intuitiveness for a base classifier.

Multinomial Naive Bayes

Naive Bayes classifiers are one of the most popular classifiers for text processing. Drawbacks of Multinomial Naive Bayes include:

- The assumption of complete independence from one word to another.
- The disregard for word ordering in a sentence.

This essentially means that all context of a word in a sentence is lost.

Random Forest

Decision Trees can filter out irrelevant attributes that occur in our text (gain=0). Initial

inspections showed some noise in the data around repeated words, punctuation, and tokens, as well as many stop words. Because of this Random Forest could be ideal for classifying noisy data.

Unfortunately Random Forests and Decision Trees do not handle large data sets very well as they are prone to overfitting but hopefully this can be overcome by reducing the depth of the trees.

SVM

SVM is not prone to overfitting which makes it an ideal model for our uses, along with this Linear and RBF kernels are tested with the hypothesis that RBF kernels will greatly outperform Linear kernels. This hypothesis comes from the fact that the decision boundary between the classes is expected to be too complicated for a linear SVM.

Logistic Regression

Multinomial logistic regression makes predictions based on multiple independent variables. This is an efficient model when we want to make categorical predictions and the dependent variables are not entirely continues

Ensemble Stack/Voting

Combining all of the above methods, an Ensemble will combine the predictive power of both of them to lower variance and increase test performance.

Hyperparameter Tuning

Grid search was used in parameter searching for hyperparameters and a 3 fold cross validation was used.

The parameters that were chosen for parameter tuning were the ones that varied the model's complexity (max_depth, n_estimators) as well as parameters that impact the selection of features (gamma)

Hyperparameter tuning showed a huge difference in models such as SVN and logistic regression, but the Random Forest for example could not be optimised very well because of limitations with the models themselves.

3. Results

Out of the models tested, rbf SVM, Naive Bayes and Logistic regression were all very similar in terms of average accuracy.

Model	Training Accuracy	Test Accuracy
0R Baseline	0.5787	0.5834
Random Forest	0.8181	0.6421
SVM - Linear	0.7012	0.6507
SVM - RBF	0.7120	0.6601
MNB	0.8218	0.6506
Logistic Regression	0.7063	0.6625
Ensemble Stacking	0.7064	0.6731

Table 5: Training and Test Accuracy across different models

4. Discussion

0R

As expected the largest label has a recall of 1, and precision of 0.59, which is the amount that neutral labels show in the testing set.

label	precision	recall	f1-score
negative	0.00	0.00	0.00
neutral	0.59	1.00	0.74
positive	0.00	0.00	0.00

Table 6: evaluation of 0R classifier

Multinomial Naive Bayes

label	precision	recall	f1-score
negative	0.50	0.46	0.48
neutral	0.70	0.75	0.73
positive	0.61	0.53	0.57

Table 7: precision, recall and f1 score for logistic regression

The Multinomial Naive Bayes algorithm had

one of the highest values for recall for minor labels (negative, positive). This is likely because Naive Bayes calculated the relative frequency of each term and is less likely to be negatively influenced by an uneven amount of data for some labels than other classifiers.

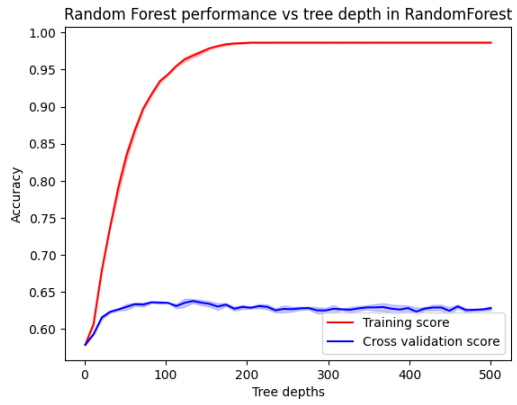
Random Forest

label	precision	recall	f1-score
negative	0.66	0.13	0.22
neutral	0.64	0.91	0.75
positive	0.65	0.36	0.46

Table 8: evaluation of random forest

Random forest had the highest variance throughout all the classifiers that we tested. The accuracy of Random Forest seemed to peak at around 150 at the same tree depth that maximises the training score.

