COMP30027 Assignment 2 Report

Anonymous

1. Introduction

With the advancing technology and easier access to the internet, people are free to express their opinions through blogs, social media and websites. Twitter is one of many social media platforms, generating a large number of sentimental messages which can then provide business to monitor the needs of the customers, advertise the right product, and observe the public views toward their companies (Go, 2019). It also benefits consumers to do any relevant research before purchasing products.

The aim of this project is to build supervised Machine Learning methods and critically analyse them to automatically predict the sentiment of given Tweets. The sentiments are classified as either positive, negative and neutral.

2. Feature Engineering

The raw text is transformed into a more appropriate presentation of features, so that we can train the data to build our final model. Pre-processing stages such as text cleansing, lemmatisation and train-test split are implemented to optimise the data set.

2.1 Raw Data

We are provided with a labelled training set and unlabelled test set. Each instance in a train set contains the ID of the Tweet, test and the sentiment of the Tweet. It contains 12659 neutral instances, 5428 positives, and 3715 negative Tweets. The test set, on the other hand, does not have a respective sentiment to the text.

2.2 Pre-processing

1. All punctuations and emojis are removed from each Tweet text because punctuations are not relevant to the sentiment of the text and there are no emojis presented in the given set, but to get rid of possible noisy labels and to reduce the size of the dataset, they are neglected.
2. All upper-case words are converted to lowercase.
3. Tweets contain many words that are not significant to the classification of labels, for instance. To minimise the time processing unnecessary data, we remove them.
4. Verbs and nouns are lemmatised to accurately classify each word in the text. For example, the first instance of the train set is “doctors hit campaign trail as race…” and is lemmatised to “doctor hit campaign train as race…” which reduces the number of tokens while delivering the same message.

2.3 Train-test Split

The given train data is a large set with 21802 instances and the test set is unlabelled which makes it difficult to evaluate our model at the end. So, we randomly split the train set into 90:10 holdout to use the 10% of the instances as the test set to evaluate the unseen data.

2.4 TFIDF

TF-IDF (Term Frequency Inverse Document Frequency) is used to vectorise over 20000 texts with appropriate parameters such as applying stop_words, min_df, max_df and ngram_range.

Stop_words = “english” removes words like *I, we, the, etc* that do not contribute to the sentiments of the text. They also increase the complexity of the model.

Max_df and min_df are used for removing the least or most frequent words, so that we

only consider more meaningful terms in analysing the sentiment of the text. We set min_df = 5 and max_df = 0.8 to ignore all terms that appear in less than 5 instances and in more than 80% of the instances.

The ngram_range was set to (1,3) to consider unigram, bigram and trigram to identify multiword expressions or certain phrases that occur in the text.

3. Predictive Models

We decided to implement various predictive models to compare the accuracy and select the best fit model. Multinomial Naïve Bayes is used as a baseline and other classifiers such as Linear SVM, Logistic Regression, K-Nearest Neighbours and Stacking are applied and observed to increase the accuracy of our prediction model.

3.1 Multinominal Naïve Bayes

Naïve Bayes classifier is one of the most commonly used supervised machine learning algorithms. The Naïve Bayes algorithm assumes that there is no interdependence between variables and if there are matching variables, it will classify as a particular feature. The advantages of using the Naïve Bayes algorithm are that the train data can be relatively smaller than other algorithms, and mislabelled data can be used as training data as well. In addition, the Naïve Bayes classifier is less time-consuming than logistic regression and decision tree, so it is a very efficient algorithm. Specifically, multinominal Naïve Bayes can be applied to classifying documents on the basis of probabilistic points of view and it can admit the frequency of the particular word in the document filled with texts. Moreover, a multinominal Naïve Bayes classifier is used for textual data classification because it is an adjustable algorithm for discrete features that can easily be counted in the form of frequency.

