Logistic and Multiclass Regression on IMBD Movie Reviews and 20 News Group Data

Cecilia Jiang, David Kronish and Matt Ludwig

1 Abstract

In this assignment we constructed and investigated the performance of two machine learning models on two textual datasets. We found that the logistic regression model achieved better accuracy than the K - Nearest Neighbor (kNN) model for predicting whether movie reviews were positive or negative and that the logistic regression model achieved better accuracy than the kNN model for predicting the topic of message from the 20-newsgroup dataset. We found that our logistic model achieved accuracy of 84.34% on the IMBD testing dataset which performed better than kNN which achieved an accuracy of 72.95% on the IMBD testing dataset. We also found that our multiclass logistic model achieved 76.2% accuracy on the testing data which performed significantly better than kNN which had a correct classification accuracy of 39% on the 20-news group dataset. Finally, in order to compare the accuracies of our models with Scikitlearn's implementations, we used an ensemble method from Scikitlearn called VotingClassifier which made predictions on majority vote of scikitlearn's implementation of k-Nearest Neighbors, Random Forest, and Logistic Regression. We found that the ensemble method had an accuracy of 83.26% on the IMBD movie set data and tried to implement the the ensemble method for the 20-newsgroup dataset but ran into computational issues with the implementation.

2 Introduction

In this assignment we were tasked with preprocessing and conducting exploratory analysis of two different text-based datasets, constructing and implementing logistic and multiclass regression models, evaluating these models under different features and parameters and with comparing their performance them with the k - Nearest Neighbors model. The first dataset consisted of $50,000$ highly polar movie reviews split equally into training and testing sets. It was compiled by Andrew Maas et al. and first used in their paper "Learning Word Vectors for Sentiment Analysis." [MDP+11] In the dataset, no more than 30 reviews are allowed for a single movie as reviews for the same movie tend to have correlated ratings. A movie review is classified as negative if it has a rating less than or equal to four and positive if it has a rating greater than or equal to seven. These reviews with ratings 5 and 6 are deleted. The test and train dataframe both contain 89527 features and 25,000 samples.

The second data we used was the $20-$newsgroup dataset which consists of 18828 newsgroups posts on 20 topics split in two subsets: one for training and the other one for testing where the split between the train and test set is based upon a messages posted before and after a specific date. Each newsgroup post is associated with a specific newsgroup and topic and contains a document id number, subject and sender information. Each newsgroup post had a different topic and the topics ranged from computer graphics, automobiles and sports to politics, religion and outer space. This dataset was originally collected by Ken Lang in 1995 and used to train a netnews-filtering system, but has since then become a very popular dataset for training machine learning models for text classification and text clustering. [Lan95] Rather than training our multiclass regression model on all 20 of the newsgroup topics, we choose our favorite four topics: Computer graphics, Motorcycles, Hockey and Space. These topics represent a subset of the original dataset and the number of documents.

3 Datasets

After loading the bag of word documents from the training and testing folders and the corresponding vocabular set, we removed all stop and rare words by taking out any word that appeared in more than 50% and less than 1% of the documents. Then we constructed testing and training dataframes with all of these words and all of the documents. After this we used Scikitlearn's StandardScaler implementation to standarize the data. Then we used Scikitlearn's implementation of linear regression for feature selection. After fitting the model with the training data we calculated the z-scores for each of the features. These z-scores corresponded to the statistical test that is used to determine if a certain feature is significant in predicting the outcome in a linear regression model. We then found the features with the 10 most positive and 10 most negative z-scores and saw that the corresponding words made sense in predicting whether a movie review was positive or

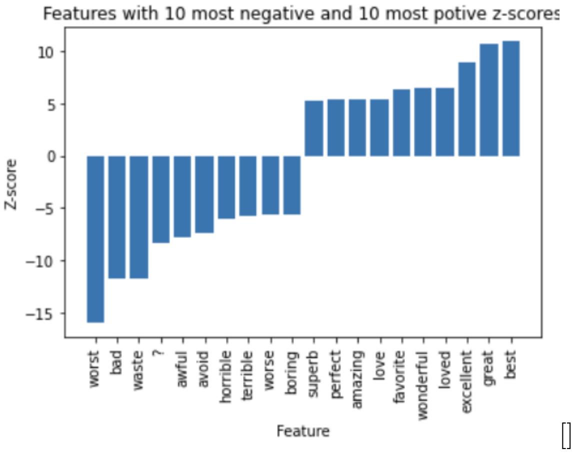


Figure 1: A bar plot showing the top 20 features (10 most positive and 10 most negative based on absolute Z-scores from Simple linear regression on the IMDB data

negative. A barplot of the these features and z-scores is shown below.Our final dataset used to train the model used the 150 features with the largest absolute z-scores.

