COMP30027 Report

1. Introduction

There is no doubt that online messaging plays an indispensable role in our modern life. One of the most popular social network app, Twitter, is such a place where people post around 500 million tweets per day for sharing their daily lives (Anon., 2020). Arise from the requirement of some businessmen (e.g., restaurant owners), and it is crucial to know costumers' feedback, which acts as a stimulus to implement Natural Language Processing(NLP). This report aims to introduce one of its applications, sentiment analysis, that evaluating and comparing methods applied to predict costumers' ratings via their feedback. Three main topics will be included, classifier development, feature selection, and performance evaluation.

2. Project Overview

Sentiment feedback depends on two ingredients, feeling of each vocabulary and emotional rating (e.g. rating funny). Related work will be focused on how to use these two parts to build our best model. Firstly, we convert a training count-vectorizer into a vocabulary dictionary, where keys and values are words and sentimental scores. In this problem, we will build sparse matrice for training instead of data frames, since David Z. (Ziganto, 2017) suggests that the sparse matrix runs faster when dealing with enormous features and instances. Meanwhile, we also use doc2vec, a combination of a paragraph matrix and a series of word2vec, as training data because it takes gramma into consideration. Additionally, single statistical models such as support vector classifier(SVC), stochastic gradient descent classifier(SGD), logistic regression(LR) have been applied to make predictions. As stated by Sangarshanan, sometimes a single model would have less accuracy than a combined model (Sangarshanan, 2018), compared with others'. It is predictable that ensemble approaches would dramatically increase accuracies as they are considered to be less biased.

3. Processing

3.1 Baseline Model Choosing

Before conducting the project, building a baseline is essential. Without a baseline, we are not able to analyze performance or adjust out choices in the project (Røberg, 2020). Hence, we firstly choose a collection of Naïve Bayes(NB) models as baselines. *Edwin C*. demonstrated "*Naïve Bayes classifier will converge quicker than discriminative models like logistic regression*" (Chen, 2011) and even the training dataset is large, usually NB classifier is well-performed. According to the table below, accuracies of NB classifiers are generally acceptable.

Type of NB	Accuracy (Train)	Accuracy (Develop)
Multinomial	0.90	0.82
Gaussian	0.73	0.72
Bernoulli	0.97	0.74
Complement	0.90	0.82

Table 3.1.1- Naïve Bayes classifiers and corresponding accuracies

Obviously, GaussianNB is the worst NB classifier, where it requires continuous data following normal distribution. The reason why it is ill-performed could be occurred by using doc2vec, which it is not strictly continuous. Besides, when setting up doc2vec as training data, doc2vec does not consider distance, losing information. The average accuracy is 0.82, thus, it is the baseline when evaluating our models.

3.2 Filtering and Feature Selection

Since we have a tremendous number of features, it is necessary to make feature selection by removing words(e.g. "the") with less sentiment. Further, in such document classification, impact of less informative but more frequently-occurred features should be scaled down. Thus, we convert training data into tf-idf to adjust each feature's weight. To identify best combination of parameters of tf-idf and SelectKBest, we build a pipeline through grid search. After testing all possible hyperparameters, the best result will be

generated.

Name of Parameters	Values
kbest__k	all
tfidf__norm	l2
tfidf__sublinear_tf	True
clf__loss	hinge
clf__penalty	l2

Table 3.2.1- Best combination of hyperparameters selected by grid search(though SGDClassifier)

The table demonstrates that in the best combination, feature selection is not necessary. In addition, hinge loss is preferred, indicating that support vector machine(SVM) has a better performance than logistic regression.

We are also noticed that in the best five results from the grid search, some suggest that feature selection is required. By keeping all other parameters the same, 0.845 and 0.844 are the accuracies for without feature selection and with feature selection. Apparently, applying feature selection will shrink accuracy slightly, but overall our model will run faster.

4. Building and Evaluating Models

Due to the fact that our training data is linear separable, thus, in this project, we apply two linear models, linear support vector classifier(SVC) and logistic regression(LR). When constructing predictive models, general practice is to split the dataset into three parts: training, validation, and testing. In supervised learning, the training set will be input into classifiers for the learning algorithms. The validation set is conjunct with the training set and is usually used to modify hyperparameters according to some guidelines (Brownlee, 2018). Essentially, the testing set is not used in validation because it is alienated to our models.

