COMP 551 Assignment 2

Matthew Barg, Jack Kelly, Cleo Norris

February 29, 2024

Abstract

In this project we investigated the performance of two linear classification models, logistic regression and multiclass Regression, on two benchmark datasets, IMDB Reviews and 20 Newsgroups. We compared these models to the performance of Decision Trees (DT), and we found that in general, both logistic and multiclass regression models performed better than Decision Trees. This improvement was more pronounced for the multiclass regression. The AUC for Logistic Regression was 0.71 while the AUC for DT was 0.58 for the IMDB Reviews dataset. The testing accuracy for our multiclass regression was 77.79% while the accuracy for DT was 70.9% for the 20 Newsgroups dataset.

1 Introduction

The goal of this assignment was to familiarize ourselves with two linear classification techniques, logistic and multiclass regression, and compare these algorithms against Decision Trees. To do so, we examined two datasets: IMDB Reviews and 20 Newsgroups. In general, both models performed better than Decision Tree.

In existing literature, the IMDB dataset has been used in benchmarking a vector space model that learns word representations that capture semantic and sentiment information. The methodology included both supervised and unsupervised techniques, and this model provides an alternative to matrix factorization-based techniques. The IMDB dataset has also been used to create a zero-shot classification pipeline using the DeBERTa transformer model. Among other things, this pipeline can be used for natural language inference (NLI) and zero-shot entailment-based classification for arbitrary labels (ZS).

The 20 Newsgroups dataset has been used in topic extraction with non-negative Matrix Factorization and Latent Dirichlet Allocation. The non-negative Matrix Factorization is applied with two different objective functions, one of which is the Frobenius norm. It has also been used as a language network benchmark in testing a proposed Structural Deep Network Embedding method. The methodology included using the tf-idf vectors of each word to represent the document and the cosine similarity as the similarity between two documents.

2 Datasets

Similar to Assignment 1, two datasets were analyzed in for this assignment. Dataset 1 (DS1) is IMDB reviews, which contained highly polarized movie reviews (25,000 each for training and testing). Each review contained an overall score out of 10 and a reference to the number of times certain words appeared in the review. The list called imdb.vocab contains over 89500 unique words (for example: 'what') and symbols (for example, '!'). The vocabulary list acted as features for the data analysis. First, however, to be able to perform logistic regression, the number of features needed to be heavily reduced. Our process of filtering had two parts. To start, we filtered out words that appeared in less than one percent of the documents and words that appeared in more than 50 percent of the documents. This effectively removed commonly used words that do not give relevant information (also known as stop words) and removed rare words. This gave a modified filtered vocabulary list. After this initial filtering, the vocabulary list was narrowed down to 150 words by ranking words by their absolute regression coefficients with rating scores. This process was completed by performing simple linear regression with the movie score as the target variable. In the end, the filtering reduced the number of features by over 99 percent and gave a much more manageable number of features that can be used to complete the analysis.

Dataset 2 (DS2) is 20 Newsgroups, which is a set of 18,846 newsgroup documents (including message board messages) belonging to 20 different topic categories (classes). Before pre-processing, each document contained a header including information about the sender and subject of communication. In order to perform our desired classification, we imported DS2 without this header information as it represented noisy features. We also selected five distinct categories on which to train our multiclass regression model: ‘comp-graphics’, ‘misc.forsale’, ‘rec.sport.hockey’, ‘sci.med’, and ‘talk.politics mideast’. Selecting distinct categories assisted in determining whether our classification was successful by examining the identified top features for each class. We imported the training dataset (and later imported the testing dataset) and performed explorations to understand the data’s attributes. One such attribute is “target”, as in “training_data.target”, which produces the true class of each document and would later become our y vector in the multiclass regression. To perform regression on text data, we converted the data into feature vectors using sklearn’s CountVectorizer method. In doing so, we filtered the dataset based on English stop words, words that appeared in less than one percent of the documents, and words that appeared in more than 50 percent of the documents. Fitting and transforming our training dataset with the CountVectorizer produced a sparse matrix with one row for each document, and one column for each unique word (feature) in the dataset. To further filter the number of features upon which we trained our regression model, we used Mutual Information (MI) to select the top 50 feature words per class, leaving us with a matrix of 250 columns to train our model. This was useful in benchmarking that our training data was filtered accurately; for example, the top 10 words for the ‘rec.sport.hockey’ class were “team, hockey, game, nhl, season, play, players, teams, league, playoffs, rangers”, which seemed reasonable. The testing data was imported similarly and was transformed by the CountVectorizer that was fit to the training data to avoid data leakage.

