COMP30027 Project2 Report

1. Introduction

Twitter is an online microblogging service where users broadcast short posts known as tweets. These tweets sometimes show the emotion and attitudes of the users. Therefore, capturing and predicting the tweet sentiment is essential in some fields, such as analyzing personal preferences and personalities. In this report, we mainly focusing on building and critically analyzing supervised Machine Learning methods, which can perform better in extracting and predicting sentiments based on the tweet in English. To facilitate the training of the model, the labeled training set of 21802 Tweets has been classified as "positive", "negative" and "neutral". The instances consist of user id, tweet text and sentiments. As for the model performance testing, the raw unlabeled testing dataset is provided with 6099 instances, which has the same structure as the training dataset, except for labels. The real predication accuracy can be shown on Kaggle.

2. Method

2.1 Text pre-processing

We aim to choose the qualified words which could be used in feature selections to reduce the noises and unmeaningful feature dimensions. After applying pre-processing, the feature dimension was reduced from 44045 to 21746. It indicates that text pre-processing improves the efficiency of feature selection.

Part of the pre-processing which is different to the common text-processing skills will be illustrated.

2.1.1 Remove username and URL links

We remove the username and URL links by taking advantage of following the unique properties of the Twitter language model (Go, 2009).

2.1.2 Expand contractions and remove repeated letters

Twitter language is relatively causal (Go, 2009), so the contractions and repeated letters will make noises and create more features in building models.

As for the repeated letters in a word, we treat any letter that occurs more than two times in a row as two occurrences which is unusual for a word to have three closed repeated letters, such as "huuuungry", "goood" (Go, 2009).

2.1.3 Remove stop words

In our process, all the stop words with non-negative meanings have been removed, because the negative words, such as "not", "no", "never", have high correlation with negative sentiment.

2.1.4 Lemmatize

We need to convert the words with the same meaning to its root to create a qualified general feature, such as "running", "ran". They need to be converted to "run".

In this case, Lemmatization is better than stemming. When lemmatizing a word, the word remains its original meaning.

2.1.5 Remove retweets and repeated tweets

We remove the tweet with "RT.", because the retweets have the same meaning. They will add weights on these words in training models.

The tweet which has been shown in the latest 100 tweets also needs to be removed (Go, 2009), same reason as removing retweets.

2.2 Feature selection

We aim to find valuable and meaningful features for training and testing to increase the model performance**.**

2.2.1 Data split

In order to test the performance of each model, we use K-fold Cross Validation to split training set into sub training set and sub testing set. Taking the time complexity into consideration, we determine to use 4-fold, because both SVM and Random Forest are time consuming when

the dataset and feature dimensions are large.

Moreover, we apply under sample strategy to make both the sub training set and sub testing set balanced distributed.

2.2.2 TFIDF

As for the feature vectorizing, tf-idf considers the term ordering and harness of the term, which is better than BoW.

Additionally, it is unnecessary for us to change the "ngram" in tfidfVectorizer(). Although some two-word phrases are more meaningful than one word, most of the two-word phrases include a stop word, such as "like to", "love to", "not like", and not love". In this case, only the sentiment stop word, which has not been removed by us is useful.

2.2.3 Remove low variance features

Some of the features rarely appear but have strong correlation with one specific label. And some of them are not meaningful enough, which have almost the same appearance frequency for each instance.

However, the feature selection method, MI and chi-square are biased toward rare, uninformative features.

Therefore, we remove the words which appear less than five times in each sub training dataset.

2.2.4 Select features with K best and chi-square

After the previous processes, the features are numeric, but the labels are discrete. It points that chi-square is more suitable than MI. After checking the correlation of features and labels, we choose the best K features for each model. It is a further way to declaring more qualified features.

2.3 Models

In order to find the best model for our dataset, we test different classifiers: Zero-R, support vector machines, and random forest.

2.3.1 Baseline

We make the most frequent label dominates the predicting result. Based on this model, tweets will be classified to the most frequent label. In this case, all the previous efforts are unnecessary, and no machine learning models have been implemented. This model provides an accuracy benchmark for other models.

2.3.2 Select hyper-parameters

To determine the hyper-parameter for a model is important, we use halving grid search instead of grid search to save time. The 'HalvingGridSearchCV' is able to find parameter combinations that are just accurate as 'GridSearchCV', in much less time (Comparison between grid search and successive halving, n.d.
).

2.3.3 Supported Vector Machines

SVM still remains high efficiency and stable accuracy when there is a large amount of features. The more features, the easier the dataset satisfies linear separable.
In our SVM model, we choose "rbf" as the kernel function, "1" as the value of C, and "ovr" as the decision function shape.