3.2 Linear SVM

Linear Support Vector Machine is a binary classifier that can only separate two classes into positive and negative class. With our large dataset, Linear SVM is appropriate. Linear SVM finds a straight line for one dimension or hyperplane for more than two dimensions to classify the features. If the feature is more confident, it should be far from the decision boundary and if the feature is close to the decision boundary, a small change to the decision boundary can change the result, so it is less confident. The minimum distance between the decision boundary and training instances is called margin. If the margin is maximised, it is the best hyperplane for SVM. To optimise the hyperparameter C, we used GridSearchCV and used C that has the highest score.

Compared to the K-nearest neighbours, the training data is used to learn the weight vector and intercept, whereas the KNN model should memorise all training data.

3.3 Logistic Regression

Logistic regression is a classification model that predicts whether an outcome is more likely true or not based on a given feature set. The model uses a logistic function, also known as the sigmoid function, on top of linear regression to predict whether an input belongs to a class. As opposed to linear regression, there is no analytical solution that minimises the mean squared error of the sigmoid function, and thus the optimal beta coefficients have to be approximated iteratively. The aim of logistic regression is to take out the association between one dependent binary variable and nominal and ordinal variables. The reason for using logistic regression is that the dependent variable of linear regression is only continuous and even sometimes, the probability exceeds 1 or go even below 0. However, because logistic regression is based on the sigmoid function, the output probability is always confined to a value between 0 and 1. Also, grid search is used to tune the hyperparameter C and maximum number of iterations are limited to 5000.

3.4 K-nearest Neighbours

K-NN is a non-parametric supervised learning method, which is useful for both classification and regression. It classifies the test input according to the majority class of the K

nearest training instances. K-NN assumes that the similar features are close together and group them into the same class. The advantage of using K-NN is that it does not require building an additional model and it is easy to apply.

The most important aspect of this supervised learning algorithm is the choice of the appropriate "K" value. Large K leads to high bias and low variance, whereas small K results in low bias and high variance. By implementing various K values, we can build the most suitable fit model to our data.

3.5 Stacking

Finally, stacking classifier is implemented to get the output of using the baseline and other models with the meta-classifier, Random Forest with n_estimators = 100. Since KNN performed worse than other models, we excluded KNN from the model. Stacking uses the strength of each classifier by using their outputs as its inputs, so it can possibly improve the baseline classifier and make a better prediction.

4. Final Implementation / Result

For each predictive model, we decided to implement pipeline, combining all data processing methods, including TFIDF and SelectKBest and each classifier to measure its performance. SelectKBest only retains the k number of features that have the highest scores; therefore, it could make an improvement.

Classifiers	Train Accuracy	Holdout Accuracy	5-Fold CV Mean Accuracy
Multinomial Naive Bayes	0.7023	0.6502	0.6336
Linear SVM	0.8230	0.6575	0.6604
Logistic Regression	0.7555	0.6786	0.6658
KNN	0.6197	0.6204	0.5890
Stacking	0.8856	0.6813	0.6658

Table 1- Accuracy of the Models

As it can be seen in the Table 1, most predictive models, Linear SVM, Logistic Regression and stacking had higher accuracies than Multinomial Naïve Bayes. Stacking performed the best overall with the highest accuracy of 0.89, 0.68 and 0.67 while KNN performed most poorly.

Fold	1	2	3	4	5
Multinomial Naive Bayes	0.6462	0.6340	0.6401	0.6200	0.6275
Linear SVM	0.6750	0.6618	0.6753	0.6390	0.6511
Logistic Regression	0.6863	0.6705	0.6757	0.6424	0.6541
KNN	0.5898	0.5895	0.5892	0.5870	0.5894
Stacking	0.6771	0.6670	0.6805	0.6399	0.6642

Table 2- 5-Fold Cross Validation Score

Stratified K-Fold Cross Validation was used instead of normal Kfold because Stratified ensures that the distribution of the labels is maintained in each fold of the dataset as it preserves the original imbalance of the data. According to the Table, all five classifiers have low variance, producing similar scores between 5 folds.

