**COMP30027 Project 2 Report**

1. **Introduction**

Machine learning is the art of prophecy. By training a model with a huge amount of data, the system can accurately predict the behaviour of an instance based on given features. And the key step is increasing the predicting accuracy among all types of data.

In this Project, the methods to improve predicting accuracy of sentiment analysis problem star ratings for reviews on restaurants, is investigated by building and critically analysing some supervised Machine Learning methods. We have collected some data of review text and the corresponding rating with some other data of reviewers. The goal is to automatically identify and extract polarity (e.g. positive, negative, or neutral) from reviewer text. Sentiment analysis has a variety of applications in recommender systems, marketing, economics, and social and political sciences. Although this problem has been well-studied, a general solution remains elusive.

1. **Model selection**

**2.0 Input data analysis**

For each instance prediction, a paragraph of review text is provided in a csv file with another csv file contains meta data, including rating of the review. Since the concept is ordinal data with only rating 1, 3 or 5, and the instances are text, we pre-processed the data by vectorize the review paragraph. Since we also obtain part of testing data set, we are able to select model though testing

**2.1 SVM**

Among all types of SVM models, we chose the linear SVM which applies a linear support vector to classify clusters. We did some tests and found that is the most out-performed type of model compared to others.

**2.2 Neural Network**

Can explore simple logic relationship

Slow

Bias

Variance

**2.3 Random Forest (Bagging)**

Random forest models involve instance manipulation techniques, so we have decided to use bagging type of random forest rather than a boosting method. Because boosting involves iterative sampling to minimise the instance bias, it tends to have higher model variance due to over focus on some samples.

Random forest is a bagging technique with instance manipulation approach to combine decision tree classifiers. Weight - equal weight, no boosting.

**2.4 Stacking**

In stacking, we implemented four best performed models found before: NB, SVM, Random forest and MLP neural network. We tried to conduct 5 folds to do cross-validation in metadata.

We also trained a classifier to predict rating based on meta data of classifier, including “vote funny”, “vote cool” and “vote useful”, as we reckon these features are highly corelated with final rating.

Stacking introduces a meta classifier to select good base classifiers in prediction. We select four types of models as those base classifiers: Gaussian Naive Bayes, Random forest, Linear SVM, MLP Neural network. In addition, we also train the other information provided in meta data to predict the rating by Linear SVM. While implementing, we divide the data set into 5 folds and use cross validation method to generate features in the meta classifier. Each time, we train different types of models based on the data from 4 other folds, and make predictions of rating in the fold. After combining the predicting result of 5 folds with these different types of models, we get meta data to classify. Finally, we trained a logistic regression model in classifying the data. As logistic regression applies gradient descent to find a classifying function by optimising a convex set of errors in predicting the rating (Rodrigo, 2019).

In testing data, we have to predict the rating by different types of classifiers to get the test meta data and fit into the final logistic regression model to get the result.

In practice, the stacking system has the best accuracy. The accuracy of one-fold is satisfyingly 84%, the feature manipulation method successfully reduces bias and variance of the model in practice.

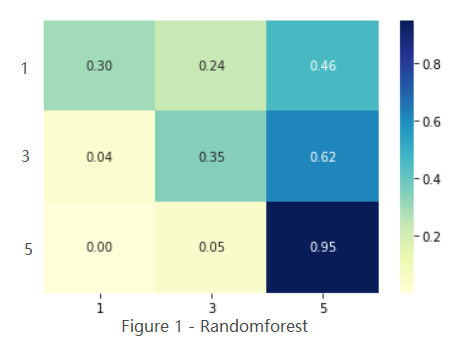
However, the accuracy of 5 folds stacking is only 67%. This demonstrates although cross-validation has reduced model variance, the split data set may be failing in containing all features of instances, resulting in inaccurate metadata. The accuracy in classifying data in each other groups is largely reduced. Finally, the stacking model is underfit and has great instance bias.

After testing lots of different models, we decided to implement four systems in the end: SVM, Neural network, Random forest and Stacking. SVM and Neural network are the best performing systems with single classifier. Random forest and Stacking are combinational classifiers based on instance manipulation or feature manipulation.

**2.1.1 Subsubsection**

Figures should be placed in the text, not at the end. Figures must be captioned and explicitly mentioned in the text (Figure 1).

**Figure 1-** Figure captions should appear below the image (Times New Roman 9, Aligned Left, Single Line, 0 pt before, 12pt after, no indentation).



