COMP90049 Project2 Report: Sentiment Analysis of Tweets based on Machine Learning

Ye Yang

1. Introduction

With the rise of social networks in recent years, various social networking sites such as Twitter, Instagram and Facebook have become a popular channel for people to express their sentiments and opinions publicly. Obviously, the emergence of such real-time online social media like Twitter has spawned a new effective strategy for public opinions gathering.

A great deal of research about sentiment analysis of online social media based on machine learning technologies has been conducted so far. Go, Bhayani and Huang (2009) propose a distant supervised approach based on three machine learning algorithms (Naïve Bayes, Maximum Entropy and SVM) for sentiment analysis on tweets with emoticons, and the result shows that the classification accuracy can achieve above 80%. Calderón, Álvarez and Mariño (2019) discuss the implementation, scalability and limitation of parallelized machine learning approaches for sentiment prediction based on tweets under a distributed environment.

In this report, three typical machine learning algorithms for classification, including Naïve Bayes, Multinomial Logistic Regression and Random Forest, are implemented based on WEKA, a free suite of machine learning software, to build a sentiment classifier for tweets.

2. Dataset and Data Pre-processing

The tweets dataset established by Rosenthal, Farra and Nakov (2017) contains a training set, an evaluation set and a test set (see Table 1).

Dataset	Number of Tweets	
training	22987 instances with label (47 attributes)	negative (5062)
		positive (6471)
		neutral (11454)
evaluation	4926 instances with label (47 attributes)	negative (1038)
		positive (1488)
		neutral (2400)
test	4926 instances without label (47 attributes)	negative (?)
		positive (?)
		neutral (?)

Table 1: The Dataset for Sentiment Analysis

In order to maximize the utilization of data for a better performance, a combination of training set and evaluation set (22987+4926=27913 instances in total) is applied based on 10-folds cross validation in the process of modelling. There are 47 attributes in total including the id, some linguistic vocabularies and the sentiment. The attribute id is removed since it is irrelevant to the performance of classification. In order to build the machine learning models in WEKA, the CSV dataset files are transformed into ARFF files, and the attribute of sentiment in test set is set as '?' for prediction. The predicted results are multiclass with positive, negative or neutral.

3. Hypothesis

One target of this report is to verify whether the tweet text can be used to help us to identify the sentiment of Twitter users. In general, people tend to express different emotional tendencies on different topics. Based on such premise, this report assumes that the predicted sentiment labels would show a correlation to the specific topics. For example, in view of the fact that people used to criticize politicians, the predicted sentiment result might be dominated by negative class for some political topics like 'Trump'.

4. Methodology

Three machine learning classifiers are generated based on Naïve Bayes, Multinomial Logistic Regression and Random Forest. All the relevant parameters of the algorithms remain the default values in WEKA. The WEKA knowledge flow is shown in Figure 1.

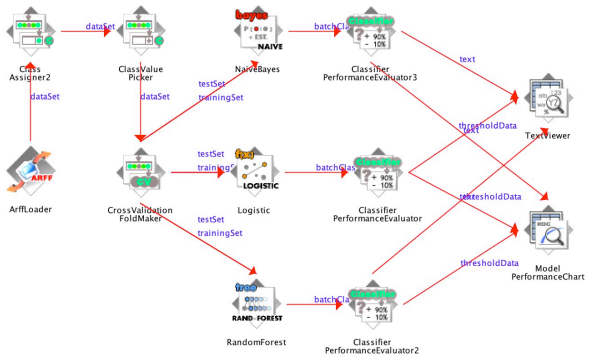

Figure 1: WEKA Knowledge Flow Diagram

4.1. Naïve Bayes

Naïve Bayes is a classification algorithm which is categorised as a simple probabilistic classifier based on Bayes' Theorem with assumption that attributes are conditionally independent (Lewis, 1998). Suppose that there is a training set with n dimensional attribute vector $X = (x_1, x_2, \ldots, x_n)$ and K classes labels $C_1, C_2, \ldots, C_K$. For any given X, the probability that it belongs to class C_i is:

$$P(C_i|X) = \frac{P(X|C_i) \cdot P(C_i)}{P(X)} \propto P(C_i) \cdot \prod_{k=1}^{n} P(x_k|C_i)$$

The instance with attribute vector X is predicted to be classified as C_i if and only if $P(C_i|X)$ achieves the maximum among all the K classes.

