COMP 551 Assignment 2 Write-up

Faith Ruetas, Joey Marten, Gary Zhang

November 8, 2022

Abstract

We implemented the Logistic Regression and Multi-class Logistic Regression classifiers, then ran our models on the Large Movie Review (IMBD) and 20 News Groups Text (Twenty) datasets, respectively. After pre-processing the data and training the models, we found that Logistic Regression consistently outperformed Multi-class Regression, though both of our models outperformed Sklearn's KNeighborsClassifier.

1 Introduction

In machine learning projects, it is common to rely on publicly available classes such as Sklearn's LogisticRegression and MultiOutputRegressor. For this assignment, by contrast, our task was to implement the Logistic Regression and Multi-class Logistic Regression classifiers ourselves. We used the Large Movie Review (IMDB) dataset to evaluate the Logistic Regression model and the 20 News Groups Text (Twenty) dataset to evaluate the Multi-class Regression model. The first dataset comprises the text and integer ratings of highly polar movie reviews on IMDB; the second comprises newsgroup posts on 20 different topics.

Overall, our Logistic Regression model outperformed our Multi-class Regression model on their respective datasets, though both outperformed Sklearn's KNeighborsClassifier. Logistic Regression had a maximum testing accuracy of 80.62% on the IMDB data compared to 70.22% for Multi-class Regression on the Twenty data and 70.04% for Sklearn's KNeighborsClassifier on the IMDB data.

There is ample prior work on using regression models to classify text-based data. In "Sentiment Analysis using Logistic Regression and Effective Word Score Heuristic," for instance, Tyagi and Sharma use Logistic Regression to categorize Twitter messages as either positive, negative, or neutral using certain keywords [1]. In a similar vein, Madigan et al. in "Bayesian Multinomial Logistic Regression for Author Identification" use Multinomial Logistic (the same as our Multi-class) Regression to classify texts' authors based on writing style [2]. In short, our project follows pre-existing scholarship on Logistic Regression and Multi-class Regression, though with different datasets and slightly different classification aims.

2 Datasets

2.1 IMDB

The IMDB dataset comprises 50,000 highly polar movie reviews split into two equal sections for training and testing. Each review entails text-based content as well as an integer rating (out of 10).

After loading the data into a sparse matrix, we removed all the words that appeared in more than 50% of the reviews (stopwords) and less than 1% of the reviews (rare words). From there, we calculated the z-scores of the remaining words and filtered our dataset to only include the 100 words with the highest absolute z-scores, following the Simple Linear Regression hypothesis.

As expected, the words with the most positive z-scores (those most correlated with a movie's integer rating) were all positive words such as "great," "excellent," and "wonderful." Likewise, the words with the most negative z-scores (the words most negatively correlated with a movie's integer rating) were all negative words such as "bad," "worst," and "waste" (see Fig 1).

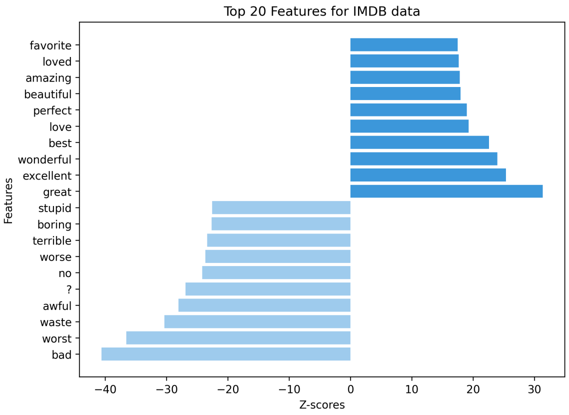

Figure 1

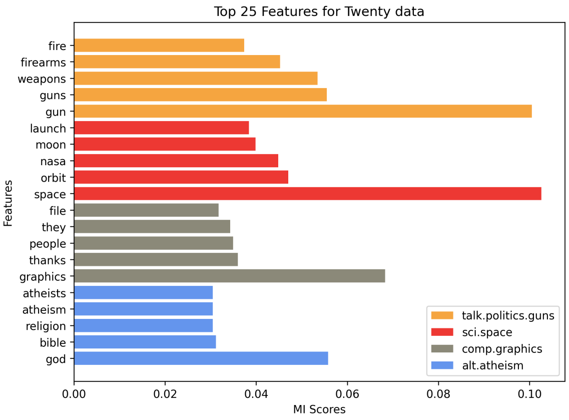

Figure 2

2.2 Twenty

The Twenty dataset comprises 18,000 newsgroup posts on 20 topics split into two equal sections for training and testing. For our purposes, we only took the data from four topics: alt.atheism, comp.graphics, sci.space, and talk.politics.guns.

