Sentiment Classification of Tweets

1. Introduction

1.1 Background

Sentiment analysis is the process of classifying textual instances into discrete classes based on the overall sentiment within the text. It has a range of real-world applications, for example, in a business setting it can be utilised to filter brand reviews or company feedback, as well as improve targeted online advertising. This paper presents a comparative study evaluating the effect of applying different feature selection methods and classifiers to determine the most effective method for sentiment analysis.

1.2 Dataset

The 'Train' dataset contains 21,815 instances, each consisting of an ID (ignored for the purpose of sentiment classification), a string of textual information and a sentiment label. The textual information from each instance contains alpha-numeric characters, punctuation, URLs, hashtags as well as usernames, all which will be considered for feature selection (see Section 2.2).

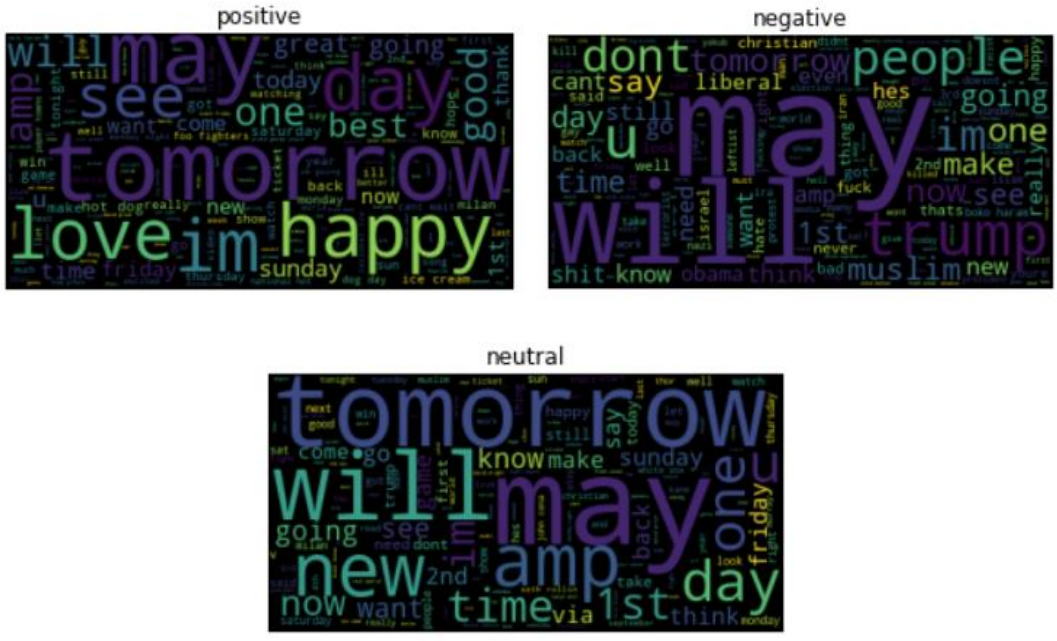


Figure 1: Wordcloud diagrams generated from each class before processing the dataset using TF-IDF

The class wordclouds shown in Figure 1 give an overview of the most frequent words from each class. Since they were generated before the application of TF-IDF many of the words (especially in the case of the neutral class) appear to be common/non-descriptive stop words (e.g., will, may, you).

Since these words appear frequently in all classes their importance will be reduced after applying the TF-IDF vectorisation due to their frequent occurrence.

2. Method

2.1 Pre-processing

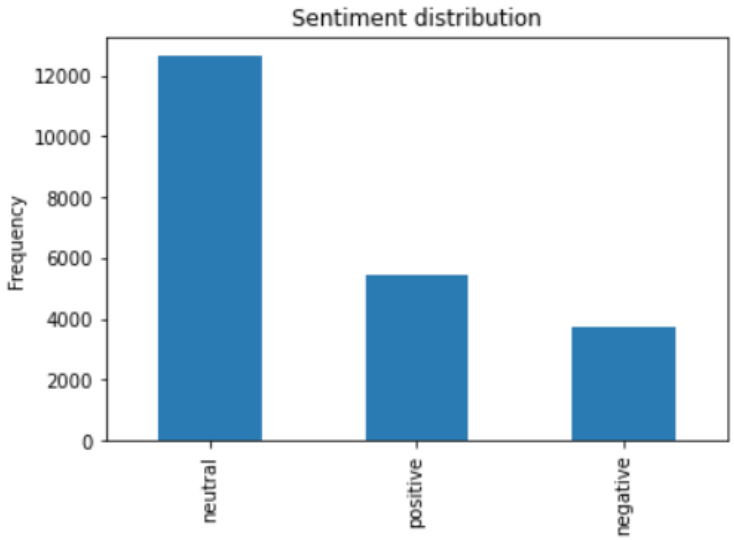


Figure 2: A bar graph showing the distribution of class labels within the dataset

