MiniProject 2: COMP 551

Rebecca Jaubert: 260720767, Mohamad Ataya 260991767, Roy Eyono: 260964877

February 2021

Abstract

Textual data is an example of unstructured data, which poses a challenge for extracting relevant information accurately and timely. In this paper, we use two machine learning models, namely naive bayes and logistic regression to classify textual data from a news and a movie review dataset. The optimization of the hyperparameter of these models is done using 5-fold cross validation. A multiclass classification was used for the newsgroup dataset while a binary classification for the IMDB movie review dataset. Using the multinomial naive bayes model we report an accuracy of 85% and 68 % for the IMDB and 20news group datasets respectively. These values were compared to our implementation of the logistic regression which showed an accuracy of, respectively, 87 % and 66 %.

1 Introduction

In this project, we set out to classify textual data using two models, namely naive bayes and logistic regression. Textual data is often characterized as unstructured data given the difficulty in extracting information from raw text. The implementation of machine learning models can help users to automatically structure texts from a variety of sources such as e-mails, surveys and social media. Classification of such data can allow for fast sorting and organization, and aid in answering questions such as how many people prefer a specific genre of movies. Furthermore, ML can offer real-time analysis of text and has importance in analysing crisis and potential threats. Finally, the accuracy of text classification using ML has a consistent criteria unlike human annotators who are more likely to subjective or fail to hold a consistent stand throughout.

Several research groups have approached the development of these models for similar datasets [1, 2]. In our report, we will look into the implementation of these models on the 20 newsgroups text dataset provided by scikit-learn [3], and the large movie review benchmark dataset available at Stanford [4]. The naive bayes model is based on the bayes theorem developed by Thomas Bayes in 1763 [5]. It is a probabilistic classifier with an implied independence between features. The logistic regression model, on the other hand, is a type of predictive analysis utilized when the dependant variable is binary, initially developed by Joseph Berkson in 1944 [6]. The performance of each model on each dataset

will be evaluated based on the accuracy prediction metric. The tuning of the hyper parameters will be conducted using K-fold cross validation. This tuning will allow us to evaluate whether our model can be generalised to independent datasets. Given that our goal is to predict specific classification, cross validation will help in estimating how accurate our predictive model is in practise. For the purpose of this report, 5-fold cross validation will be used.

2 Preprocessing and Data Analysis

2.1 IMDB Dataset

The dataset contains roughly 25000 reviews from the IMDB dataset, where each review is labelled to be either have a positive or negative sentiment. The dataset is a benchmark for sentiment analysis. In our code, we make use of the Tensorflow library to load the IMDB reviews and then subsequently reconvert them back to text format. Given the text format, we convert the data into a bag-of-words format, while also removing some key english stop words.

2.2 Newsgroup 20 Dataset

The newsgroup dataset is a collection of news dataset labelled across 20 categories. In our code, we leverage the sci-kit learn dataset to download the newsgroup dataset. Like the IMDB dataset, we remove English stop words and simultaneously convert the data into a bag-of-words format.

3 Methodology

3.1 Naive Bayes

In our implementation of Naive Bayes, we applied the multinomial Naive Bayes algorithn for both the IMDB Dataset and the Newsgroup Dataset. To construct our likelihood values for each word, we count the unique occurrences of the given word per class and divide by the total number of words in the given class. To compensate for words of count "zero", we add some smoothing by adding one to each word per class and subsequently add the total number of words per class by the total vocabulary size of the dataset. For the prior of each class, we simply count the total number of instances labelled with the respective class and divide that by the number of documents.

Given a document, in order to calculate the posterior of each class, we multiply all the likelihood values for each word in the document including the prior of the class and take the log of the product resulting in the log-likelihood. The class with maximum posterior is the prediction of our model. For a given word in a document that is not found in our training vocabulary, we assign the word a likelihood value of $1/|V|$, where $|V|$ is our vocabulary size.

In our results, we evaluate our model with 5-fold Cross Validation against logistic regression.

3.2 Logistic Regression

To compare against our Naive Bayes classifier, we implemented a logistic regression classifier with Stochastic Gradient Descent. To find the optimal hyperparameters for logistic regression, we leverage the Random Search algorithm. We evaluate our logistic regression classifier with 5-fold cross validation.

Our search space for the Random Search Algorithm consists of the learning rate, learning rate scheduler and the regularization term α. We implemented our classifier on the Scikit-learn python library.

4 Results

We initially run experiments to compare the performance of our naive bayes classifier across different feature vector lengths. In our feature vector, we select the top N most frequent words (features) in our dataset excluding stop words.

