Sentiment Analysis by Star Rating Prediction of Yelp Reviews

Giacomo Anerdi
Department of Data Science and
Knowledge Engineering,
Maastricht University

Hendrik Simon Baacke
Department of Data Science and
Knowledge Engineering,
Maastrich University

1 Introduction

With the rise of social media, online forums and other user driven websites, applications of sentiment analysis have become more widespread. In the domain of Natural Language Processing tasks, sentiment analysis stands out as one of the most applicable and promising for data focused corporations. One example of where sentiment analysis can be applied is in predicting user scores in 'Yelp' reviews. 'Yelp' allows users to write reviews on businesses and rate them with a score between 1 and 5 stars. Because there are 5 different scores that users can give, the kind of supervised learning that is being performed is multiclass classification. Differentiating between more than two classes offers benefits in certain applications over labelling a text segment as either positive or negative. Emotions are not well defined, therefore ordering sentiments on a scale from 1 to 5 is more expressive. A trained classifier could, for example, be implemented in a customer service response software which orders customer inquiries from most to least urgent based on where the identified sentiment fits in the spectrum. Two different model approaches are considered in this report, firstly a "Bag of Words" (BoW) model with a Naive Bayes classifier and secondly a "Word2vec" model with either XGBoost or a Support Vector Machine (SVM) classifier.

2 Previous Work

A description of how to construct word vectors for sentiment analysis is given by Maas et al. (2011). The conceptual foundation of word vectors and their feasibility for measuring syntactic and semantic word similarities is provided by Mikolov et al. (2013). Research on sentiment analysis with the Word2vec model is conducted by Lilleberg et al. (2015) and Acosta et al. (2017). Both papers describe which classifiers among SVM and logistic regression perform best in the given context of supervised text classification tasks. A model to build a sentiment dictionary using Word2vec is presented by Xue et al. (2014). The concept of appraisal which denotes how language is used to express attitude towards a target is introduced by Whitelaw et al. (2005). This is then used to build a lexicon of appraising adjectives and their modifiers based on extracting and analysing appraisal groups. XGBoost finds usage in sentiment classification in Jabreel and Moreno (2018), where it is compared with a deep learning approach called N-Stream ConvNets.

3 Data

The dataset was retrieved from the official Yelp website (Yelp). However, it contains over 6.5 million instances which is too large for the scope of this report. Therefore the original dataset was cut randomly. The training set now contains 40'000 reviews and the hold-out (test set) consists of 10'000 entries which follows the 80/20 guideline. Moreover, 8'000 instances (20%) of the training set are used as validation which helps to optimise the hyperparameters of the classifier and leads to less overfitting of the training set.

3.1 Analysis of the dataset

In this section, a brief analysis of the dataset is provided to understand the balance of the classes as well as their proportions in the training and test set. As illustrated in Figure 1, both the training and test set have unbalanced classes. The 'extremes' of the sentiment spectrum which are 1 and 5 star reviews are more numerous while the 2, 3 and 4 star reviews occur less often. However, as seen in Figures 1 and 2 the training set and test set have roughly the same proportions for each class which means that this unbalance is likely natural in the context. The proportion of the classes differs by 0.36% on average. This negligible difference in group composition hints that the given data is well suited for training and testing a classifier.

Stars	Reviews	Percentage
1	5821	14.55
2	3205	8.01
3	4391	10.98
4	8845	22.11
5	17738	44.35

Table 1: Training set description

Stars	Reviews	Percentage
1	1424	14.24
2	750	7.50
3	1076	10.76
4	2217	22.17
5	4533	45.33