2.3.4 Random Forest

Random Forest has a stable performance in getting high accuracy in general, and it can handle large datasets efficiently (Great Learning Team, 2020). Besides, it could be time saving. RF extends the advantages of Bootstrap and Bagging, each of the random tree is training in parallel. Moreover, it prevents the overfitting, because a subset of features is selected randomly each time.

In our Random Forest model, we choose "balanced" as the class weight, "gini" as the criterion, and "log2" as the max features.

2.4 Evaluations

We use accuracy score, precision score, recall score and F1 score to evaluate the models.

As for multi-classification problem, macro or weighted averaging will be used as evaluation metrics. Based on the balanced data sampling method, macro and weighted averaging may has the same values.

3. Result

In this aspect, we illustrate the specific performance of SVM and Random Forest models with different selection of K.

3.1 SVM results

The testing accuracy of two model is the average accuracy of the 4 sub testing sets. The real testing accuracy is the accuracy of predicting the original test set.

We choose the best 2000 features in the feature lake, because our SVM model's accuracy shows a local maximum at this point with range of [0, 2499]. Based on the average testing result, the model is not overfitted, and the test accuracy is higher than 0.5806 (Table 1), which is the benchmark accuracy provided by baseline model.

accuracy	Training accuracy	Testing accuracy	Real testing accuracy
average	0.913503	0.625117	0.62525

Table 1- the average training accuracy and testing accuracy

Our SVM model performs better in predicting correct sentiment when given sentiment is the same (Table 2).

Evaluation metrics	Precision score	Recall score	F1 score
macro	0.621538	0.625117	0.622871
weighted	0.621538	0.625117	0.622871

Table 2- the weighted and macro averaging of svm model

3.2 Random Forest result

The accuracy of our Random Forest also has an improvement from the baseline model. We select 2499 features in each sub training set, which is the total number of features of the smallest sub training sets (Table 3).

accuracy	Training accuracy	Testing accuracy	Real testing accuracy
average	0.999644	0.614906	0.61500

Table 3- the training and testing accuracy of random forest model

Our RF model also performs better in predicting correct sentiment when given sentiment is the same (Table 4).

Evaluation metrics	Precision score	Recall score	F1 score
macro	0.610842	0.614906	0.612052
weighted	0.610842	0.614906	0.612052

Table 4- the weighted and macro averaging of random forest model

3.3 illustrative example entail

Examples:

- hahahahahah so much of this #westworld season was me screaming expletives at this idiot https://t.co/mcek4okziz (Example 1)
- " rumours that chris harris may join chris evans on top gear. they'll have to make extra wide doors for 2 such fantastically egotistical twats " (Example 2)

	SVM	RF
Example1	positive	negative
Example2	neutral	positive

Table 5- the partial prediction result

The main reason might be the different selection of K. When increasing the K, random forest model has a general increasement on testing accuracy. However it might take some unvaluable words, like "idiot" or "fantastically", which could perform different sentiment in different situation.

4. Critical analysis

4.1 Data distribution and data balancing

Different data distribution will lead to different model performance. The accuracy of baseline model shows that the training dataset is unevenly distributed.

Based on the label distribution (Figure 1) and

confusion matrix of SVM (Figure 2), which initially uses the original dataset. The "neutral" labels dominate the whole distribution.

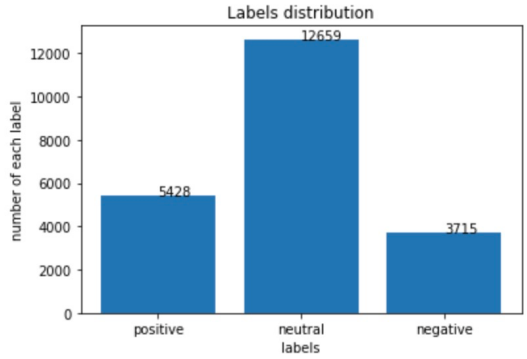

Figure **1-** the distribution of three labels

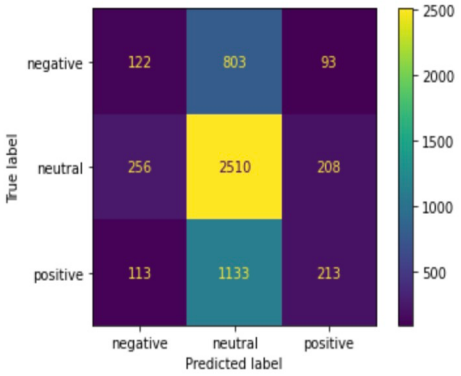

Figure **2-** confusion matrix of SVM model based on one of the unbalanced sub training folds

In this case, the unevenly distributed data leads to a biased predicting performance of SVM model (Table 6). It predicts more "neutral" labels. However, the Zero-R accuracy value in Kaggle shows that the given test dataset distributed evenly. Therefore, although the sub testing accuracy is high, the real testing result performs oppositely.