As shown in Figure 2., the dataset's class labels are significantly imbalanced which has the potential to cause biases in classification and result in poor generalisation (Zheng et al., 2015). In response to the class imbalance, two different resampling techniques were tried: Random Under-sampling (RUS) and Random Over-sampling (ROS). RUS deletes examples from the majority class in the training dataset (in the case of our model the majority class is 'neutral'), with the benefit of improving computational time and storage. However, it may result in the loss of information, underfitting and can produce a non-optimal solution if the resampled training dataset is biased (not representative of the true population). ROS duplicates examples from - subsequently giving more weight to - the minority class in the training set (in our case 'positive' and 'negative' classes), which doesn't result in any information loss. Due to this, ROS often outperforms RUS (as demonstrated by the difference in accuracies and F-scores in Tables 1. and 2.).

Table 1: The effect on classifier accuracy after applying different techniques to combat class imbalance in the dataset.

Technique	**Accuracy**				
	Multinomial NB	**Linear SVM**	**Logistic Regression**	**Stacked Model**	**Average**
None	0.6007	0.6427	0.6583	0.6542	0.6390
RUS	0.4926	0.5427	0.5553	0.5507	0.5353
ROS	0.5377	0.6072	0.6235	0.6254	0.5985

Table 2: The effect on the F-score of each classifier after applying different techniques to combat class imbalance in the dataset.

Technique	F-Score				
	Multinomial NB	**Linear SVM**	**Logistic Regression**	**Stacked Model**	**Average**
None	0.3055	0.5690	0.5468	0.5674	0.4972
RUS	0.4966	0.5363	0.5475	0.5454	0.5315
ROS	0.5359	0.5726	0.5966	0.5576	0.5657

The highest average accuracy across all the models occurred when no sampling techniques were applied (see Table 1.), attributed to a similar distribution of classes in the training and test datasets. However, the increasing F-scores when sampling techniques are applied (see Table 2.) shows an increase in classifier performance over minority classes. ROS is the technique with the highest F-score and will therefore be the chosen method.

2.2 Feature Selection

Tweets contain a variety of non-alphanumeric features which may affect sentiment analysis. The removal of some of these features are shown in Table 3.

Table 3: Comparison of classifier accuracy after applying different feature reduction methods

Feature Selection	Accuracy			
	Multinomial NB	**Linear SVM**	**Logistic Regression**	**Average**
None	0.5933	0.6491	0.6609	0.6344
Username	0.5939	0.6510	0.6592	0.6347
Hashtag	0.5920	0.6461	0.6563	0.6315
URL	0.5973	0.6458	0.6588	0.6340
Non-alphanumeric characters	0.5944	0.6504	0.6580	0.6343
All	0.6007	0.6427	0.6582	0.6339

Further Investigation of Hashtags, Usernames and URLs

As demonstrated by Figure 3, the distribution of hashtags across the three classes is very different and may be correlated to a Tweet's sentiment. Table 4. depicts the results of whether the content of hashtags is helpful in predicting the sentiment (as shown by an increase in classifier accuracy). The same procedure can be applied to usernames and URLs, to determine if only removing the '@' (instead of the entire username) or replacing the URL with an appropriate token is able to increase accuracy.