3 Results

3.1 Top 10 Positive and Negative Coefficients for IMDB Data

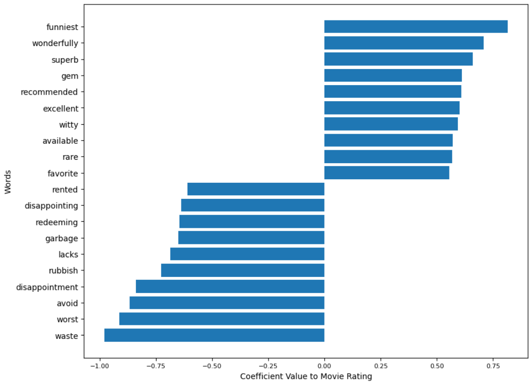

Figure 1: Top Words Related to Movie Rankings for Simple Linear Regression

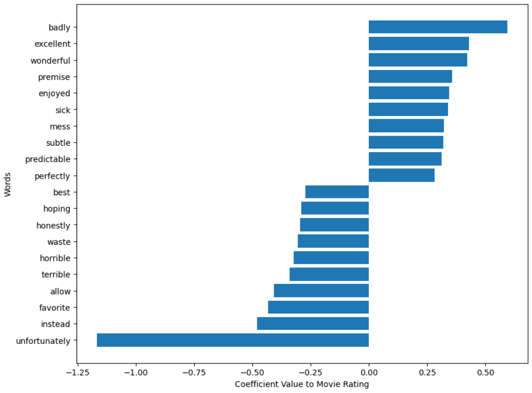

Figure 2: Top Words Related to Movie Rankings for Logistic Regression

Figures 1 and 2 show the top 20 features (10 most positive and 10 most negative) from both the simple linear regression and logistic regression on the IMDB data. Figure 1 was produced after the filtering process explained in the Datasets section. Figure 1’s top positive words consisted of words such as ‘funniest’, ‘wonderfully’, ‘superb’ and negative words ‘disappointing’, ‘garbage’ and ‘disappointment’. These findings make sense in terms of predicted language that would be used in positive and negative movie critiques. Figure 2 showed a similar theme however there were some unexpected differences. A possible reason for this could be the different interpretation of coefficients for logistic regression, since each coefficient represents the rate of change in the log odds of the response variable as the x-variable changes, as opposed to just the rate of change in the response variable.

3.2 Convergence Plots

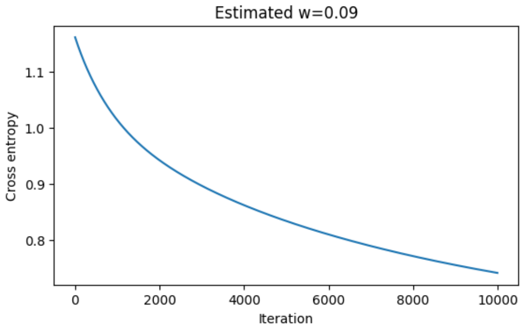

Figure 3: Logistic Regression Convergence

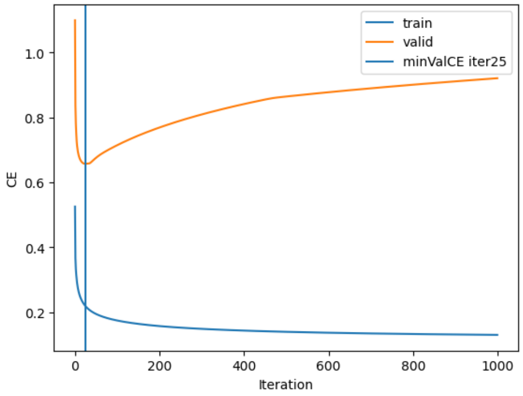

Figure 4: Multiclass Regression Convergence

Figure 3 shows the cross entropy monitored for the logistic regression. Due to computing issues, the iterations were stopped after 10,000 instead of 100,000, but we still see convergence displayed as the cross entropy is minimized due to gradient descent. Figure 4 shows the convergence for multiclass regression, and demonstrates that the optimal number of iterations is just before overfitting begins, as seen by the point at which the validation cross entropy begins to increase.

