COMP551 Assignment 2 Group 16

Rohan Agarwal, Eric Chao, Meilin Lyu

November 2022

1 Abstract

In this paper we examine the performance of Logistic Regression on two benchmark datasets for text data classification tasks. The first task consisted of performing sentiment analysis to classify documents into positive and negative movie reviews. The second task trained a model to predict class probabilities for a news article belonging to any of four possible categories. In both settings we ran a standard K-Nearest Neighbors classifier for comparison. Binary Logistic Regression was shown to be the better model based on ROC-AUC score, but as data set size grows, KNN achieves a higher accuracy. Multi-class logistic regression achieved a better prediction accuracy than KNN on various choices of training data size.

2 Introduction

We compared the classification accuracy of the Logistic Regression classification algorithm with the accuracy of the non-parametric KNN classifer on real-world textual data. Our datasets of choice are the IMDB dataset [1] and 20 News Groups dataset [3], both employed using a bag-of-words (BoW) model. The IMDB dataset contains two target labels - positive and negative reviews. Our experiment using Binary Logistic Regression produced a classification accuracy of 84.9% and an AUROC score of 0.92, which outperforms KNN on the same data set(AUROC 0.85), suggesting that Logistic Regression is a better model for this specific task, regardless of threshold. Previous work with the same data set by Naeem et al. found that Support Vector Machines with TF-IDF features achieved an accuracy of 92% on the IMDB Reviews dataset [2]. From the 20 News Group dataset, we choose 4 unique evenly distributed categories - comp.graphics, rec.sport.hockey, sci.med, soc.religion.christian - out of the 20 categories available as our target labels. Our experiment using Multiclass Logistic Regression produced a testing accuracy of 71.2%, outperforming the 62.2% classification accuracy of KNN. Previous work by Thangairulappan and Kanagavel using Neural Network Classifier with this data have achieved a peak accuracy of 98.14% [4].

3 Datasets

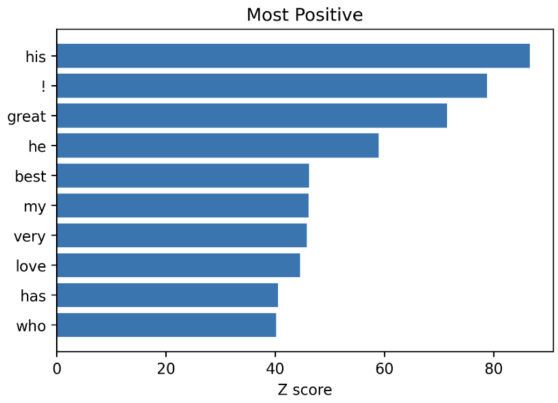
1. **IMDB Reviews**: a set of 50,000 film reviews, split half and half into training and testing sets. We use a BoW representation where each record consists of word counts from a fixed vocabulary of 89527 words and is labeled by its movie rating on a scale from 1 to 10. Our preprocessing work consisted of selecting the set of words that provide high predictive power for sentiment classification. First, we filtered out stopwords and rare words based on a minimum (less than 1%) and a maximum (more than 50%) number of appearances across the data set. Next, we computed the z-score of each variable against a signed version of the target variable, where ratings lower than 5 where mapped into a scale from -1 to -5, and otherwise mapped into +1 to +5. We selected the 250 words with the greatest absolute value of z-score for training the model. Finally, we relabeled the observations into positive (1) or negative (0) sentiment based on reviews rated greater than 5. Class distribution is perfectly split between the two categories, 12,500 observations each.

2. **20 News groups**: a set of 18,000 newsgroup posts on 20 different topics, split into training and testing data sets based on their posted dates. For multi-class classification, we also utilised a bag of words representation. However, the same was not presented to us by the data repository. We had to transform each document using sklearn's CountVectorizer. In order to filter out stopwords (appearing in more than 50% of documents) and uncommon words (appearing in less than 1% of documents), we created a bag of words on the entire data set and filtered out feature words which did not fit the above criterion. Next, we needed to reduce our feature list by only considering the top 10 values using mutual information for each category. The target list had to be one-hot encoded using Label Binarizer. Then we updated our bag of words using a vocabulary of 40 feature words. This updated bag of words which had a shape of (num of documents, 40) was utilised in our implementation and testing.