To pre-process the data, we converted each newsgroup post into a word vector. We once again removed all words appearing in more than 50% of the posts (stopwords) as well as the words appearing in less than 1% of the posts (rare words).

From there, we used Mutual Information (MI) to find the top 25 words per newsgroup topic and filtered our dataset to only include the union of these words (for a total of 88 words).

As expected, the words with the highest MI score for each topic were highly related to that topic, such as ”god” for alt.atheism, ”graphics” for comp.graphics, ”space” for sci.space, and ”gun” for talk.politics.guns (see Fig 2).

2.3 Analysis

To better understand each dataset, we plotted their label distributions. For the IMDB data, the data was equally split into positive and negative reviews (Fig 3). For the Twenty data, there were more instances in the sci.space category and fewer instances in the alt.atheism category (Fig 4).

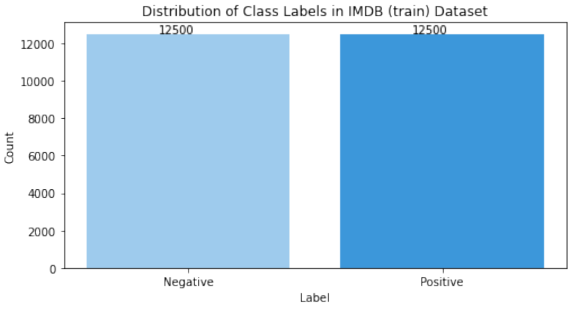

Figure 3

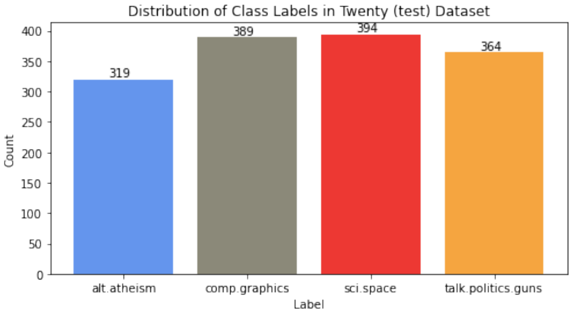

Figure 4

3 Results

3.1 Logistic Regression

Our Logistic Regression model had an 80.89% training accuracy and an 80.62% testing accuracy on the 100-featured IMDB data.

3.1.1 Top 20 features

After fitting our Logistic Regression model to the IMDB training data, we identified the following features with the most positive and negative coefficients (Fig 5).

The features with negative coefficients represent more than half (i.e., six) of the words identified by the Simple Linear Regression Hypothesis (Fig 1). Although this is the case for only three of the features with positive coefficients, there are many semantically similar words between the two groups, such as "perfectly" and "beautifully" in the coefficient-identified words, and "perfect" and "beautiful" in the z-score-identified words.

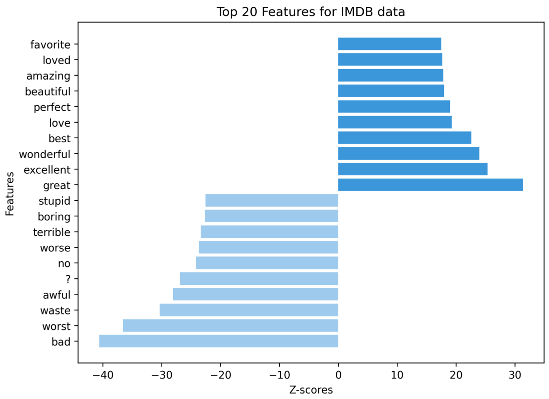

Figure 1: (repeated from page 2)

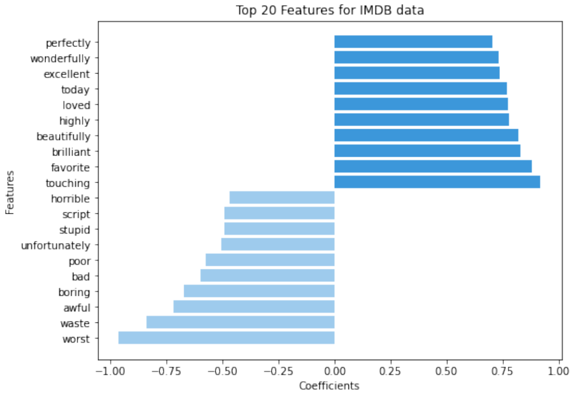

Figure 5

3.1.2 Comparing with KNN

Our Logistic Regression model had a 1.1% higher training accuracy and a 10.58% higher testing accuracy than Sklearn's KNeighborsClassifier when run on the 100-featured IMDB data.

To further investigate our model's performance compared to KNN, we plotted the ROC curve for both (Fig 6) and compared the resulting AUROC values when training the models on different percentages of the IMDB training data. Our model consistently outperformed KNN with a minimum increase of 0.14 AUROC across all tested training sizes (Fig 7).