Table 2: Test set description

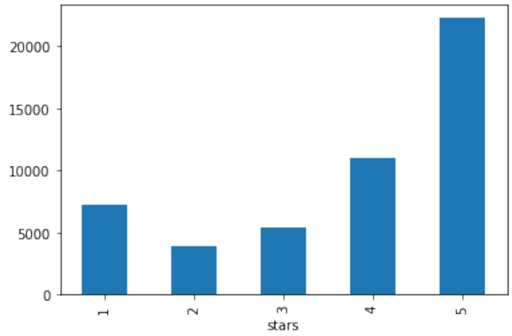

Figure 1: Distribution of dataset used

4 Model

Before starting with the actual sentiment classification task, text pre-processing needs to be done which entails balancing different heuristics. On one hand, performing invasive techniques like stemming may dramatically reduce the type count and clean the text up neatly. On the other hand, important and non-obvious clues for the classifier might get lost. In order to find the best mix, experimenting with different techniques has been conducted which will be further discussed in the 'Analysis' section. A less invasive approach is hereby found to yield the best overall results. The text is cleaned by making every letter lower case and removing punctuation. Furthermore, all numbers are omitted from the text since they may be present in any sentiment context and do not provide additional useful information for the classifier. Additionally, all stop words in the "*nltk*-library" are removed from the text with exception of personal pronouns: 'he', 'she', 'they' and 'we'. These nouns are usually used in the context of referring to staff of businesses, most often in a negative way. For example when a customer service assistant has been impolite towards the customer. Overall the goal is to keep expressiveness for the classifiers high whilst omitting noise. The same pre-processing was performed for both the BoW approach and the Word2vec based classifiers. Further research can be conducted about what is the best choice for either model which is done by Haddi et al. (2013).

4.1 Bag of Words

The Bag of Words model is a vector representation of a given text. The occurrence of each word is counted and saved as a vector. This kind of model completely disregards the ordering of these words and therefore their context.

4.1.1 Naive Bayes classifier

Naive Bayes is a very simple classifier which treats the occurrences of each word in the corpus as a probability of it appearing in any other unlabelled set. This classifier serves as a baseline of what can be achieved with the dataset at hand as well as it gives a general lower bound of the performance that the other more advanced classifier can try to beat.

4.2 Word2Vec

The implemented Word2vec model uses a method called review2vec() which sums all the vectors of the words of each sentence. By doing so, a unique vector per review is generated to compare it more easily with its duplicate candidate. The vector is then normalised since some questions might be longer than others potentially resulting in some bias when fed into the classifier. A matrix is then returned, containing a separate vectorised review in each row and its features made with Word2vec in each column.

4.2.1 XGBoost classifier

XGBoost is a decision tree boosting system which combines a large number of regression trees with a small learning rate. Moreover, it has the advantage of a vast amount of possible customisation with parameter optimisation. According to Chen and Guestrin (2016), trees which are constructed earlier are significant, whereas lately added trees are regarded as unimportant. This property is especially desirable in sentiment analysis because text data is noisy even after text cleaning. The effect of this noise on the classifier is mitigated since rules solely derived from the noise are likely to be learned late in the fitting process. The XGBoost algorithm described here is based on XGBClassifier from the Python (version 3.7.3) *xgboost* library. Firstly, manual parameter tuning is applied. The maximum depth of the tree is limited to 5 (compared to default of 6) and the learning rate is decreased from 0.3 to 0.03. The parameter constellation yields higher accuracy while the classifier is less likely to overfit the training set. The gamma parameter is defined by the best regularisation parameter found in the hyperparameter tuning with the validation set.

4.2.2 Support Vector Machine classifier

SVMs describe a class of supervised learning algorithms. These offer probabilistic binary linear classification which scales well into high-dimensional feature spaces using the kernel trick. SVMs are well suited for sentiment prediction tasks as analysed in previous research conducted by Lilleberg et al. (2015) and might therefore also yield good results for this application. The SVM in this particular case is imported from the *sklearn* Python (version 3.7.3) library Here the C parameter, which determines the magnitude of avoiding to misclassify a given training example, is defined in hyperparameter optimisation with the validation set, similarly to how it works for XGBoost's gamma value.

4.3 Hypothesis

It is expected that the BoW model with the Naive Bayes classifier performs worse than the Word2vec model and its classifiers. This is because the model disregards the context of words. The classifier simply learns their occurrence probability patterns. It is expected that classifiers using the Word2vec model perform better because of the approach to represent words as vectors and their ability to interpret the context of words within the sentence. Additionally, the model should scale better with large datasets as described by Mikolov et al. (2013). However, the experiments must show which of the two classifiers, SVM or XGBoost, performs better in the given context. It also needs to be noted that in multi-class sentiment classification without additional heuristics it is expected for the classifiers to not achieve particularly high accuracy scores. This is because the task requires to discretise emotions in multiple classes of a spectrum from most negative to most positive in contrast to a relatively simple binary good/bad classification.