When tuning the hyperparameter for Linear SVM and Logistic Regression, grid search was done which returned maximum accuracy at C=1 for both classifiers. However, this only impacted the accuracy very slightly.

5. Discussion

After pre-processing the raw data and applying TF-IDF, the size of the words significantly decreased to 7260. This was due to removal of too frequent and infrequent words, so that the machine only considers the meaningful terms that are related to the sentiments of the Tweet. However, it could also mean that we have less data to process and refer to when classifying the text with correct labels. In fact, this issue was already observed from the raw data that was heavily skewed towards the neutral sentiment.

Label	Count
Neutral	12659
Positive	5428
Negative	3715

Table 3- Data Distribution for Each Labels

Relatively low accuracy for our

predictive models was mainly due to the imbalance of the class. The ratio of each labelled sentiments is unevenly distributed as presented in Table 3. Instead of having equal amount of text and labels, this biased data impacts the result of the training model. After training the data, the train accuracy and holdout accuracy had a big gap, and this can be predicted that the big difference in accuracy is due to the overfitting of the data.

Overfitting can happen in several cases during the process of training the sample data. For this situation, a limited training set is not a problem because we have a large size of training data. However, we are expecting that the overfitting happened as a consequence of the non-randomness of given training data. The training data is weighted heavily to the neutral labelled tweets, and we do not have sufficient training samples for Negative and Positive which led to overfitting. Due to these issues, the prediction model of the test set was also skewed towards neutral sentiment as observed in the distribution table below.

Label	Count
Neutral	4097
Positive	1153
Negative	849

Table 4- Data Distribution of Prediction for Each Labels

Label	**Precision**	**Recall**	**F1**
Negative	0.52	0.39	0.45
Neutral	0.69	0.79	0.74
Positive	0.62	0.52	0.57

Table 5- Classification Report of Linear SVM

Table 5 displays the precision, recall and f1 score of Linear SVM. Precision is the pro rata between true positive and false positive where recall evaluates the accuracy of trained model in terms of true positives. From the table above, we observed that the precision rate of neutral label is higher than precision of negative and positive sentiment. Precision is influenced by biased train sample because the given data include more neutral tweets as mentioned previously. Moreover, in the train sample, the number of positive Tweets were greater than the negative, thus positive precision has to be higher than the negative. However, we can also say that recall for neutral data is significantly higher than the other sentiments also because of the imbalance of the data.

F1 score refers to the harmonic mean of recall and precision, and generally, it is better than accuracy. It can only be 1 when precision and recall are equal to 1. Moreover, as F1 score gets close to 1, it means that precision and recall are both high which is very ideal. Obviously, F1 score for neutral sentiment should be high, because precision and recall are relatively higher than any other sentiments, and it means the model is working decently on the neutral labelled data. However, as mentioned above because of the biased training data, results for negative and positive label data are relatively not good for F1 score.

In terms of the performance of our predictions, KNN has shown the weakest performance in classifying the sentiment for train and holdout set among other machine learning models. It has low train data accuracy than other models, but we can assume that it has a higher precision rate compared to other models. Generally, precision and recall are trade-offs, so it works the opposite.

Stacking ensemble of three best performing classifiers have the highest accuracy among using a single model because it uses multiple predictions to build a new model. It incorporates the result of the most effective classifiers, thus performing even better in predicting the test data with the highest accuracy.

6. Conclusions

By using various kinds of machine

learning methods such as Naïve Bayes, SVM, and etc., high accuracy for classifying the sentiment of the Tweets was attainable. Although the model was appropriate for the data type, there might be some problems such as overfitting that can affect the accuracy. Hence, in future studies, we should find possible methods to improve the prediction perhaps, through adjusting the more complex hyperparameters aside from C, and enhancing the text cleansing to leave only the significant information to process.

7. References

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