Thus, in the following parts, we will discuss training and validation scores for each model.

4.1 LinearSVC (On Sparse Matrix)

We firstly apply SVC with default parameters, which achieves 0.847 as training accuracy and 0.852 as development accuracy. To find the overall accuracy of this default model we need to derive its confusion matrix as below.

Label	Precision	Recall	F1-score	support
1	0.84	0.63	0.72	717
3	0.71	0.65	0.68	1854
5	0.89	0.94	0.92	5850
accuracy			0.85	8421
Marco avg	0.81	0.74	0.77	8421
Weighted avg	0.85	0.85	0.85	8421

Table 4.1.1- Confusion matrix of default SVC

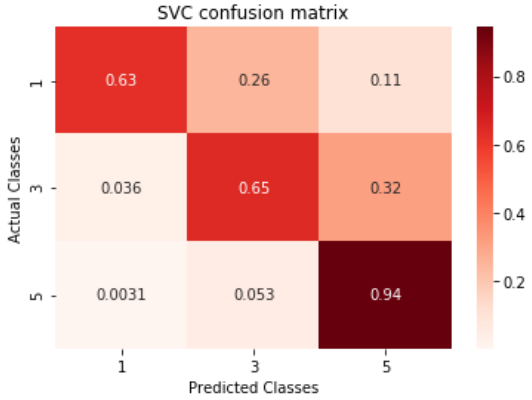

Figure 4.1.1- Normalized confusion matrix of default SVC in heatmap

Overall, except label 5, SVC has unacceptable accuracies. A possible reason of low accuracy of label 3 would be that it is in the middle, which is more ambiguous and confusing than other two labels. When it comes to low accuracy of label 1, table 4.1.1 illustrates that we train the model with inadequate instances with label 1. Increasing training size would possibly rise label 1 accuracy.

We plot two graphs, validation curve and learning curve.

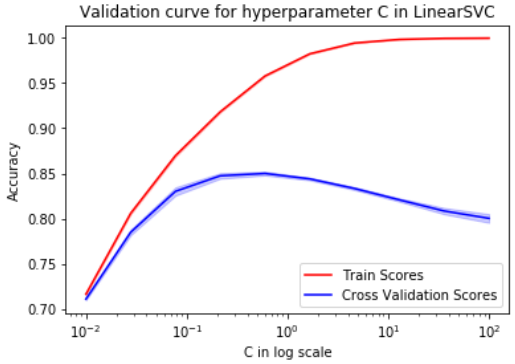

Figure 4.1.2- Validation curve for hyperparameter C in SVC(shadow area for std)

Firstly, it is the validation curve on log scale versus its accuracy. Apparently, training score is continuously climbing up, and it is paralleled with x-axis when logC is greater than 10, where the training result is more precise. While cross validation score is growing followed by decline. The larger C is, the larger std for cross validation, which could be attributed to overfitting with big-sized training set.

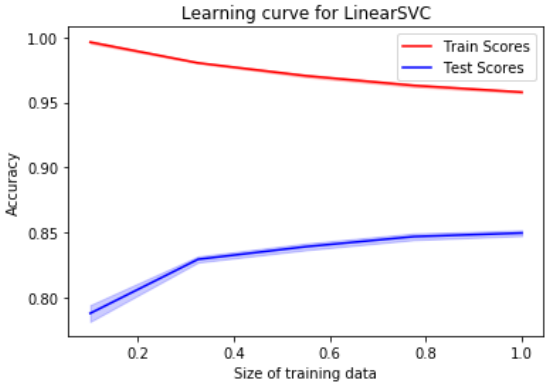

Figure 4.1.3- Learning curves for SVC based on size of training set(shadow area for std)

Then, it is the learning curve of training size versus its accuracy. By inputting all data, both training and testing accuracies are greater than 0.8. With training data incrementing, training score shrinks but test scores rise. The more inputting training data, the less std for test scores. The difference between two curves shrinks as training data increases, which overfitting is more likely to happen. Besides, this model is stable because two curves are nearly paralleled.