5 Results

In the following section, the measures of the overall accuracy, as well as a weighted recall are given. A weighted recall is used because of the large imbalance amongst the classes. This is also the reason why normalised confusion matrices are presented. The confusion matrices of each classifier are provided in the appendix in section A. The first three matrices, Figures 2, 3 and 4 are normalised, the standard confusion matrices instead are shown in Figures 5, 6 and 7. The normalised confusion matrices are mostly used in the analysis due to the large unbalance of the dataset.

5.1 Naive Bayes

Accuracy	60.76%
Recall	0.6076

Table 3: NB results

5.2 XGBoost

Accuracy	62.76%
Recall	0.6276

Table 4: XGBoost results

5.3 Support Vector Machine

Accuracy	62.93%
Recall	0.6293

Table 5: SVC results

6 Analysis

As expected, the accuracy of the Word2vec based classifiers is higher than that of Naive Bayes over the whole test set. Both XGBoost and SVM classifiers achieve results of around 63% while Naive

Bayes is at 60%. As it can be seen in Figures 2, 3 and 4, the two most dominant classes, 1 and 5 were predicted correctly most often with an accuracy of over 72%. The 4 star reviews were correctly labelled over 60% of the times. However, the 2 and 3 star reviews were classified correctly less than 25% of the times. The likely cause of this poor performance is the small class size in relation to the other classes as shown in Figure 1. Regarding this poor classification of 2 and 3 star reviews, XGBoost performs better than the SVM approach with 8% more correctly classified instances of theses classes. This suggests that the SVM is more of a majority classifier and overfits the data and its class distribution more than XGBoost. This can also be seen when comparing the normalised confusion matrices shown in Figures 3 and 4.

Overall, a clear superiority of one classifier in terms of accuracy for this test set can not be determined. However, the overfitting of SVM must be taken into account when using the classifiers on similar data with different class distributions. The accuracy of around 60% is not very high for all three classifiers. However, as shown in Figure 6, it can be seen that most of falsely predicted scores were off by only one star, which means that the sentiment has been identified correctly, but the discrete label is wrong. A discrimination between close classes can therefore be considered hard for the classifiers. When analysing accuracy with a margin of error of one star, all classifiers scored over 90% as it can be seen in Figures 9, 8 and 10 in the appendix B. The normalised confusion matrices support this statement. These show that the largest proportion of false positives are next to the true positive diagonal for all classifiers.

7 Conclusion

In this report an attempt was made to predict the star rating of Yelp by using different classifiers. It is found that the classifiers XGBoost and SVM which use the Word2vec model perform better than the Naive Bayes classifier using the BoW model.

The imbalance in the Yelp dataset causes the classifiers to be biased towards classes with more instances. Since the training and test set had similar distributions of classes, the trained classifiers still performed relatively well. However, if they were tested on other corpora with different class distributions, a lower accuracy is expected compared to the Yelp data. The SVM algorithm is affected more by that than XGBoost. This hints that adding other heuristics to the model, for example an Emotional Dictionary as described by Xue et al. (2014), should be considered. Further research regarding this would make sense in order to see whether these heuristics offer improvements to the classifier's generalisability. The report furthermore supports the hypothesis that multi-class labelling is quite difficult because of the discrimination between close classes in whose the sentiment is related. This discretisation of a continuous range of emotions is by far the biggest contributor to the relatively low accuracy which was achieved with the classifiers. Nonetheless, if an error margin of just one 'star' unit is taken into account, the classifiers all are able to predict the rating with an accuracy of over 90%. Based on these findings, the task the group set out to do can be viewed as successfully accomplished.

References

Joshua Acosta, Norissa Lamaute, Mingxiao Luo, Ezra Finkelstein, and C Andreea. 2017. Sentiment analysis of twitter messages using word2vec. *Proceedings of Student-Faculty Research Day, CSIS, Pace University*, page 7.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754.

Emma Haddi, Xiaohui Liu, and Yong Shi. 2013. The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, 17:2632.