After loading the 20-news group dataset from scikitlearn, we selected a subset of the data corresponding to our four selected topics and then used scikitlearn's CountVectorize to convert the testing and training set of text documents into two sparse matrices of token counts where each row corresponded to a text document and each column to a word. Since we did not provide an apriori dictionary, we used the dictionary of words generated by CountVectorizer. Next we filtered out rare and stop words by removing any word in the matrix that occurred in less than 1% or more than 50% of the documents. After this, we selected the top 100 features involved in the prediction of each topic using Scikitlearn's implementation of Mutual Information in order to train our multiclass regression model. For more information on mutual information see [PVG+11]. We also produced a heatmap of the top 5 most important features that were used to predict each topic.

4 Results

In order to better understand how are models achieved their given accuracies, we generated both another bar plot of the coefficients from the logistic regression model of the top 20 features used in the logistic regression model and a heatmap of the top 5 most postive features used to predict each of the newsgroup topics. These plots are shown below:

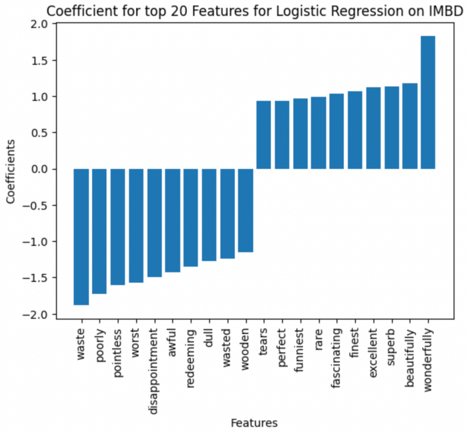


Figure 2: A bar plot showing the top 20 features (10 most positive and 10 most negative) from the logistic regression on the IMDB data

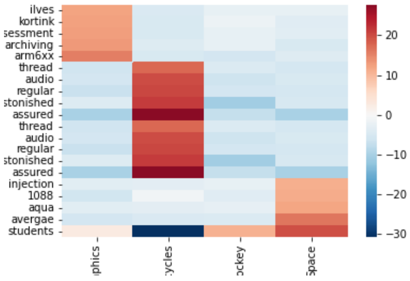
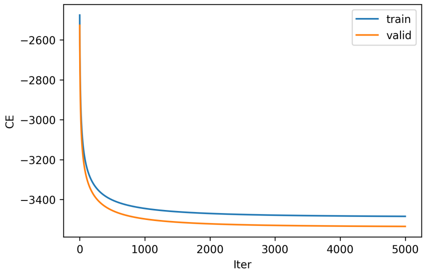


Figure 3: A heatmap showing the top 5 most positive features as rows for each class as columns in the multiclass classification on 4 the chosen classes from the 20-news group datasets.

When implementing our multiclass and logistic regression models, we decided to train the models on the training set using gradient descent. We also implemented functions in both models to compare the numerically and analytically calculated gradients. Using a learning rate of $\alpha = .5$ with 5000 iterations for the logistic regression model and $\alpha = .05$ with 1000 iterations for the multiclass regression model, we generated the convergence plots shown below. These plots show how the cross entropy loss for our models with the training and testing sets decreased with more iterations of gradient descent. We also determined that cross entropy loss was minimized with 60 iterations of gradient descent for the multiclass regression and over 5000 iterations for the logistic regression model.