Feature Vector (N)	IMDB	Newsgroup 20
50	69%	22%
250	78%	38%
1000	82%	53%
10000	**85%**	67%
20000	84%	**68%**
|V|	83%	66%

Table 1: Comparing top word count feature vector length for naive bayes across the IMDB and Newsgroup dataset. (5-fold CV)

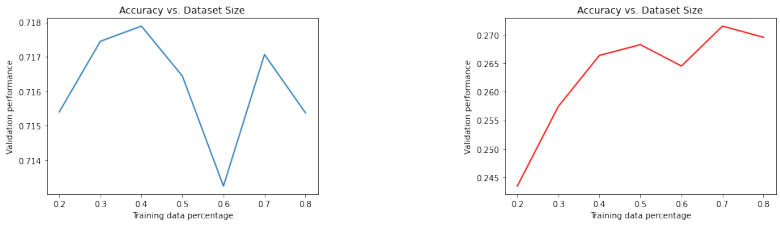
In Table 1, we report the best performing feature vector sizes across the IMDB and Newsgroup dataset. We show that the IMDB dataset is optimal with a top feature vector of size 10000 and an accuracy of 85%, while the Newsgroup dataset is optimal with a feature vector size of 20000 and an accuracy of 68%. We evaluate each feature vector with 5-fold cross validation.

	IMDB	Newsgroup 20
Logistic Regression	**87%**	66%
Naive Bayes (5-fold) (**ours**)	85%	**68%**

Table 2: Evaluating logistic regression classifier on IMDB and Newsgroup datasets against our naive bayes classifier. (5-fold CV)

In Table 2, we compare the optimal naive bayes classifiers from Table 1 with an optimal logistic regression classifier. The resulting configuration with random search for the IMDB dataset is {'learning_rate': 'adaptive', 'η': 100, 'α': 0.001}, while the resulting configuration for the Newsgroup dataset is {'learning_rate': 'constant', 'η': 10, 'α': 0.1}. Their respective results are 87% and 66% with 5-fold cross validation. For the logistic regression, we used a feature vector of 10000 and 20000 respectively for the IMDB and Newsgroup datasets. Further details of the implementation can be found in the code available.

For the IMDB dataset, we observed that the logistic regression classifier (87%) outperforms our Naive Bayes classifier (85%), while for the Newsgroup dataset, our Naive Bayes classifier has improved performances as compared to logistic regression.



(a) IMDB (b) Newsgroup 20

Figure 1: A comparison between the accuracy and the training data percentage for the IMDB and Newsgroup dataset. We train our naive bayes classifier with a feature vector size of 100.

In Figure 1, we conduct a survey comparing across the varying percentage of training data used in our naive bayes model on IMDB and Newsgroup. We observe that IMDB performs well with 40% of training data, and 60% validation. While Newsgroup performs well with 70% training data.

5 Discussion & Conclusion

In our results, we observed the importance of feature vector size for the naive bayes classifier. Although the model underperformed on the IMDB dataset against our benchmark logistic regression model, we observed improved performances on the Newsgroup dataset with 5-fold cross validation. The learning mechanism is different for each of the implemented models. The naive bayes classifier is a generative model, predicting the posterior probability by modeling the joint distribution of features and targets. On the other hand, logistic regression is a discriminative model which predicts posterior probability by learning through mapping the input to output while minimizing the error. Given the class independence assumption of the naive bayes model, classification of text data, where there is a larger number of features compared to sample size, becomes feasible. It is important to note that, while the independence assumption

is helpful, however, it is rarely the case in real life to have completely independent features. In this report we used logistic regression as a comparative model, as such it is important to understand the underlying advantages and disadvantages of this model. While it is easier to implement and interpret, and can be extended to multiple classes, it suffers from a major limitation. Logistic regression assumes a linear relation between the variables, as a result, when faced with a non-linear dataset, it becomes difficult to understand. Finally, logistic regression is more likely to over-fit in highly dimensional data, therefore considering L1 or L2 regularization could help improve its performance.

Future work could be to evaluate feature vector size across multiple other datasets beyond the IMDB and Newsgroup datasets. Other text classification models can be implemented for comparison. Such models include support vector machines, or the utilization of deep learning algorithms.

6 Statement of Contribution

Initially, all group members developed their own code for implementing the models, including the preprocessing step of the data. The initial attempt to run the processed data and start predicting results was started by Roy and further developed by Rebecca. The final resultant code was reviewed by all members. The inception of the report was a collective product of the entire group, which included an initial draft, editing and reviewing the pdf.

References

[1] Sourav Kunal, Arijit Saha, Aman Varma, and Vivek Tiwari. Textual dissection of live twitter reviews using naive bayes. *Procedia computer science*, 132:307–313, 2018.

[2] Fredrik Johansson. Supervised classification of twitter accounts based on textual content of tweets. In *CLEF (Working Notes)*, 2019.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

[4] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P11-1015`.

[5] Thomas Bayes. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. *Philosophical transactions of the Royal Society of London*, (53):370–418, 1763.

[6] Jan Salomon Cramer. The origins of logistic regression. 2002.