1. **Critical analysis**

**3.1 SVM**

Suppose the number of instances in the data set is n. The time complexity is O(n^3). It grows at least like n^2 when C, the bound of the support vector, is small and n 3 when C gets large. (Bottou, 2007)

The space complexity is O(n^2).

Support vector machine classifiers use a kernel function to classify the data, most popular types of kernel are linear, RBF and polynomial kernels. In model selection, we use the train data and the test data only based on the review text. Testing different types of kernel in SVM classifiers, we found the SVM implemented linear kernel function is surprisingly the most accurate one. This implies the word data usually has a typical tendency in meaning. Each vocabulary has positive or negative meaning, the review with a lot of positive words would generally obtain a higher rating. As a result, the frequency of vocabulary used is linearly correlated with the rating. Although using a “RBF” or polynomial kernel enriches feature space to add more power of explanation, they may cause overfitting problems which increase the model variance. (Gori 2018)

Linear SVM has balanced model bias and model variance; it is neither underfitting nor overfitting too much. So, it has the best accuracy.

**3.2 Neural Network**

Can explore simple logic relationship

Slow: The time complexity is

The space complexity is

Bias

Variance

**3.3 Random Forest (Bagging)**

Slow; The time complexity is

The space complexity is

Bias

Variance

**3.4 Stacking**

Time complexity of Gaussian naive bayes is O(nd), space complexity is O(nk), where n is number of instances and k is number of class labels. Time complexity of Linear SVM is O(n^2) while space complexity is O(1). Time complexity of other instances with Linear SVM is O(n^2) with space complexity O(1). Random forest has time complexity O(n\*log(n)\*d\*m) and space complexity O(dm), where d is max depth of the trees and m is the number of trees. Neural network has time complexity of O(n∗t∗(ij+jk)) and space complexity O(i+j+k), where n is number of instance, t is number of iterations, i, j, k are number of nodes in first 3 layers, which are 80, 100, 100 in model built. The final logistic regression step obtains time complexity of O(nd), and its space complexity is O(d), where d is dimension=5, n is number of instances. (Kumar, 2019)

Therefore, the whole stacking system has time complexity is O(f(n^2+nd+n^2+nlog(n)dm+ n∗t∗(ij+jk))+5n)=O(f(nlog(n)dm + n∗t∗(ij+jk))), as they are the most dominated term, where f is the number of folds applied in the end. The run time complexity is very slow compared to just implement a single system.

And the space complexity is O(1+ nk + 1+i+k+j+ d)=O(nk). Overall, the memory space it takes is not very large.

Bias is reduced compared to each classifier, because the possibility for all classifiers to misclassify a certain type of instance is low. Model bias is averaged out to be closer to true distribution of data, by the combination of these classifiers.

Variance is reduced as different types of classifiers supplementarily handle instance prediction correctly; successfully cover the mistakes each other. These made the final prediction model more accurate. By applying n-fold cross-validation, we can further reduce the model variance by allowing models build on proportion of data set to classify different instances, rather than just the training set. This can avoid overfitting.

Filling text; filling text. Filling text; filling text. Filling text; filling text. Fillingtext;. Filling text; filling text.

Filling text; filling text. Filling text; filling text.

Tables also need to be captioned and explicitly mentioned in the text (see Table 1). If you don’t want the text in your tables to be counted in your word count, you can add your tables as pictures.

|  |  |
| --- | --- |
| Document | Features |
| AAA | text |
| BBB | text |
| CCC | text |

**Table 1-** Table captions should appear below the image (Times New Roman 9, Aligned Left, Single Line, 0 pt before, 12pt after).

Text of the subsubsection. Text of the sub- subsection.

1. **Future improvement**

In the future, we can also include PCA in the final model to conduct dimension reduction in predicting the meta data.

In the future, we shall try differing models with different numbers of folds to investigate the ideal number of folds of data to balance the model.

1. **Conclusions**
2. **References**

Bottou, L. & Lin, C.-J., 2007. Support Vector Machine Solvers. *Large-Scale Kernel Machines*. Available at: <https://leon.bottou.org/publications/pdf/lin-2006.pdf>.

Gori, M., 2018. *Machine learning: a constraint-based approach*, Cambridge, MA: Morgan Kaufmann Publishers.

Kumar, P., 2019. Time Complexity of ML Models. *Medium*. Available at: <https://medium.com/@paritoshkumar_5426/time-complexity-of-ml-models-4ec39fad2770> [Accessed May 27, 2020].

F MELLO, RODRIGO. ANTONELLI PONTI, MOACIR, 2019. *MACHINE LEARNING: a practical approach on the statistical learning theory*, S.l.: SPRINGER.

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.

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.