Predicting Star Ratings of Restaurants

1. Introduction

The goal of this project is to successfully predict the star ratings of restaurants as 1, 3 or 5 stars based on its reviews. This is a form of sentiment analysis where one needs to analyse and obtain ‘sentiment’ from text. A basic task in sentiment analysis is to look at emotional states such as ‘happy’ and ‘sad’ and assign these to ‘positive’ and ‘negative’ predictions. The usefulness of these predictions can be very helpful in social media monitoring and targeted advertisements.

2. Models

Since this is a multi-class classification problem, we need to make sure any classifications such as Logistic Regression have an extension to multi-dimensional counterparts. Time complexities is a potential problem as a result of multi-dimensional approaches. Therefore, a Naïve Bayes classifier was chosen as an indicator as it grows linearly with the number of instances and for its simpleness. Other classifiers used are: Logistic Regression and a Stacking classifier.

2.1 Naïve Bayes Classifier

As stated above, this model was used as a benchmark to test time and space complexity. A Naïve Bayes model consists of a class prior; this should perform well for sentiment analysis and text classification as it can assign an instance without any shared features with the training data.

2.2 Logistic Regression Classifier

In this problem, Logistic Regression was used as a multinomial variant Logistic Regression. Logistic Regression requires large N in order to be effective and the training data has over 28,000 features. This is ideal to be used in sentiment analysis and text classification.

2.3 Stacking Classifier

Stacking works by smoothing the errors over a range of algorithms with different biases. A classifier was trained over the outputs of three base classifiers by using nested cross-validation to reduce bias. The three base classifiers were: Random Forest, Naïve Bayes and Logistic Regression.

3. Pre-process and Vectorization

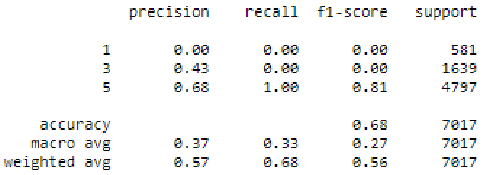
The training data included unwanted numbers, special characters and spaces. The features were stripped of all punctuations and non-word characters such as numbers and special characters. Whilst a review that includes ‘!!!!’ such as ‘AMAZING!!!!’ or ‘AWFUL!!!!’ can be classified pretty easily as 5 stars and 1 star respectively, the same decision can be made just as easily without the exclamation marks. Therefore, the decision was made to strip all punctuation and special characters.

Single characters at the beginning of the text were replaced by a space, multiple spaces were replaced by a single space as multiples spaces are not ideal. The data was also converted to lower case in order to match up words and finally lemmatization to avoid having features that mean the same thing but are written differently such as ‘table’ and ‘tables’.

The vectorization methods of Frequency Counts and Term Frequency Inverse Document Frequency had to be chosen for each classifier. After testing classification accuracy for both vectorization methods, a conclusion of Term Frequency Inverse Document Frequency was used for the Stacking classifier, whilst Term Frequency was used for the Naïve Bayes and the Logistic Regression classifiers. The decision was made based on the highest classification accuracy on a 75-25 split on the training data.

4. Evaluation of Classifiers

4.1 Naïve Bayes Classifier

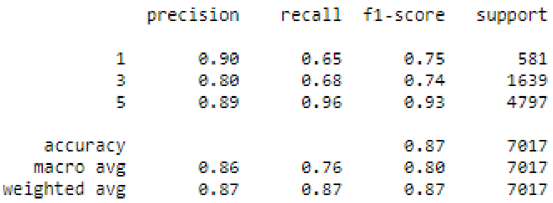


```
              precision    recall  f1-score   support

           1       0.00      0.00      0.00       581
           3       0.43      0.00      0.00      1639
           5       0.68      1.00      0.81      4797

    accuracy                           0.68      7017
   macro avg       0.37      0.33      0.27      7017
weighted avg       0.57      0.68      0.56      7017
```