4.2 Logistic Regression (On doc2vec)

Secondly, LR was applied in this project. Before we implement this model, we use grid search to ascertain the best combination of hyperparameters.

Name of Parameters	Values
C	0.5
multi_class	multinomial

Table 4.2.1- Best combination of hyperparameters selected by grid search(though LR with default solver)

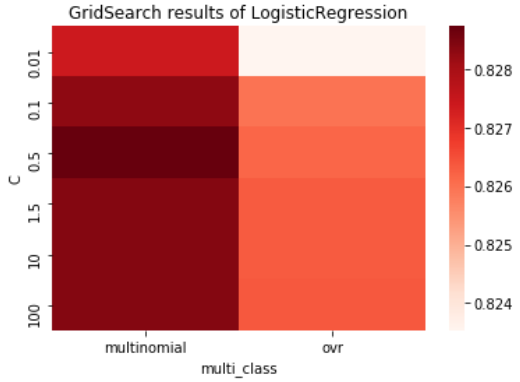

Figure 4.2.1- Grid search result of logistic regression in heatmap(multinomial versus one-vs-rest)

According to two charts above, no one will dispute that multinomial has a better score than one-vs-rest(ovr) no matter which value C is. Especially when C is around 0.5, LR's accuracy hits the peak. Multinomial is a joint model while ovr is a stratified model, the former model would be possibly satisfied with joint distributions in doc2vec.

Figure 4.1.3- Learning curves for LR based on size of training set(shadow area for std)

Compared with SVC, this figure shows the similar trend. However, these two curves converge quicker than SVC, indicating that we only need to use about 35% training data, even though losing slight accuracy(0.85 for SVC and 0.82 for LR) and std is larger when more training data is used.

5 Conclusions

In conclusion, we have processed the raw data and used the sparse matrix and doc2vec as our training set. Among two linear models, SVC and LR, LR is better-performed based on

the facts, because we use less training data and our model runs faster when we face a massive amount of data in practice. Its average accuracy is around 0.82, equaling to NB baselines. Overall, our final LR is a competent model to predict labels from twitter.

6 References

1. Anon., 2020. *Twitter Usage Statistics.* [Online]
   Available at:
   https://www.internetlivestats.com/twitter-statistics/
   [Accessed 25 May 2020].

2. Brownlee, J., 2018. *A Gentle Introduction to k-fold Cross-Validation.* [Online]
   Available at:
   https://machinelearningmastery.com/k-fold-cross-validation/
   [Accessed 25 May 2020].

3. Chen, E., 2011. *Choose a machine learning classifier.* [Online]
   Available at:
   https://blog.echen.me/2011/04/27/choosing-a-machine-learning-classifier/
   [Accessed 25 May 2020].

4. Kirk, D. B. & Hwu, W.-m. W., 2017. Parallel patterns: sparse matrix computation: An introduction to data compression and regularization. In: S. Merken, ed. *Programming Massively Parallel Processors.* s.l.:Morgan Kaufmann, pp. 215-230.

5. 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.

6. 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.

7. Røberg, Ø., 2020. *Why Your Baseline is Essential in Project Management.* [Online]
   Available at:
   https://www.safran.com/blog/why-baseline-is-essential-in-project-management
   [Accessed 25 May 2020].

8. Sangarshanan, 2018. *Two is better than one: Ensembling Models.* [Online]
   Available at:
   https://towardsdatascience.com/two-is-better-than-one-ensembling-models-611ee4fa9bd8
   [Accessed 25 May 2020].

9. Suresh, K. et al., 2017. Comparison of joint modeling and landmarking for dynamic prediction under an illness-death model. *Nation Library of Medicine,* 59(6), pp. 1277-1300.

10. Ziganto, D., 2017. *Sparse Matrices For Efficient Machine Learning.* [Online]
    Available at:
    https://dziganto.github.io/Sparse-Matrices-For-Efficient-Machine-Learning/
    [Accessed 25 May 2020].