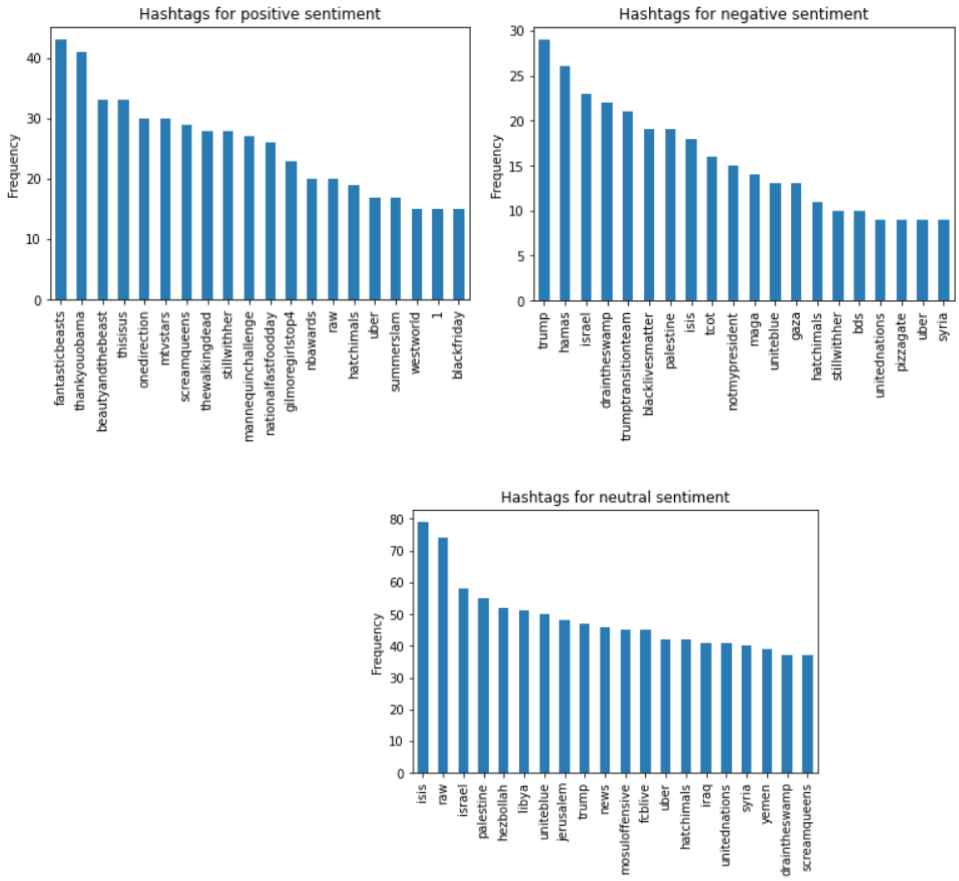


Figure 3: Bar graphs showing the top 20 most frequent hashtags from each class

Table 4: Comparison of classifier accuracy when the entire username, hashtag or URL is removed, or an alternative method

Feature Selection	Method	Accuracy			
		Multinomial NB	Linear SVM	Logistic Regression	Average
Username	Removing Username	0.5939	0.6510	0.6592	0.6347
	Removing @	0.5933	0.6502	0.6611	0.6349
Hashtag	Removing Hashtag	0.5920	0.6461	0.6563	0.6315
	Removing #	0.5930	0.6490	0.6611	0.6344
URL	Removing URL	0.5973	0.6458	0.6588	0.6340
	Replacing with token	0.5969	0.6471	0.6620	0.6353

The feature selection method resulting in the highest average accuracy across all classifiers is replacing URLs with a token (as shown in Table 4.). Therefore, this is the sole feature selection method applied to the dataset before applying TF-IDF.

2.3 Machine Learning Methods

Zero-R was chosen as a baseline model to determine a benchmark accuracy. The three other classifiers selected were Multinomial Naïve Bayes (NB), Linear Support Vector Machines (SVM) and Logistic Regression. Multinomial NB combines optimal time performance and reasonable accuracy (Gamallo & Garcia, 2014), (Ahmad et al., 2017) and works well on large datasets. Linear SVM works well on high-dimensional datasets and is efficient and adaptable to real world requirements due to multiple extensions (e.g. Soft Margins SVM) (Ahmad et al., 2017). Logistic regression is classification algorithm which makes no independence assumptions for its features (unlike NB) as well as no assumptions about distributions of classes in the feature space (Go et al., 2009). Additionally, Multinomial NB, Linear SVM and Logistic Regression have been widely used in practice as base classifiers for ensemble Stacking classifiers (da Silva et al., 2014), which we have also implemented.

2.4 Evaluation Metrics

Performance Metrics

The two main performance metrics used to assess each classifier are accuracy and F-score. Accuracy is the proportion of correctly classified classes, whilst F-score is a weighted measure of Precision and Recall. Precision is a useful measure when the cost of False Positives is high whereas Recall is a useful model metric when a high cost is associated with False Negative classifications. In sentiment analysis, both Precision and Recall are equally as important, therefore F-score is the best choice for model metric. Confusion matrices were also included for the final versions of Multinominal NB, SVM and Logistic Regression in order to observe which classes were most misclassified. This information can be used to determine if the individual classifiers are making similar errors (in which case combining them as base classifiers for an ensemble stacking classifier would be unwise).