Mohammed Jabreel and Antonio Moreno. 2018. Eitaka at semeval-2018 task 1: An ensemble of n-channels convnet and xgboost regressors for emotion analysis of tweets. *CoRR*, abs/1802.09233.

J. Lilleberg, Y. Zhu, and Y. Zhang. 2015. Support vector machines and word2vec for text classification with semantic features. In *2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pages 136–140.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 142–150, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tomas Mikolov, G.s Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. pages 1–12.

Casey Whitelaw, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal groups for sentiment analysis.

B. Xue, C. Fu, and Z. Shaobin. 2014. A study on sentiment computing and classification of sina weibo with word2vec. In *2014 IEEE International Congress on Big Data*, pages 358–363.

Yelp. Yelp dataset challenge.

A Confusion Matrices

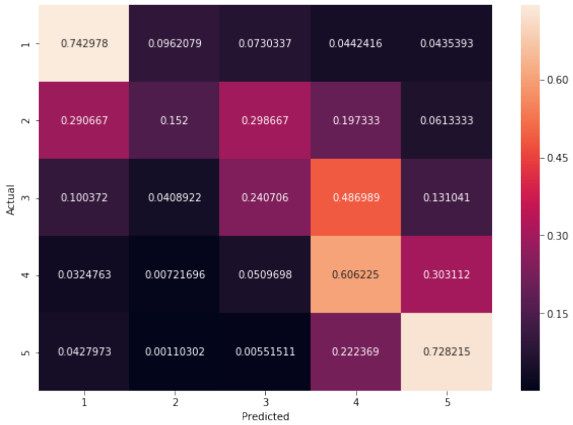

Figure 2: Normalised confusion matrix for the NB classifier

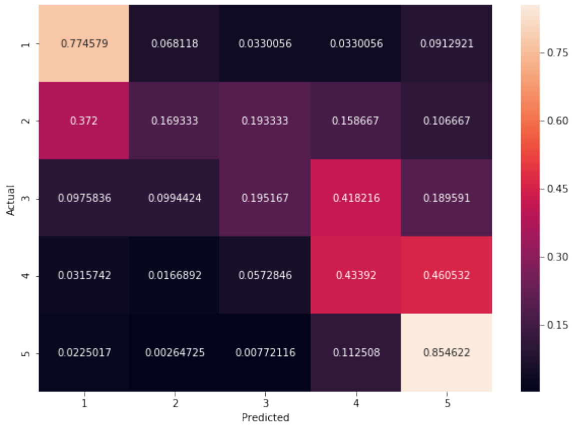