3.3 ROC Curves and Accuracy Analysis of Logistic Regression and DT for IMDB Data

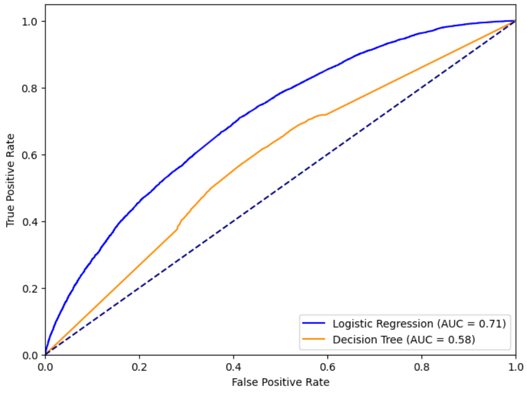

Figure 5: ROC DT vs LR in predicting IMDB Rating

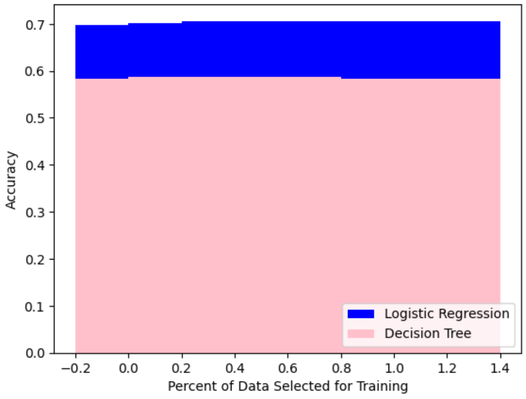

Figure 6: Percent Training Data vs Accuracy for IMDB

When looking at the shape of the ROC curves, a higher accuracy is shown with Logistic Regression (71%) compared to the Decision Tree algorithm (0.58%). This is a significant difference and this gap is supported as well by the accuracy analysis in Figure 6. An interesting observation is the accuracy of both Logistic Regression (LR) and Decision Tree (DT) stays consistent amongst different percentages of training data. LR has an average fluctuation of 0.022 between different percentages of training data while DT has an average fluctuation of 0.00124. This could be because the models are not overfitting to the training data and there is good class distribution amongst all the training sizes.

3.4 Multiclass Classification Accuracy for Multiclass Regression and DT for 20 Newsgroups

For multi-class regression, the model had training, validation and testing accuracy scores of 83.8%, 78.89% and 77.79% respectively with a 50/50 train/validation split. This is an improvement over the Decision Trees test accuracy of 70.9%, as shown in Figure 5. An even higher test accuracy of 82.5% was also achieved by the multiclass regression when an 80/20 train/test split was utilized.

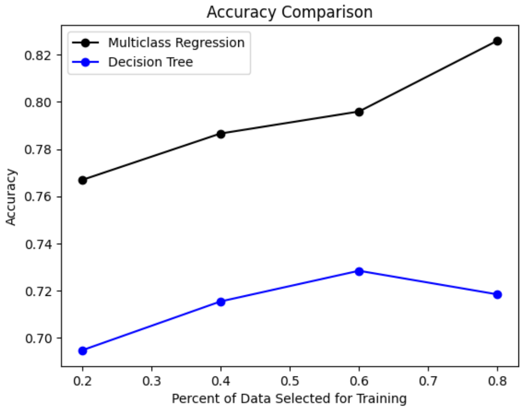

Figure 7: Percent Training Data vs Accuracy for DS2

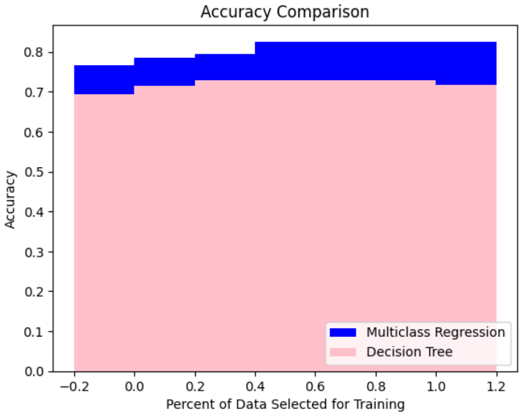

Figure 8: Percent Training Data vs Accuracy for DS2

As shown in Figures 7 and 8, the test accuracy of the multiclass model increased when a higher percentage of data was used for training. The relatively strong prediction accuracy of our model makes sense when examining the heat map in Figure 9.