4.2. Multinomial Logistic Regression

Multinomial Logistic Regression is a machine learning algorithm for multiclass problems which is generalized from the classic binomial logistic regression (Greene, 2012). It uses a linear predictor function $f(k, i)$ to predict the probability that observation i has outcome k:

$$f(k, i) = \beta_k \cdot X_i$$

where β_k is the set of regression coefficients associated with outcome k, and X_i (a row vector) is the set of explanatory variables associated with observation i.

The probability equation is:

$$P(Y_i = c) = softmax(c, \beta_1 \cdot X_i, \beta_2 \cdot X_i, \ldots, \beta_K \cdot X_i) = \frac{e^{\beta_c \cdot X_i}}{\sum_{k=1}^{K} e^{\beta_k \cdot X_i}}$$

where Y_i is the predicted class of observation i has outcome k and the *softmax* function here is similar to the *sigmoid* function in classic binomial logistic regression.

The instance will be classified as class c when the corresponding probability reach the maximum over the whole class set.

4.3. Random Forest

Random Forest is an extended variant of Decision Tree based on Bagging (one of the most famous parallel ensemble learning methods), which introduces the random selection of attributes in the training process (Breiman, 2001).

To be specific, when selecting a splitting attribute, the traditional Decision Tree chooses an optimal attribute in the attribute set of the current node (assuming there are d attributes), while in the Random Forest, for each node of the base decision tree, a subset of k attributes is first selected randomly from the current node attribute set and then the best attribute is chosen from the subset used for division. The parameter k here controls the degree of introduction of randomness. If $k = d$, then the construction of the base decision tree is the same as that of the traditional decision tree, and if $k = 1$, then a random attribute is selected for partition. Normally, the recommended value of parameter $k = \log_2 d$.

5. Evaluation Metrics

In this report, some typical evaluation indicators for machine learning models, including Accuracy, Precision, Recall, F1-score, Confusion Matrix and ROC curves, are introduced for system evaluation and result analysis.

5.1. Accuracy, Precision, Recall and F1-score

In the multi-classification, for any specific class i, giving the following definitions:

TP(*class i*): the number of actual class i instances that are also predicted as class i.

FP(*class i*): the number of instances predicted as class i which are not in fact.

TN(*class i*): the number of instances predicted as other classes instead of i which also not belong to class i actually.

FN(*class i*): the number of instances predicted as other classes instead of i which are class i actually.

The corresponding Precision, Recall and F1-score can be calculated as below:

$$Precision(class\ i) = \frac{TP(class\ i)}{TP(class\ i) + FP(class\ i)}$$

$$Recall(class\ i) = \frac{TP(class\ i)}{TP(class\ i) + FN(class\ i)}$$

$$F1 - score(class\ i) = \frac{2Recall(class\ i) \cdot Precision(class\ i)}{Recall(class\ i) + Presicion(class\ i)}$$

The overall Precision, Recall and F1-score of the whole model can be calculated as the weighted average values of the corresponding results for each class. The overall Accuracy denotes the percentage of total number of correctly classified instances among the total instances over the test set, and it can be simply calculated as:

$$Accuracy(overall) = \frac{\sum TP(class\ i)}{total\ instances\ in\ the\ test\ set}$$

5.2. Confusion Matrix and ROC Curves

The Confusion Matrix (also known as error matrix) is often used to visually evaluate the performance of supervised learning algorithms. In this project, the Confusion Matrix of sentiment classification is shown as below:

	negative (predicted)	positive (predicted)	neutral (predicted)
negative (actual)			
positive (actual)			
neutral (actual)			

Table 2: The Sample of Confusion Matrix

ROC (Receiver Operating Characteristic) curve and AUC (Area Under Curve) are widely used in binomial classification evaluation, especially for the case that the classes are not well balanced. In the ROC curve, the horizontal axis represents the FPR (False Positive Rate) = $FP/(FP+TN)$, and the vertical axis reflects the TPR (True Positive Rate), which equals $TP/(TP+FN)$.