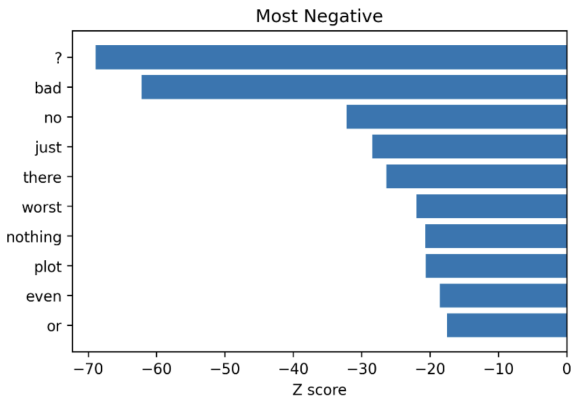
4 Results

4.1 Binary Classification - IMDB Reviews

During feature selection we found that the set of words most correlated with a positive label included "great", "best", "love", "very". The words correlated with a negative label include "worst", "nothing", "bad" and "?", the lattercould be related to users who pose questions challenging the writing or development of a movie.

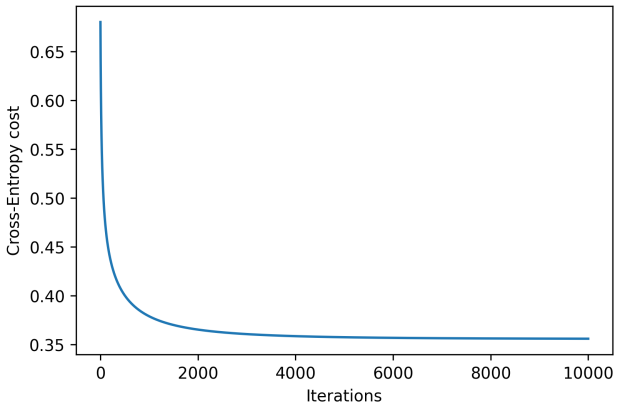


(a) Positively correlated words



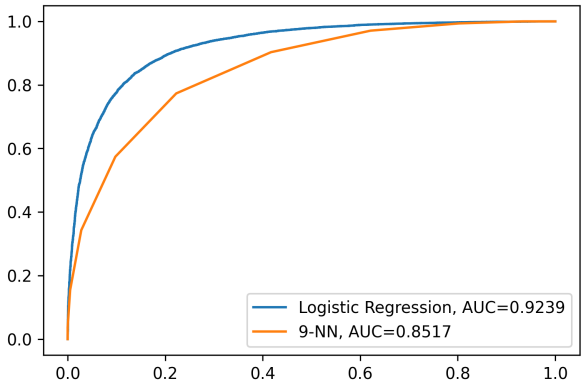
(b) Negatively correlated words

Training our Logistic Regression implementation with a 0.1 learning rate results in a relatively quick convergence of CE cost to 0.35. Training after 1000 iterations achieved an accuracy of 84.9% on the test set with 25,000 reviews.

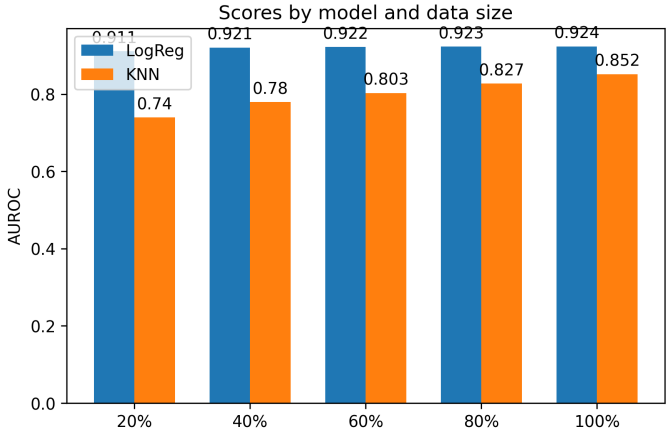


In parallel, we trained a 9-Nearest Neighbors model after picking an appropriate K with a held-out validation set from the training data. The ROC score indicates the quality of our classifier across