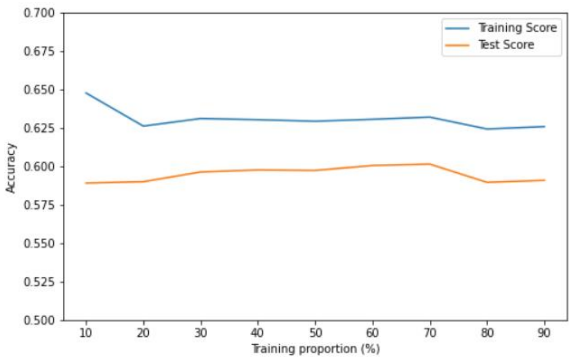
Averaging method

There are multiple choices of averaging when evaluating multiclass classification problems. Micro-averaging takes the average over the document set and typically results in performance being dominated by the majority class(es), whereas macro-averaging takes the average over the class set and typically results in performance being dominated by the minority class(es) (Prabowo & Thelwall, 2009). Macro- and micro-averaging have similar performance unless the dataset is imbalanced. Therefore, macro-averaging was chosen to allow us to observe how our classifiers are performing on the minority classes (positive and negative) when the dataset is imbalanced. If there is no significant class imbalance then macro-averaging will indicate the performance across all classes.

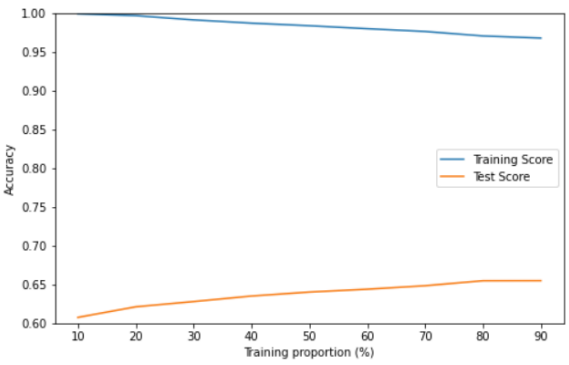
Learning Curves

The method for generating training and test data was the random-holdout method and in order to determine the optimal training-test split, learning curves with 5-fold cross validation depicting the accuracy of each classifier were explored (illustrated in Figure 4.).

(i)



(ii)



(iii)

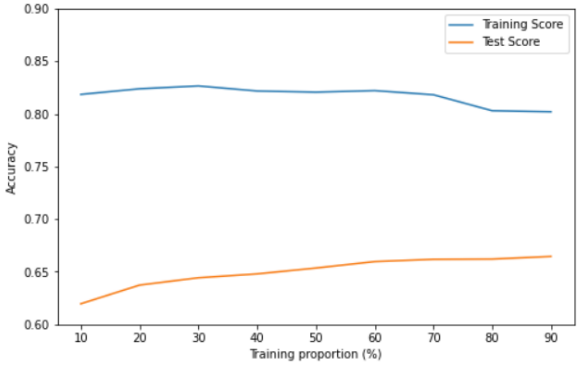


Figure 4: Learning curves for (i) Multinomial NB (ii) Linear SVM and (iii) Logistic Regression

The optimal training-test split for both Multinomial NB and Logistic Regression is 70-30, with the dip in training accuracy beyond 70% training proportion indicating overfitting. Linear SVM indicates the optimal training-test split is closer to 80-20, however the majority optimal split of 70-30 was selected.

3. Results

3.1 Hyperparameter Tuning

Multinomial NB

The smoothing hyperparameter α represents the pseudo-count added to each conditional probability in order to prevent zero probabilities. It is necessary in this dataset since there is a high probability of encountering words during testing that were not present in the training dataset. The effect of varying α can be observed in Table 5.

Table 5: Comparison of Multinomial NB accuracy and F-score with different values of the smoothing parameter α

Value	Accuracy	F-Score
$\boldsymbol{\alpha = 0.01}$	0.6107	0.5193
$\boldsymbol{\alpha = 0.05}$	0.6285	0.5286
$\boldsymbol{\alpha = 0.1}$	0.6263	0.5193
$\boldsymbol{\alpha = 0.5}$	0.6163	0.3654
$\boldsymbol{\alpha = 1.0}$	0.5969	0.2943
$\boldsymbol{\alpha = 5.0}$	0.5835	0.2469

For our dataset the optimal value is 0.05, producing the highest accuracy (0.6285) and F-score (0.5286).