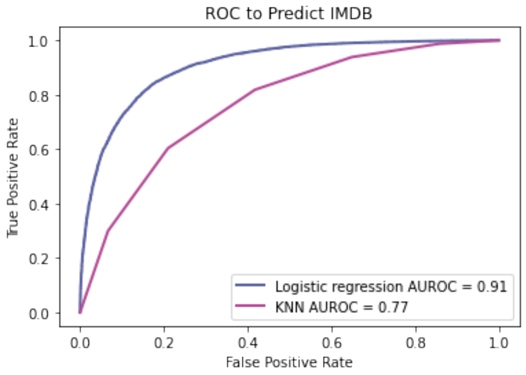

Figure 6

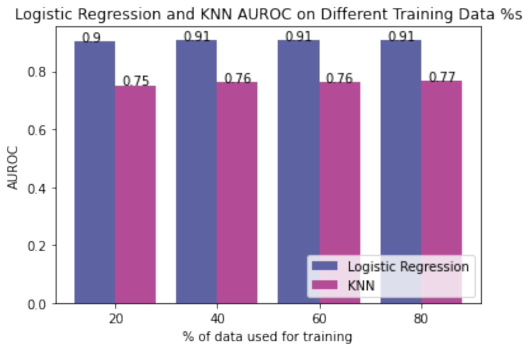

Figure 7

3.2 Multi-class Regression

Our model had a 78.47% training accuracy and a 70.22% testing accuracy on the 88-featured Twenty data (based on the words with the highest MI score for each topic).

3.2.1 Top 20 features

After fitting our Multi-class Regression model to the Twenty training data, we took the five words with the highest coefficients for each of the four categories we chose earlier (Fig 8). As displayed in the heatmap below, these words were both logically and mathematically correlated with their respective categories, such as ”god” with alt.atheism, ”graphics” with comp.graphics, ”space” with sci.space, and ”gun” with talk.politics.guns.

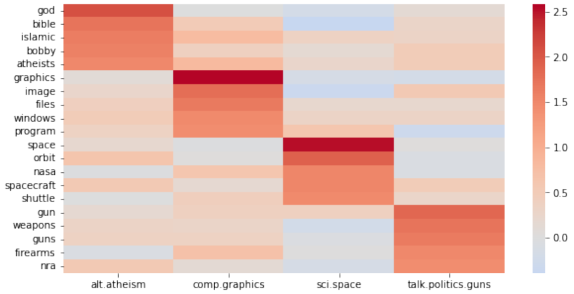

Figure 8

3.2.2 Comparing with KNN

Our Multi-class Regression model had a 2.95% higher training accuracy and a 6.2% higher testing accuracy than Sklearn’s KNeighborsClassifier when run on the Twenty data.

From there, our model continued to outperform KNN when trained on different percentages of the Twenty dataset. The former demonstrated an average classification accuracy of 69.75% compared to the latter’s 62.5% with a minimum 6% difference across all tested training sizes (Fig 9).

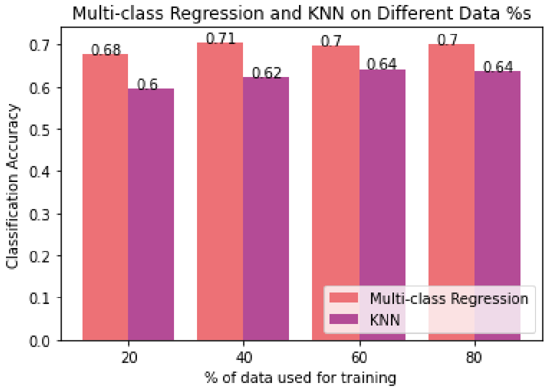

Figure 9

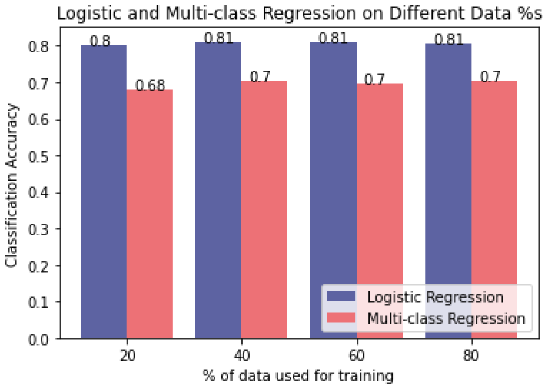

Figure 10

3.3 Comparing Logistic and Multi-class Regression

At its best, our Logistic Regression model was 10.4% more accurate at classifying positive/negative movie reviews from the IMDB testing dataset than was our Multi-class Regression at sorting newsgroups from the Twenty testing dataset into four topics. This may be due to the fact that the latter had to classify data into twice the number classes than the former (Fig 10).