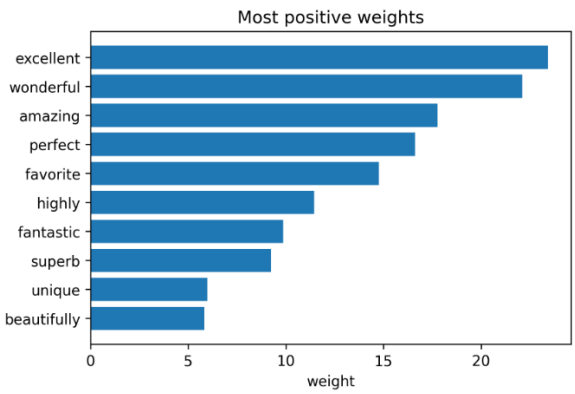
varying thresholds. The AUC score obtained from the predicted class probabilities suggests Logistic Regression is a better model when compared against 9-NN.



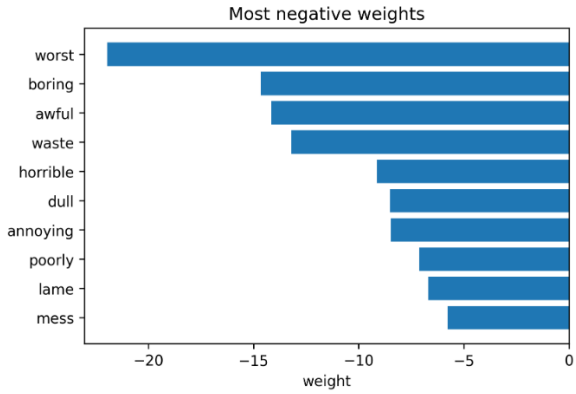
To visualize the effect of data size on performance, we trained and computed the AUROC score on the IMDB data set by using a varying percentage of the original 25,000 instances for training. As the size of data set increases, the performance of both models increases, but it is shown that Logistic Regression is overall a better model than 9-NN, since the score is higher in all instances.



Further, we present the most important features for our logistic classifier model as a list of words having the largest coefficient positive and negative values. We want to note that this list of words is much more explainable than the initial lists we obtained from simple z-score computation. This experiment shows that our method was capable of identifying keywords that indicate the sentiment of the movie review, such as "excellent", "beautifully", or "boring" and "awful".