Figure 2: Tree depth performance on Random Forest accuracy performance



Unfortunately the random forest performs poorly because there is an uneven class distribution in the training set (58% neutral). This is reflected in the very low recall scores in table 7 as there are many false negatives for the labels “negative” and “positive”. In order to increase this recall statistic a smaller tree depth could be chosen, but this would lead to lower accuracy. Another option would be to weight features based on their frequency in the training set.

SVM

Linear SVM

	precision	recall	f1-score
negative	0.54	0.32	0.40
neutral	0.67	0.85	0.75
positive	0.66	0.43	0.52

Table 9: evaluation of linear kernel SVM

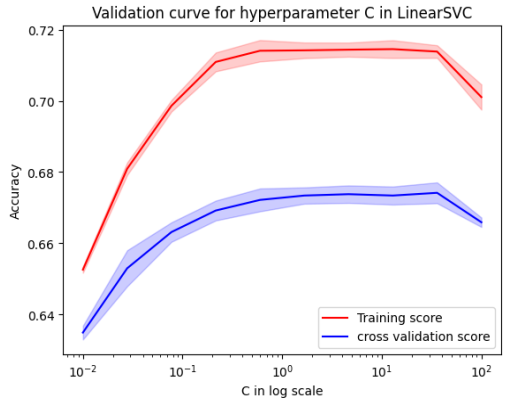
Radial Basis Function SVM

label	precision	recall	f1-score
negative	0.54	0.32	0.40
neutral	0.68	0.86	0.76
positive	0.68	0.43	0.52

Table 10: evaluation for RBF kernel SVM

Linear and RBF kernels were tested for SVM classifiers, Initially it was suspected that the RBF Kernel would work better on the dataset due to the presence of non binary labels. We originally suspected this model would vastly outperform the linear model as multiple classes exist. The fact that average accuracies for RBF are so similar to SVM indicates that the decision boundaries between the two models have converged.

Figure 3: Hyperparameter C on Linear SVM accuracy



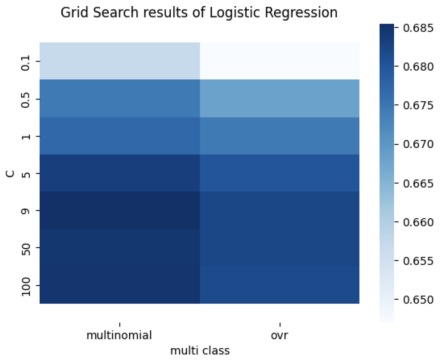
Logistic Regression

label	precision	recall	f1-score
negative	0.62	0.33	0.43
neutral	0.68	0.87	0.76
positive	0.67	0.45	0.54

Table 11: precision, recall,f1 score for logistic regression

Multinomial logistic regression generalized logistic regression for the situation where classification has more than two possible outcomes (Ramadhan et al., 2017).

Figure 4: Grid search results on Logistic Regression

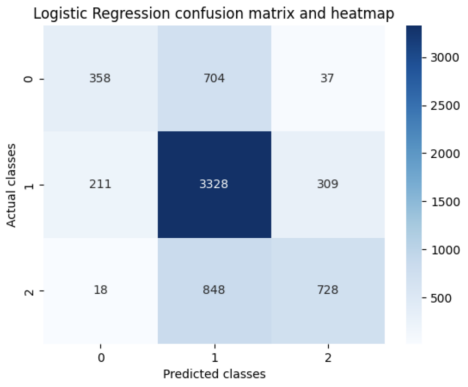


Two logistic regression models (multinomial and OVR) are tested with C values ranging from 0.01 to 100. From the above graph, both multinomial and OVR(one-vs-rest) generates great performance when C value lies around 10. Multinomial is chosen to optimize logistic regression model performance since it generates higher accuracy at every C value. OVR assumes independence of outcomes, which makes it more suitable for binary classification problems.

The above data is the classification report of a logistic regression model using optimal C from grid search. The recall score is 0.87 for neutral sentiment prediction, 0.33 and 0.45 for "negative" and "positive" respectively. The large gap is showing the model is biased.

The heat map below provides a more intuitional representation. A large number of "negative" and "positive" sentiments are falsely predicted as "neutral". We conduct that this data skew is the result of the uneven distribution in the original dataset. Given the size of the data is limited, methodologies like cross validation and ensemble stack are applied for improvement.