In general, if the ROC curve of one model is completely wrapped by the curve of another model, it can be asserted that the performance of the latter is better than the former. If the ROC curves of the two models cross each other, the more reasonable criterion is to compare the AUC and the model with larger AUC can be considered better than the other one. Since the sentiment analysis is a multi-classification problem, this report will evaluate the ROC and AUC of the three machine learning models by class.

6. Result Analysis

6.1. Accuracy, Precision, Recall and F1-score

The overall and detailed results of Accuracy, Precision, Recall and F1-score are shown in Figure 2 and Table 3 respectively.

Figure 2: Overall Results of Three Models

ML Models	Class	Precision	Recall	F1-score
Naïve Bayes **Accuracy:** 53.2655%	negative	0.421	0.255	0.317
	positive	0.525	0.387	0.445
	neutral	0.558	0.739	0.636
	average	0.518	0.533	0.512
Multinomial Logistic Regression **Accuracy:** 55.6157%	negative	0.581	0.169	0.262
	positive	0.623	0.273	0.380
	neutral	0.544	0.889	0.675
	average	0.575	0.556	0.501
Random Forest **Accuracy:** 53.3837%	negative	0.448	0.184	0.261
	positive	0.537	0.299	0.384
	neutral	0.543	0.823	0.655
	average	0.521	0.534	0.491

Table 3: The Detailed Results by Class

From Figure 2, it can be seen that the Multinomial Logistic Regression model performs best in terms of overall Accuracy (55.6157%), Precision (0.575) and Recall (0.556), and second only to Naïve Bayes model on F1-score.

6.2. Confusion Matrix and ROC Curves

The Confusion Matrix for three machine models are shown in Table 4.

Naïve Bayes	negative (predicted)	positive (predicted)	neutral (predicted)
negative (actual)	1553	737	3810
positive (actual)	575	3078	4306
neutral (actual)	1564	2053	10237
Multinomial LR	negative (predicted)	positive (predicted)	neutral (predicted)
negative (actual)	1032	323	4745
positive (actual)	201	2176	5582
neutral (actual)	543	995	12316
Random Forest	negative (predicted)	positive (predicted)	neutral (predicted)
negative (actual)	1123	571	4406
positive (actual)	407	2379	5173
neutral (actual)	978	1477	11399

Table 4: The Results of Confusion Metrix

As can be seen from the Confusion Matrix, all the three models tend to classify most of the negative and positive classes as neutral. The potential reason may be that the neutral class dominates the training set and the attributes selected may have a certain tendency.

As for the ROC results, since it is inconvenient to view and compare them in WEKA, the result files

are transformed into CSV format and processed in Python. The results are shown below:

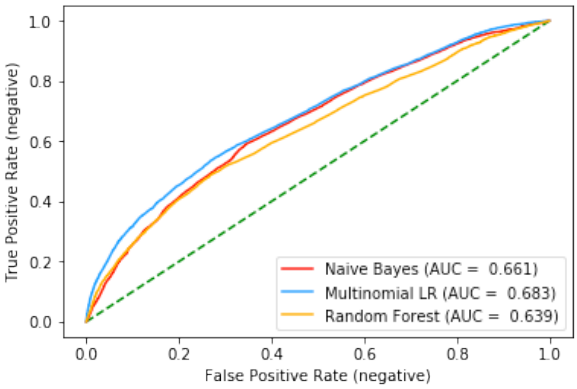

Figure 3: ROC and AUC of Negative Class

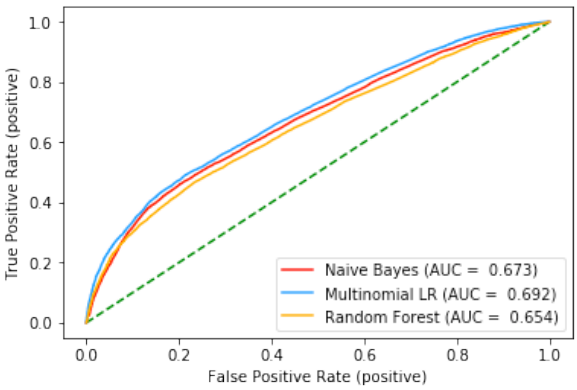

Figure 4: ROC and AUC of Positive Class

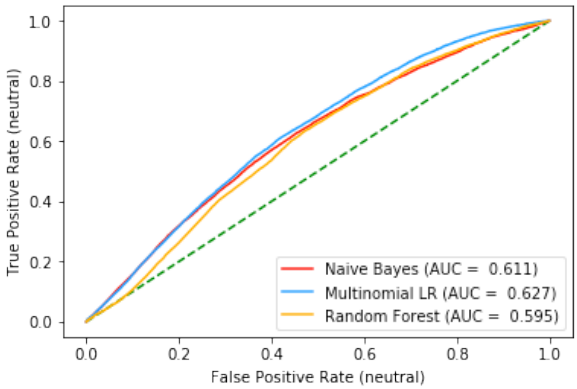

Figure 5: ROC and AUC of Neutral Class

It can be seen that, for all the three sentimental classes, Multinomial Logistic Regression achieves the best performance on ROC curves and AUC.

6.3. Hypothesis Verification

As mentioned before, here five topics are selected as examples to verify whether the tweets text can help to identify the sentiment of users. The results based on Multinomial Logistic Regression model are shown in Table 5.

Topic	Class	Training Set	Test Set
Trump train set: 747 test set: 136	negative	368 (49.26%)	46 (33.82%)
	positive	44 (5.89%)	0 (0.00%)
	neutral	335 (44.85%)	90 (66.18%)
Happy train set: 387 test set: 63	negative	28 (7.24%)	1 (1.59%)
	positive	321 (82.95%)	49 (77.78%)
	neutral	38 (9.82%)	13 (20.63%)
Love train set: 514 test set: 95	negative	28 (5.45%)	3 (3.16%)
	positive	398 (77.43%)	88 (92.63%)
	neutral	88 (17.12%)	4 (4.21%)
Hate train set: 129 test set: 18	negative	85 (65.89%)	15 (83.33%)
	positive	10 (7.75%)	2 (11.11%)
	neutral	34 (26.36%)	1 (5.56%)
News train set: 252 test set: 49	negative	58 (23.02%)	3 (6.12%)
	positive	52 (20.63%)	0 (0.00%)
	neutral	142 (56.35%)	46 (93.88%)

Table 5: Topic Results of Multinomial LR

It can be seen that the test results are basically in line with our expectations. In terms of the topic 'Trump', the rate of predicted negative class is less than our expectation. The potential reasons could be various, such as the limitation of given attributes and the limited size of training set. In addition, some actual positive or negative tweets are classified as neutral since the sentiment expressed through emoji is ignored in this project. For example, the following two tweets classified as neutral should be positive in fact:

799314934630912000 | neutral happy nationalfastfoodday

801938487242944000 | neutral love is all you need to be thankfulfor! screamqueens thanksgiving

7. Conclusions

Overall, this report implements three machine learning models for tweets sentiment analysis and Multinomial Logistic Regression achieves the best performance on most of the evaluation indexes. Based on the result of topic analysis, it can be concluded that the tweets text can be used to help us to identify the sentiment of users. For a further improvement of the performance of the model, a training set with larger size and higher attribute quality based on some advanced feature engineering technologies such as word embedding would be a worthwhile attempt in the future.

8. References

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Calderón, C. A., Álvarez, M., & Mariño, M. V. (2019). Distributed Supervised Sentiment Analysis of Tweets: Integrating Machine Learning and Streaming Analytics for Big Data Challenges in Communication and Audience Research. Empiria: Revista de metodología de ciencias sociales, (42), 113-136.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1(12), 2009.

Greene, W. H. (2012). Econometric analysis (7th ed.) (pp. 803-806). Boston: Pearson Education.

Lewis, D. D. (1998, April). Naive (Bayes) at forty: The independence assumption in information retrieval. In European conference on machine learning (pp. 4-15). Springer, Berlin, Heidelberg.

Rosenthal, S., Farra, N., & Nakov, P. (2017, August). SemEval-2017 task 4: Sentiment analysis in Twitter. In Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017) (pp. 502-518).