Training accuracy	Testing accuracy	Real testing accuracy
0.861526	0.653748	0.51168

Table 6- the accuracy of SVM model of using the original dataset, with best 2000 features

Under this situation, we randomly under sampled the original training data set, made it distributed evenly based on the number of negative labels (Figure 3). The real testing accuracy performs much better (Table 7).

Training accuracy	Testing accuracy	Real testing accuracy
0.913503	0.625117	0.62525

Table 7- the accuracy of SVM model of using the balanced dataset, with best 2000 features

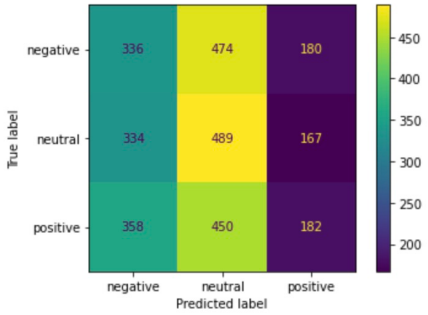

Figure **3-** confusion matrix of SVM model based on one of the balanced sub training folds

4.2 Feature selection analysis

Variance threshold makes the range of features to be selected. We choose less than 5 appearances as threshold to make the range reasonable, which is [0,2499]. About the best K, 2000 is most suitable for SVM model, because after the preprocessing and variance checking, most of the features are valuable to some extent. There is a tradeoff between the accuracy and the training dimensions.

Different feature group performance:

Variance threshold	K best	SVM Real accuracy
7.859e-05	2000	0.62525
7.859e-05	2499	0.62279
0.0001855	600	0.61541
0.0001855	500	0.61213

Table 8- the accuracy of SVM model of using the different features

It is unrealistic to try all feature combinations, we have tried some

remarkable different combinations of

variance threshold and K. Therefore, features currently selected by using variance threshold and K best only proves that the feature is relatively the best in our experimental combinations.

4.3 Model difference analysis

According to the model's accuracy and evaluation matrices, it is obvious that SVM performs slightly better than Random Forest by using selected features.

Generally, the random forest has greater ability when the dataset contains outliers. However, after the data cleaning and feature selection, we have a certain confidence that our data is reasonably clean and outlier free. The structural risk is minimized, which made less difference between two models.

Moreover, due to the reduction of feature dimensions, SVM could change the value of "C" to gain greater ability of generalization based on training with limited feature sizes. The ratio of feature dimension and sub training set size if about 0.4, which is relatively high, SVM is effective in cases where the number of dimensions is closed or greater than the number of instances.

However, we found there is a positive correlation between feature dimension and accuracy of random forest model. Due to the trade-off between running time and accuracy, the feature dimension might limit the performance of random forest.

5. Conclusions

Overall, SVM model preforms better than Random Forest model in our selected range of features when predicting tweet sentiment in a large training dataset. Besides, other aspects also influence the prediction. The text pre-processing improved the efficiency and effectiveness of feature selection. The feature selection improved the training result of model and the prediction accuracy. However, the features selection needs to be adjusted according to the model characteristic. Moreover, distribution of the training dataset has a significant effect on model performance, a balanced training set or the training set with same distribution as test dataset will improve the predicting accuracy.

6. References

Go, A. (2009). Sentiment Classification using Distant Supervision. CS224N project report, Stanford.

Rosenthal, Sara, Noura Farra, and Preslav Nakov (2017). SemEval-2017 Task 4: sentiment analysis in Twitter. In Proceedings of the 11th International Workshop on semantic evaluation (SemEval '17). Vancouver, Canada.

Scikit-learn. n.d. *Comparison between grid search and successive halving*. [online] Available at: <https://scikit-learn.org/stable/auto_examples/model_selection/plot_successive_halving_heatmap.html> [Accessed 5 May 2022].

Great Learning Team, 2020. *Random forest Algotithem in Machine learning: An Overview*. [online] mygreatlearning. Available at: <https://www.mygreatlearning.com/blog/random-forest-algorithm/> [Accessed 10 May 2022].