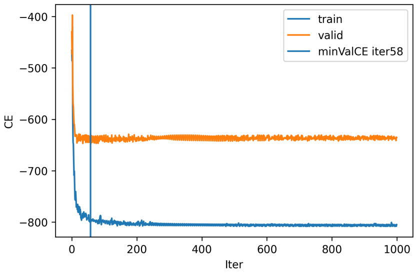
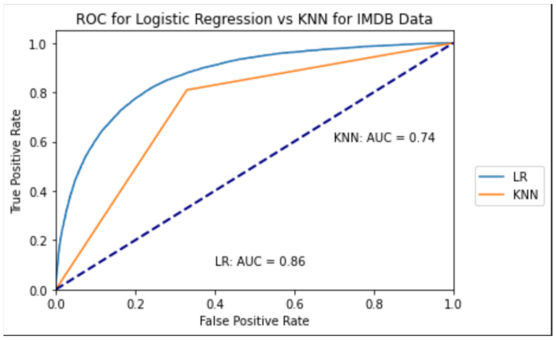


Figure 4: Convergence plot on how the logistic regression model converges given α =.5

Figure 5: Convergence plot on how the multiclass regression model converges given α =.05

In order to evaluate the classification accuracy of our logistic regression model vs. classification accuracy of kNN, we produced a plot of the ROC curves for both models as shown below. We also evaluated our logistic regression model's accuracy against kNN on the testing data as a function of training the model on 20%, 40%, 60%, 80% and 100% of the training data and produced the bar plot shown below. In both cases, our logistic regression model achieved a better AUROC score than scikitlearn's kNN implementation. After splitting the training data into training and testing sets we use Scikitlearn's implementation of grid search to determine the best number of nearest neighbors. For the IMBD we found that $k = 22$ gave the highest accuracy and for the 20-news groups data we found that $k = 1$ achieved the highest accuracy. We also found that as we trained the logistic regression model on different subsets of the training data that there was no significant difference in the AUROC scores of the model for different subsets of the training data.



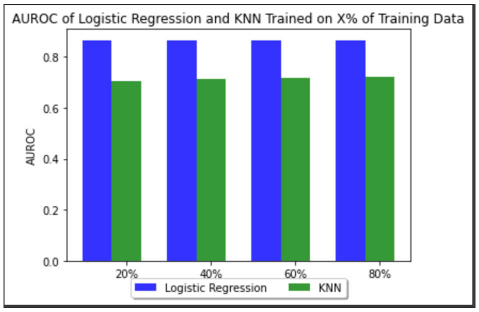


Figure 6: A single plot containing two ROC curves of logistic regression and sklearn-KNN on the IMDB test data.

Figure 7: A bar plot that shows the classification accuracies of logistic regression and KNN on the test data (y-axis) as a function of the 20(x-axis)

Since we were unable to use ROC curves for evaluating the multiclass model, due to the fact that some predictions could not simply be classified as true positive, false positive, etc... we used Scikitlearn to generate a multilabel confusion matrix where the accuracy of each topic was computed using a one-vs-rest scheme. The four multilabel confusion matrices for the prediction of each topic are shown below in Figure 8. In addition, we also evaluated our multiclass regression model's accuracy against kNN on the testing data as a function of training the model on 20%, 40%, 60%, 80% and 100% of the training data and produced the bar plot shown below in Figure 9.

Finally, in an effort to compare the accuracies of our logistic and multiclass models, we implemented an ensemble method from Scikitlearn called VotingClassifier which made predictions on majority vote of scikitlearn's implementation of k-Nearest Neighbors, Random Forest, and Logistic Regression. We found that the ensemble method had an accuracy of 83.26% on the IMBD movie set data and experienced computational issues when computing the accuracy for the 20-newsgroup dataset.

5 Discussion and Conclusion

In this assignment we found that the logistic regression model achieved better accuracy than the kNN model on the IMBD dataset and that the multiclass logistic regression model achieved better accuracy than the kNN model on the 20−news group dataset. We also found that training these models on different subsets of the training data did not substantially change their accuracies. Overall, we found that our logistic model achieved accuracy of 84.34% on the IMBD testing dataset which performed better than kNN which achieved an accuracy of 72.95% on the IMBD testing dataset. We also found that our