Linear SVM

The value of the regularisation parameter C in Linear SVM informs the model of the importance of misclassifying a training example. Large values of C result in a smaller-margin hyperplane in order to achieve the maximum number of correctly classified instances. Conversely, a smaller C will result in a larger-margin hyperplane even if some points a misclassified. The second option allows for Linear SVM to be applicable in circumstances where the data is not strictly linearly separable. Another hyperparameter for Linear SVM in the case of multiclass classification is the multiclass strategy. Each strategy splits a multi-class classification into binary classification problems, one per pair of classes for 'One vs One' (OVO) and one per class for 'One vs Rest' (OVR).

Table 6: Comparison of Linear SVM accuracy and F-score with different values of the regularisation parameter C and the multiclass strategy

Value	Accuracy	F-Score
$\boldsymbol{C = 0.1}$	0.6511	0.5128
$\boldsymbol{C = 0.5}$	0.6575	0.5788
$\boldsymbol{C = 1.0}$	0.6471	0.5797
$\boldsymbol{C = 5.0}$	0.6161	0.5619
$\boldsymbol{C = 10.0}$	0.6079	0.5566
$\boldsymbol{C = 50.0}$	0.5923	0.5425

Multiclass	Accuracy	F-Score
OVO	0.6563	0.5888
OVR	0.6471	0.5797

As shown by Table 6. the optimal value for C is 0.5 since it produces the highest accuracy (0.6575) and an optimal F-score (close to the highest value). The optimal multiclass strategy is OVO (the default being OVR) as it has a higher accuracy and F-score.

Logistic Regression

The regularisation hyperparameter C in Logistic Regression has a similar role to the regularisation hyperparameter in Linear SVM. Both hyperparameters are used to avoid over-fitting in the data and therefore impact on the ability of a model to generalise on new unseen instances. Larger values of C in Logistic Regression give more weight to outlier points/errors and therefore tend towards overfitting the model (similar in Linear SVM).

Table 7: Comparison of Logistic Regression accuracy and F-score with different values of the regularisation hyperparameter C

Value	Accuracy	F-Score
$\boldsymbol{C = 0.1}$	0.6160	0.3612
$\boldsymbol{C = 0.5}$	0.6507	0.5140
$\boldsymbol{C = 1.0}$	0.6620	0.5558
$\boldsymbol{C = 5.0}$	0.6533	0.5835
$\boldsymbol{C = 10.0}$	0.6468	0.5821
$\boldsymbol{C = 50.0}$	0.6312	0.5743

Table 7. demonstrates the optimal value for C is given by the default value 1.0, as although it doesn't produce the highest F-score (obtained when C=5.0) it has the highest accuracy (0.6620). Furthermore, higher values of C can often lead to overfitting of the model.

Stacking Model

One of the parameters for the Stacking Classifier is the choice of metaclassifier that combines the base classifiers. The default classifier is Logistic Regression however Decision Tree is also a common choice.

Table 8: Comparison of Stacking accuracy and F-score with different types of metaclassifier

Metaclassifier	Accuracy	F-Score
Logistic Regression	0.6604	0.5861
Decision Tree	0.5456	0.4969

In the case of this dataset, it is clear the better choice of metaclassifier is Logistic Regression (as seen in Table 8.).

3.2 Final Classification Over Training Set

Table 9: Accuracy and F-score of each classifier under optimised training-test split, feature reduction and hyperparameter tuning

Model	Accuracy	F-Score
Zero-R	0.1738	0.0987
Multinomial NB	0.5621	0.5390
Linear SVM	0.6340	0.6040
Logistic Regression	0.6311	0.6078
Stacked Model	0.6242	0.5761

Confusion Matrices

Multinomial NB:

	PRED: POSITIVE	PRED: NEGATIVE	PRED: NEUTRAL
ACTUAL: POSITIVE	991	95	503
ACTUAL: NEGATIVE	131	605	401
ACTUAL: NEUTRAL	1005	729	2081

Linear SVM:

	PRED: POSITIVE	PRED: NEGATIVE	PRED: NEUTRAL
ACTUAL: POSITIVE	1011	62	516
ACTUAL: NEGATIVE	86	649	402
ACTUAL: NEUTRAL	739	589	2487