3.4 Convergence

Our Logistic Regression model converges at approximately 0.5 cross-entropy loss around 800 iterations (Fig 11). Our Multi-class Regression model continues to converge to approximately 0.2 cross-entropy loss on the training data (Fig 12). On the validation set, by contrast, the model converges at 0.7 cross-entropy loss after 1270 iterations.

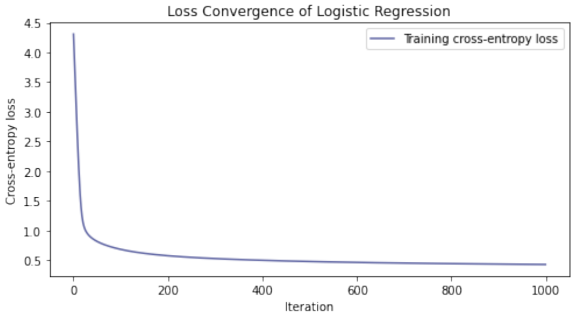

Figure 11

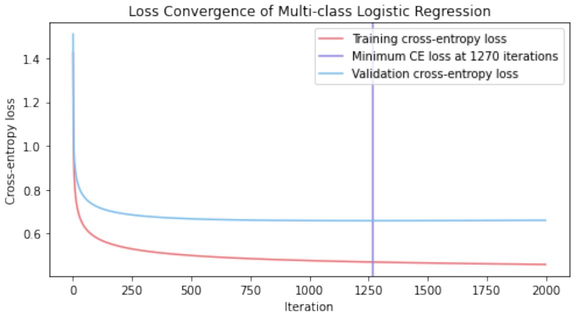

Figure 12

3.5 Comparison to the Sklearn library

The Lasso and Ridge regression models perform on par with each other but significantly worse than our Multi-class Regression implementation, given the same learning rate and maximum iteration limit. Our Multi-class model performs slightly better than Sklearn's implementation, potentially due to the modifications we made to our softmax and cost functions (Fig 13).

We chose to subtract the maximum value of xw in an effort to avoid large exponents. This makes sense because $softmax(x) = softmax(x + c)$ for some constant c. We also modified the cost function to skirt the limitations of floating-point numbers since, occasionally, a predicted y-value was 0; since $log(0)$ is negative infinity, this multiplied by 0 (from y) produces NaN in Python.

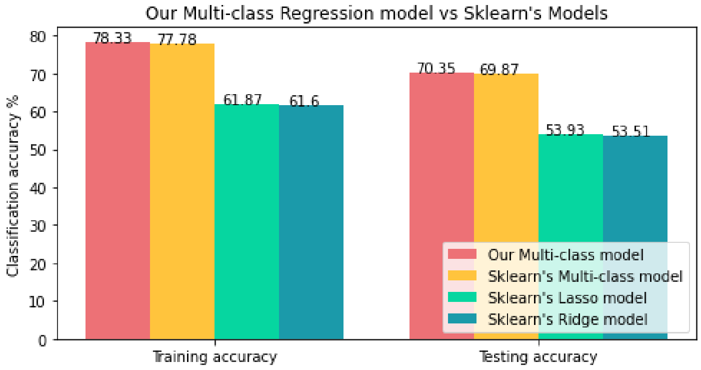

Figure 13

4 Discussion and conclusion

In summation, our Logistic Regression model on the IMDB dataset outperformed our Multi-class Regression model on the Twenty dataset with a maximum testing accuracy of 80.62% compared to 70.22%, respectively. This difference can be explained by the fact that the latter model classified data into twice the number classes than its counterpart. Nevertheless, both algorithms outperformed Sklearn's KNeighborsClassifier on both training and testing data.

In the future, possible directions of exploration include testing different text embedding methods, experimenting with different learning rate and iteration combinations, and using linear regression to predict the integer ratings of the IMDB data. To further investigate model convergence, one might also implement stochastic gradient descent and compare it with standard Multi-class Logistic Regression.

4.1 Statement of contributions

Faith pre-processed the data, encapsulated code, added bar graphs, and merged work on Github; Joey edited the Logistic and Multi-class Regression code and performed experiments; Gary added the MI and heatmap implementations. Everyone performed code quality reviews and edited this write-up.

References

[1] David Madigan et al. “Bayesian Multinomial Logistitc Regression for Author Identification”. In: *AIP Conference Proceedings* 803.509 (2005).

[2] Abhilasha Tyagi and Naresh Sharma. “Sentiment Analysis using Logistic Regression and Effective Word Score Heuristic”. In: *International Journal of Engineering Technology* 7.2.24 (2018), pp. 20–23.