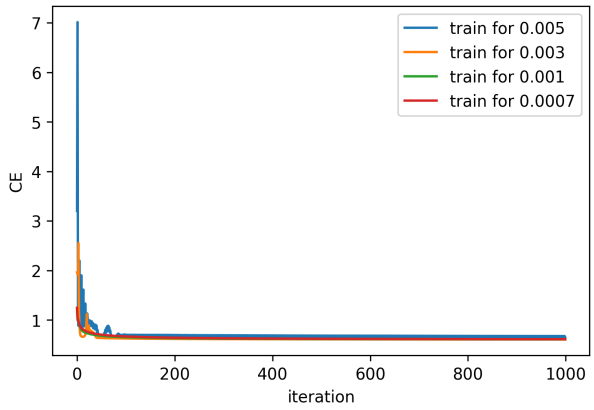
(a) Accuracy over different tree depths



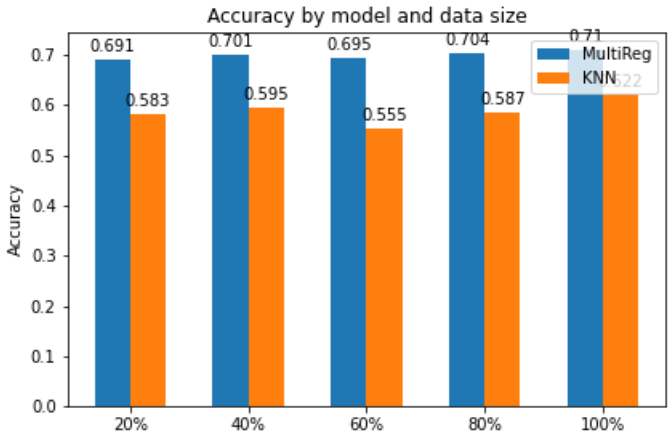
(b) Decision Boundary

4.2 Multiclass Classification - 20 News groups

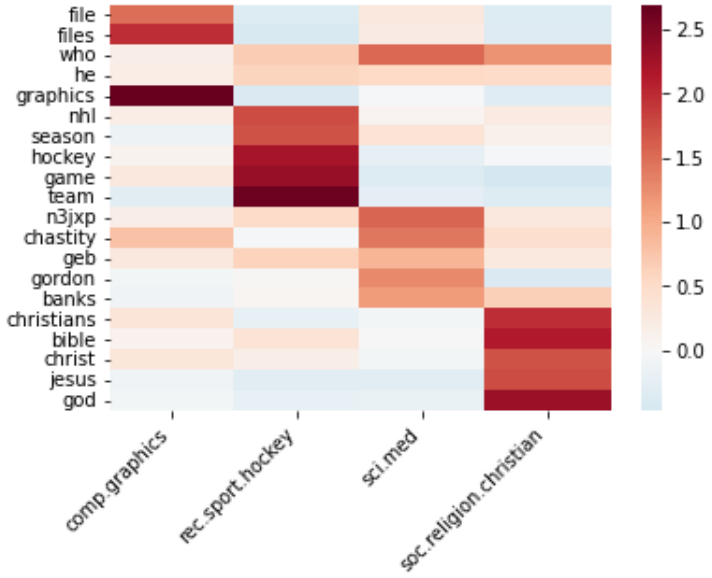
To verify our pre-processing and model implementation, we test our multiclass regression accuracy results against sklearn's implementation. With 10,000 iterations, a learning rate of 0.001 and four categories, we found that our model produced an accuracy of 71.2% while sklearn's implementation returned an accuracy of 71.4%. This gave us the confidence that our implementation and data processing steps were working as expected.
Here we have plotted different error training loss functions with varying learning rates.



As expected of us, we also examined the accuracy on the testing data set as a function of the training set size. Similar to our logistic regression, we ran this test from 20% to 100% with 20% increments. It can be observed that there is a minor dip (0.5%) in the accuracy between 40% and 60% of the data set size. This can be attributed to the randomness behind choosing a weight and the training data set split causing numerical discrepancies. Also, it is almost negligible due to its size factor being less than 1%. The overall trend of the graph shows us a positive rate.



We have represented the top 5 most positive features (words) for each of the four category, a total of 20 features in the form of a heatmap. It can be observed from the heatmap the high correlation between the five features and their respective news group, which correspond to their high mutual information scores with the new group labels.



In both of our classification models, we checked our analytically computed gradients with the numerical results to validate the accuracy of our calculations.

5 Discussion and Conclusion

Logistic Regression is a suitable model for classifying documents with textual data and using a straightforward representation like the Bag-of-Words model. Our experiments showed that Logistic Regression achieved a an AUROC score of 0.92 with as little as 5000 observations and is able to extract interpretable sets of features from the data. When fitting multiclass regression model on the 4 categories from the 20 news group dataset, a learning rate of 0.005 resulted in oscillation in the loss function using full-batch gradient descent. This is due using SSE as a loss function, which produced high gradients. After we reduced the learning rate to 0.001, we reduced the numerical instability. Our multiclass regression model produced an accuracy comparable to that of the scikit-learn regression model. Future researches should experiment using TF-IDF and other feature engineering methods for textual data to test whether this improve data quality for classification models.

6 Statement of Contributions

1. Meilin - Muticlass model training, plot, written report
2. Eric - IMDB preprocessing, Binary Classification, plots
3. Rohan - Multiclass model implementation, training, plots and report

Bibliography

[1] Andrew L. Maas et al. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (June 2011).

[2] Muhammad Zaid Naeem et al. "Classification of movie reviews using term frequency-inverse document frequency and optimized machine learning algorithms". In: *PeerJ Computer Science* 8 (Mar. 2022).

[3] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[4] Kathirvalavakumar Thangairulappan and Aruna Devi Kanagavel. "Improved Term Weighting Technique for Automatic Web Page Classification". In: *Journal of Intelligent Learning Systems and Applications* 8 (2016), pp. 63–76.