Table 1 – Classification report and confusion matrix of Naïve Bayes

With an accuracy of 68.41%, the Naïve Bayes classifier performed averagely. Since Naïve Bayes assumes all probabilities are independent, by calculating a set of priors, it calculates the posterior and assigns it with the largest corresponding prior, however, it only influences the classifier if a value of a high enough frequency occurs. As Naïve Bayes predicts by using a product of probabilities, any probability of 0 results in an overall 0. This is why the precision and f-score are set to 0 as there was some zero-division with no parameter to control it. Despite this an accuracy score 68.41% is impressive for Naïve Bayes. This is probably due to some overfitting of words with high frequency from the vectorization method of Frequency Counts.

4.2 Logistic Regression Classifier

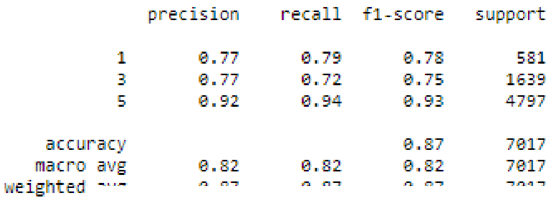


```
              precision    recall  f1-score   support

           1       0.90      0.65      0.75       581
           3       0.80      0.68      0.74      1639
           5       0.89      0.96      0.93      4797

    accuracy                           0.87      7017
   macro avg       0.86      0.76      0.80      7017
weighted avg       0.87      0.87      0.87      7017
```

Table 2 – Classification report and confusion matrix of Logistic Regression

A multinomial Logistic Regression yielded an accuracy of 87.22% on a 75-25 split. In theory, Logistic Regression should be a major improvement on its counterpart Naïve Bayes and this is true for this problem. Whilst requiring feature scaling and being slow to train, an increase in nearly 10% over the Naïve Bayes classifier is a vast improvement. Since Logistic Regression is suited to frequency-based features, Frequency Counts vectorization was used. With over 21,000 features, Logistic Regression was effective in classifying ratings.

4.2 Stacking Classifier



```
              precision    recall  f1-score   support

           1       0.77      0.79      0.78       581
           3       0.77      0.72      0.75      1639
           5       0.92      0.94      0.93      4797

    accuracy                           0.87      7017
   macro avg       0.82      0.82      0.82      7017
weighted avg       0.87      0.87      0.87      7017
```

Table 3 – Classification report and confusion matrix of Stacking Classifier

The Stacking classifier yielded the highest classification accuracy of 87.39% which is slightly better than the Logistic Regression classifier. The classifiers were run numerous times and on average the Stacking classifier had a .15% higher accuracy than the Logistic Regression classifier. The vectorization method Term Frequency Inverse Document Frequency was used because of the probabilistic counts. The three base classifiers in the Stacking classifier gave varying variances and biases which meant that the prediction of new instances based on the probabilities of the base classifiers performed very well. Whilst being mathematically simple, this classifier was the most computationally expensive out of the three main classifiers.

Comparing the Stacking classifier and Logistic Regression classifier in the Kaggle competition. Both classifiers used all of the 28,069 features in the review_text_train.csv as training data and 7019 features in the review_text_test.csv as testing data and was scored on 30% of the final 7019 predictions. The accuracies follow a similar pattern to that of the 75-25 split of just the 28,069 features. The Logistic Regression yielded an accuracy of 85.94% whilst the Stacking classifier yielded an accuracy of 86.70%. This increase of .76% is expected and reasonable as both clasifiers are within their margins of errors.

5. Conclusion

Overall, the classifiers performed well especially the Stacking classifier as it scored in the top 9.96% percentile in the Kaggle competition. Pre-processing helped improve the classifier's accuracies by an average of 0.5%. However, feature selection using Mutual Information would theoretically improve the accuracies of the classifiers. Feature selection was not performed due to time constraints. A conservative guess of 0.5% to 1% improvement in accuracy can be made if feature selection was implemented in the pre-process stage, after data cleaning but before vectorization.

6. References

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.