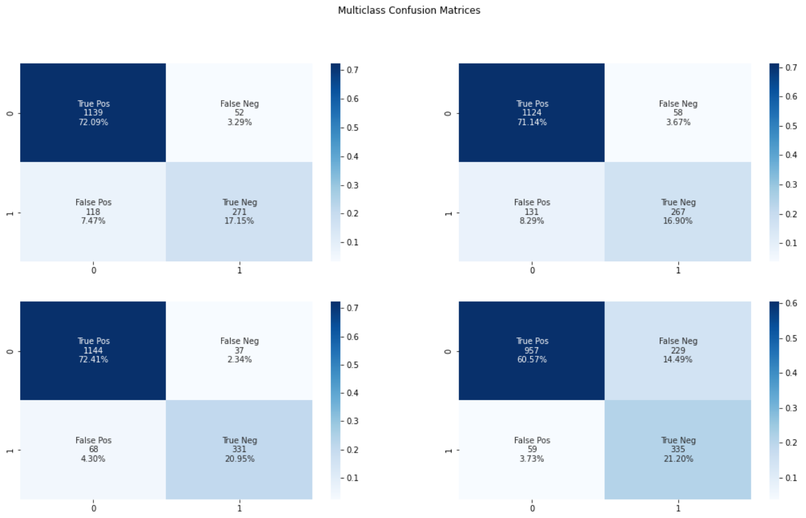


Figure 8: Multilabel Confusion Matrices for Multiclass Regression Predictions

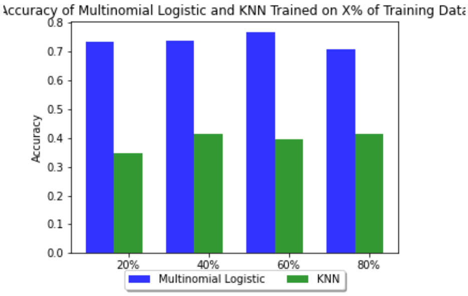


Figure 9: A bar plot that shows the classification accuracies of multiclass regression and KNN on the test data (y-axis) as a function of the 20(x-axis)

multiclass logistic model achieved 76.2% accuracy on the testing data which performed significantly better than kNN which had a correct classification accuracy of 39% on the 20-news group dataset. Finally, in order to compare the accuracies of our models with Scikitlearn's implementations, we used an ensemble method from Scikitlearn called VotingClassifier which made predictions on majority vote of scikitlearn's implementation of k-Nearest Neighbors, Random Forest, and Logistic Regression. We found that the ensemble method had an accuracy of 83.26% on the IMBD movie set data and tried to implement the the ensemble method for the 20-newsgroup dataset but ran into computational issues with the implementation.

Further investigations could consider using different preprocessing techniques in order to achieve a higher accuracy. One preprocessing technique that further work could explore is lemmatisation. Implementing this technique would involve grouping together words with similar endings. For instance, using this technique would combine the words ”slept”, ”sleeping” and ”sleep” into the word ”sleep”. This would help reduce computation time and improve model accuracy. Another preprocessing technique that is also in the NLTK package is parts of speech tagging (POS). This preprocessing technique involves assigning parts of speech (noun, verb, adjective, conjunction etc...) to each word in the text. This could also improve the models' accuracies as it is likely that only nouns and adjectives are relevant for determining the sentiment of a movie review or the topic of a message from the 20-newsgroup dataset. Another preprocessing measure that we could have implemented was spelling correction. When obtaining some of the important predicting words for the IMBD movie dataset, we noticed that some of the words were mispelled. It also seems plausible that movie reviews contained mispelled words and implementing spelling correction might have slightly improved our models, especially if we were running our models with an even larger dataset.

6 Statement of Contributions

Matt: All of the preprocessing for 20-newsgroup dataset (Task 1.2). Implementation of Multiclass classifier (entire task 2 and 2.1 for multiclass, check gradient). Implementation of logistic regression class (Cecilia wrote evaluate method and assisted). Convergence plots for both multiclass and logistic. Bar plot of multiclass vs. kNN on different subsets of training data. Multiclass heatmap for multiclass regression. Implemented confusion matrices for multiclass regression. Researched and implemented ensemble method for creativity. Wrote entire report and initiated vast majority of communication.

Cecilia: All preprocessing for IMBD dataset, assisted with implementation of logistic class, horizontal bar plot showing the top 20 features horizontal in logistic regression and bar plot of top 20 features in linear regression

David: Single plot of ROC curves of logistic and KNN, AUROC bar plot for accuracies of training the logistic model on subsets of training data. Tried to implement check gradient for logistic class. Assisted Cecilia in adressing bugs in preprocessing IMBD data.

References

[Lan95] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

[MDP+11] 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.

[PVG+11] 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.