Figure 3: Normalised confusion matrix for the XGBoost classifier

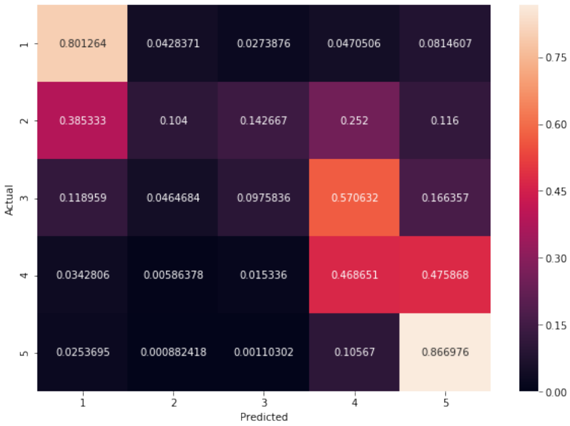

Figure 4: Normalised confusion matrix for the SVM classifier

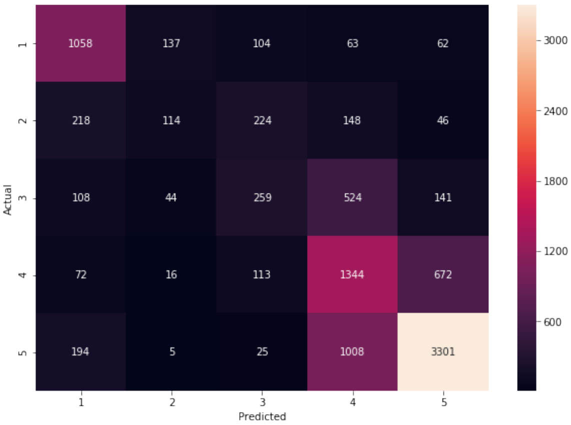

Figure 5: Confusion matrix for the NB classifier

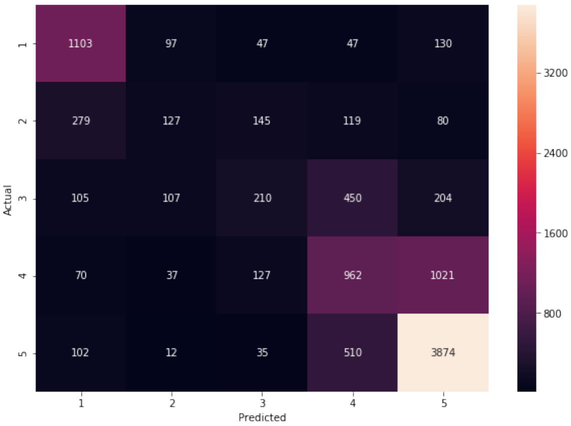

Figure 6: Confusion matrix for the XGBoost classifier

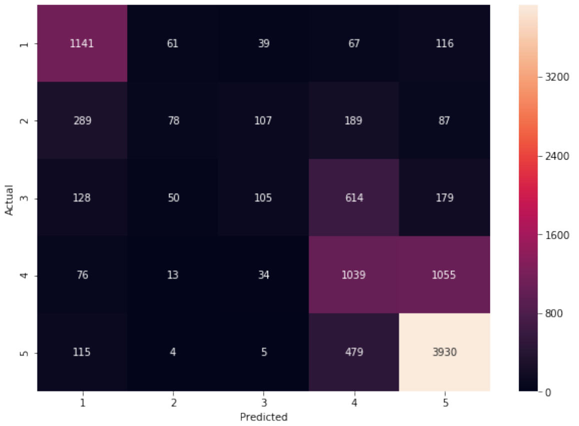

Figure 7: Confusion matrix for the XGBoost classifier

B Absolute error of predicted stars

	Distance from correct class				
Classifier	0	1	2	3	4
Naive Bayes	6076	2940	542	186	256
XGBoost	6276	2736	547	209	232
SVM	6293	2689	553	234	231

Table 6: Absolute error in star rating predicted

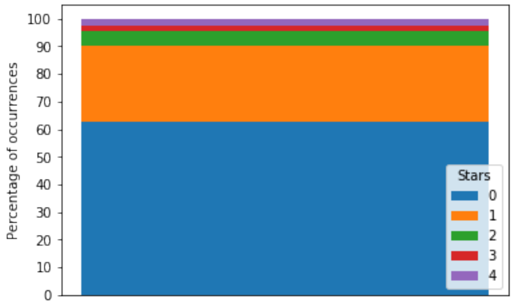

Figure 8: Cumulative representation of star error of the NB classifier

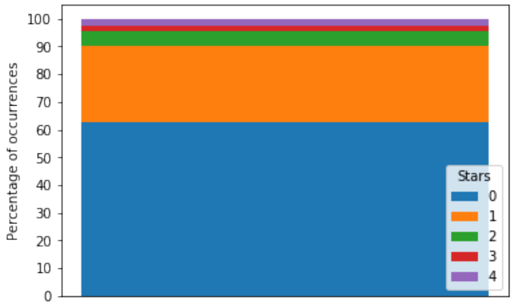

Figure 9: Cumulative representation of star error of the XGBoost classifier

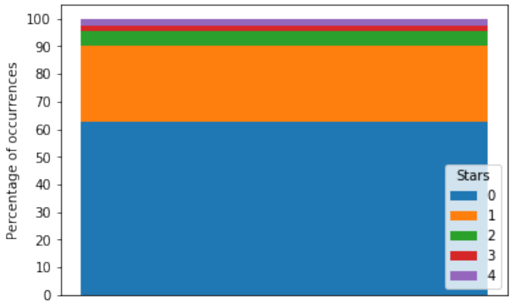

Figure 10: Cumulative representation of star error of the SVM classifier