Logistic Regression:

	PRED: POSITIVE	PRED: NEGATIVE	PRED: NEUTRAL
ACTUAL: POSITIVE	1063	80	446
ACTUAL: NEGATIVE	97	703	337
ACTUAL: NEUTRAL	780	673	2362

Stacked:

	PRED: POSITIVE	PRED: NEGATIVE	PRED: NEUTRAL
ACTUAL: POSITIVE	898	45	646
ACTUAL: NEGATIVE	63	520	554
ACTUAL: NEUTRAL	671	479	2665

4. Critical Analysis

The results from Table 9. illustrate that the worst performing classifier in terms of both accuracy and F-score (aside from the baseline Zero-R) was Multinomial NB. An assumption of this classifier is that all features conditional on the class are independent. Since the dataset involves linguistic features (i.e., features extracted from a natural language input), they cannot be considered independent as words appearing together in texts share multiple syntactic and semantic dependencies. Multinomial NB is known to still produce reasonably accurate results even if the assumption of conditional independence is violated but is outperformed by Linear SVM and Logistic Regression in this case.

Linear SVM is more robust in high-dimensional spaces compared to due to its exclusive use of support vectors to form decision boundaries. However, it can have subpar performance when the dataset is contains excessive noise (i.e., the target classes are overlapping) due to its effect on the hyperplane. It is possible the colloquial and non-consistent language often found on Twitter may be introducing excessive noise in the dataset thus impacting the performance.

In practice, Logistic Regression and Linear SVM are comparatively similar as demonstrated by their similarity in accuracies (0.6311 and 0.6340 respectively). A disadvantage of Logistic Regression is the assumption of linearly separable data, in which this particular dataset has not been shown.

The accuracy and F-score of the Stacking Method was lower than both Linear SVM and Logistic Regression potentially due to the choice of base models. Linear SVM and Logistic Regression have similar performance thus making similar errors, and Multinomial NB's poorer performance only introduces noise in the final classification. Since the performance of Stacking is optimised when the base classifiers are independent, the classifiers' similarities and overlap could be decreasing the accuracy of our Stacking model. Although the Logistic Regression metaclassifier is not hugely impacted by outliers due to being tapered by the Logistic function, the small number of base classifiers means more weight is given to classifiers that could potentially introduce noise for example Multinomial NB.

5. Conclusion

Although the classifier achieving the highest accuracy was Linear SVM, the highest F-score (0.6078) was achieved by Logistic Regression. Therefore, Logistic Regression may be the best method used for datasets where class distribution is more even.

Sentiment classification is a process with many factors in play from feature selection to classification method. There is room for further investigation to account for underlying assumptions made by the chosen models and dataset, for example conducting Non-Linear SVM using appropriate kernel functions; use of Twitter-specific language statistics as features alongside TF-IDF e.g., number of hashtags in a Tweet, number of username mentions etc.; or the use of emojis.

6. References

- Gamallo, P., & Garcia, M. (2014). Citius: A Naive-Bayes Strategy for Sentiment Analysis on English Tweets. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 171–175. https://doi.org/10.3115/v1/s14-2026
- Ahmad, M., Aftab, S., Muhammad, S. S., & Awan, S. (2017). Machine Learning Techniques for Sentiment Analysis: a Review. *Internation Journal of Multidisciplinary Sciences and Engineering*, *8*(3), 2045–7057.
- Haddi, E., Liu, X., & Shi, Y. (2013). The Role of Text Pre-processing in Sentiment Analysis. *Procedia Computer Science*, *17*, 26–32. https://doi.org/10.1016/j.procs.2013.05.005
- Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, *3*(2), 143–157. https://doi.org/10.1016/j.joi.2009.01.003
- Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *Processing*, *150*.
- da Silva, N. F. F., Hruschka, E. R., & Hruschka, E. R. (2014). Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, *66*, 170–179. https://doi.org/10.1016/j.dss.2014.07.003
- Rosenthal, S. N. (2017). SemEval-2017 Task4: Sentiment Analysis in Twitter. Proceedings of the 11th International Workshop on Semantic Evaluation . Vancouver, Canada: SemEval '17.
- Zheng, Z., Cai, Y., Li, Y., Zheng, Z., Cai, Y., & Li, Y. (2015). OVERSAMPLING METHOD FOR IMBALANCED CLASSIFICATION. *Computing and Informatics*, *34*, 1017–1037.