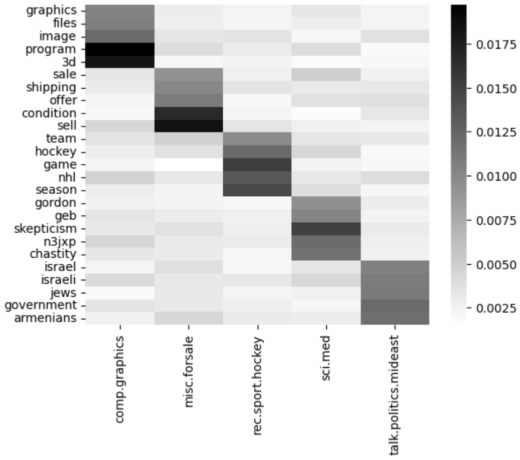

Figure 9: Top 5 Positive Features Heatmap

We can see a clear trend between stronger weights and words that are associated with certain classes, allowing the model to effectively differentiate between them.

3.5 Further Exploration

Multiple forms of a regularized multiclass regression model were tested for 20 Newsgroups. Ridge, LASSO (Least Absolute Shrinkage and Selection Operator) and ElasticNet penalties were all used, with test accuracy scores of 80.5%, 76.5% and 76.6% respectively. As these are all similar to the normal multiclass regression that was implemented, this suggests that the regularization in these models failed to help them generalize better to unseen data. Given that the original multiclass regression did not exhibit signs of overfitting, however, this is reasonable to expect. The use of a validation set to prevent excess iterations, as well as the feature selection and pre-processing done to the data, may be reasons for this. Next, we explored the performance of the multiclass regression model on the complete 20 Newsgroups data set with all 20 classes. For this analysis, a prediction was counted as correct if the true class appeared in the top K predicted classes. Values of K from 1-10 were explored and the corresponding accuracy scores were plotted in Figure 10.

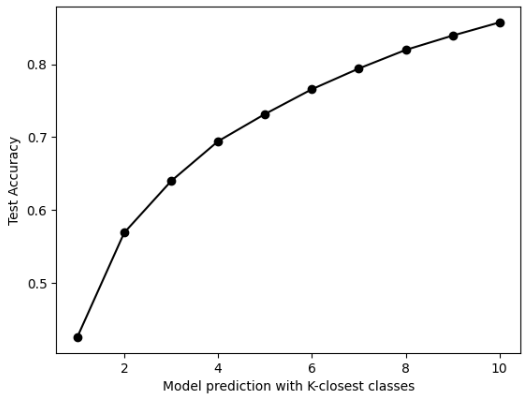

Figure 10: Test Accuracy for Experiment

We can see that the model is quite poor at predicting the class outright, but improves noticeably for small initial increases in K. The larger the K value, however, the less utility the model provides. In the future, it would be interesting to examine if the predictions of classes are grouped by subject matter (e.g. if the correct class is ‘talk.politics.mideast’, the next two might be ‘talk.politics.misc’ and ‘talk.religion.misc’). This would indicate that the model is getting the broader classification right but struggles to distinguish between classes where more granularity is required.

4 Discussion and Conclusion

As aforementioned, the key features produced at the initial simple linear regression filtering for DS1 seem logical, and there were some slight disparities when it came to displaying the top features as identified by logistic regression.

For DS2, as seen in the code output, the top features from each class aligned logically for each newsgroup category. Further, we see that multiclass regression performs consistently better than DT for every level of data selected for training. However, it is important to note that there is a limit to its performance, as seen by its slowing accuracy as number of classes increases.

As a possible area of further investigation, we can consider analyzing more than just individual words for DS1. Context can affect the words used in ratings; for example, using the word ‘gross’ for a horror movie rating might be a positive feature. Accounting for the genre of the movie could be a form of tokenization, and we can also consider common word pairings to predict. For DS2, we can test different learning rates, as the one used in our analysis was chosen on the basis of being commonly used and not using too much computing power.

5 Statement of Contributions

There was an equal contribution from team members across cleaning data, implementing the methods, experimenting, and the write-up.