Figure 5: logistic regression confusion matrix and heat map



2.4 Ensemble

Ensemble learning methods are expected to generate better prediction as they are built based on multiple base classifiers. The mechanism of the ensemble models reduces variances, hence further improved the reliability of the model and increased the average accuracy.(Polikar, R., 2012)

Different combinations of classifiers are experimented and finally the optimal combination for ensemble voting consists of 5 classifiers(SGD Classifier, Decision Tree, Logistic Regression, Multinomial Naive Bayes, Linear SVC). The test accuracy of 0.674 outperformed all the individual classifiers.

Ensemble Stack

Meta Classifier	Testing Accuracy
Logistic Regression	0.6767
Decision Tree Classifier	0.584
Multinomial Naive Bayes	0.6667
Linear SVC	0.6731

Table 12: Base classifier: Logistic Regression, Multinomial Naive Bayes

Meta Classifier	Testing Accuracy
Logistic Regression	0.6312
Decision Tree Classifier	0.6643
Multinomial Naive Bayes	0.5709
Linear SVC	0.6332

Table 13: Base Classifiers: Logistic Regression, Multinomial Naive Bayes, K Neighbours

Ensemble stacking's performance largely depends on the base classifier's quality. Graph above records parts of the procedure of selecting the best combination of base classifiers. This shows that including as many models as base classifiers does not necessarily lead to a better performance.

5. Conclusion

The best single method of classifying that we found was a match between Logistic regression and Linear SVC. Combining multiple methods we can improve our classification accuracy with an ensemble stack that uses Multinomial Naive Bayes, Logistic regression as base classifier and SVC as meta classifier in order to get a better performance than any one individually.

Future improvements

Removal of stop words

Removing stop words didn't seem to make any difference in our testing, although relevant literature (Giachanou Crestani 2016) states removing stop words as a pretty standard part of sentiment analysis. With this being said other literature does exist that makes no mention of stop words so it is assumed they are not removed (Go, 2009)

Lemmatization

Is the process of grouping similar words together so they can be analysed together. As Tweets come in many different forms this could help reduce the noise around similar words in the dataset.

Alternative Vectorisation

Only the supplied Tf-idf and BoW vectorisations were supplied. Using other vectorisation methods and more libraries to compare (word2vec, Bert) could give an indication of if a better vectorisation method could be used.

Doc2vec method is more suitable for longer text content, so will not be efficient for short sentences like tweets. Word2vec however, transforms data into vectors rather than numbers like BoW, which adds visual information in addition to the contextual info (Feng and Thuremella, 2018). Bert however takes the positions of the words into account. These models are highly accurate with natural language processing if provided with suitable data. However, when it comes to analysing a limited dataset of tweets, the result might not be optimal. As mentioned in this paper, tweets are more casual and worse textual structured compared to articles like review or news. In this case, BoW with feature selection provides high efficiency and flexibility is still considered a good option.

6. References

Mukherjee, A., Venkataraman, V., Liu, B. & Glance, N. *What Yelp fake review filter might be doing? 7th International AAAI Conference on Weblogs and Social Media*, 2013.

Rayana, S. & Akoglu, L. *Collective opinion spam detection: Bridging review networks and metadata.* Proceedings of the 21th ACM SIGKDD International Conference on

Knowledge Discovery and Data Mining, 2015. 985-994.

Go, A., Bhayani, R. and Huang, L., 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, *1*(12), p.2009.

Giachanou, A. and Crestani, F., 2016. *Like it or not: A survey of twitter sentiment analysis methods. ACM Computing Surveys (CSUR), 49(2)*, pp.1-41.

Habernal, I. Ptacek, T. & Steinberger, J. *Sentiment Analysis in Czech Social Media Using Supervised Machine Learning*, 2013

Polikar, R. *Ensemble Learning.* In: Zhang, C., Ma, Y. (eds) Ensemble Machine Learning. Springer, Boston, MA. https://link.springer.com/chapter/10.1007/978-1-4419-9326-7_1

W. P. Ramadhan, A. Novianty and C. Setianingsih, *Sentiment analysis using multinomial logistic regression*, Dec. 2017.

Feng, B. Thuremella, D., *A Tale of Two Encodings: Comparing Bag-of-Words and Word2vec for VQA*, 2018

Rosenthal, S. N. *SemEval-2017 Task4: Sentiment Analysis in Twitter. Proceedings of the 11th International Workshop on Semantic Evaluation . Vancouver, Canada: SemEval '17,* 2013