Assignment 2: Binary and Multiclass Classification with Textual Data

Mat Vallejo, Bonaventure Dossou, Minjae Kim

February 25, 2024

Abstract

This paper explores the capabilities of logistic regression and multilinear regression in predicting outcomes for binary data and multi-class data respectively. The logistic regression model is used to predict positive or negative sentiment from IMDB movie reviews, while the multilinear regression is used to match news clippings to their correct news category from the 20 News dataset. This process involves heavy data preparation and cleaning, along with carefully executed feature selection to maximize predictive results. The models are compared against built-in machine learning models from the Scikit-learn machine learning library to test for robustness. We find strong predictive capabilities from both models that regularly outperform Scikit-learn models with no hyperparameter tuning.

1 Introduction

This exploration involves a variety of cascading tasks for both datasets and model implementations. To begin, there is a necessary data preprocessing stage that organizes textual data into a workable format. Notably, the use of the Scikit-learn vectorizer function achieves the goal of indexing word frequency in the 20 News dataset. In contrast, a proprietary word counter is used in the IMDB implementation. Following the preprocessing step is feature selection, done using the union of calculated mutual information in the multilinear regression and using linear regression in the IMDB implementation. This helps us discern which words contribute most heavily to their respective outcomes. Once feature extraction has been accomplished, the models can be implemented using gradient descent to minimize cross-entropy loss, designed to terminate after N number of iterations or once convergence is reached with a validation set. The results are then reported in both numerical and graphical formats.

1.1 Data Preprocessing

To begin, we set up a new function for the IMDB dataset that will allow us to download the dataset and organize the reviews by their index, the general sentiment (positive or negative), and the rating given. Following this we organize the reviews into positive and negative groups within larger groups of training and testing data. This reflects the way the data is organized from the download itself with 50% of the data in the train folder and 50% of the data in the test folder. We use a counter to track word count and filter out rare and stopwords. For the 20 News dataset, the vectorizer function is used to calculate word count followed by the TF-IDF of the features, which is a measure of total frequency against inverse document frequency.

2 Methods

2.1 Linear Regression

We use a Linear Regression function designed to extract features with the greatest contribution to both positive and negative sentiment scores. We fit the training data to the model, make predictions on the test set, and calculate the Mean Squared Error (MSE). We then use the coefficient weights to determine the top 100 features for each class and use a sorting function to report the results. The listed terms contributing to positive/negative sentiments largely make intuitive sense such as “great” and “excellent” for the positive sentiment, with terms like “worst” and “waste” weighing highly for negative sentiment. Below are the Top 10 features contributing respectively to positive review and negative reviews:

Positive Review

No.	Feature	Coefficient
1	great	0.104445
2	excellent	0.099427
3	best	0.093569
4	perfect	0.072433
5	loved	0.071704
6	wonderful	0.071465
7	enjoyed	0.069760
8	favorite	0.066854
9	well	0.064684
10	love	0.063270

Negative Review

No.	Feature	Coefficient
1	worst	-0.159869
2	waste	-0.115960
3	bad	-0.109012
4	poor	-0.079165
5	nothing	-0.076872
6	awful	-0.075401
7	boring	-0.073486
8	supposed	-0.069068
9	poorly	-0.066930
10	bad.	-0.061575

2.2 Multiclass Regression

The features (words) selection process for the 20 News dataset is comprised of the following: deleting all newline characters from strings, removing digits, removing punctuations, splitting split into words to remove spaces, removing stop words, and removing non-English words. Since this process already ensures that we have useful and non-misleading words, we only at a later stage removed rare words (appearing in less than 1% of the entire corpus). This being done, we selected the top 250 of each class and united it to have 1000 features overall. Some of the selected features are the following: ['commander', 'ripping', 'lot', 'aside', 'gap', 'groff', 'express', 'wait', 'encouragement', 'regional', 'crusade', 'julio', 'burnt', 'number', 'carcinoma', 'coca', 'put', 'belly', 'picket', 'sampler']. Quite well related to the news domain.

We use a multilinear regression for the multiclass classification task. After calculating mutual information to determine feature importance we implement the model on top selected features. The class is defined using cross entropy as the loss function and a gradient descent loop to maximize accuracy in predictions. The gradient descent loop is coded to terminate after either a maximum number of iterations (5000 in this test) is reached or convergence is met between the training and validation sets such that the calculated loss exceeds that of the prior iteration. We then checked the gradient with a small perturbation test which was calculated to be 5.328255059620041e-13 while the best validation loss from gradient descent was measured at 0.4887632494975844.

3 Implementation of Logistic and Multiclass Classifiers

For this task, we implement our defined Logistic Regression model on the IMDB dataset with a sigmoid logistic function, cross-entropy as our cost function, and a gradient descent function. The implementation we conducted was limited to a maximum of 10,000 iterations (1e4 as opposed to the coded default 1e5) to offset long processing times and defaults to a learning rate of 0.1. We test our results against a variety of sklearn models with no hyperparameter tuning.

Model	AUROC
Our Logistic Regression	0.93
sklearn Logistic Regression	0.92
sklearn Decision Tree	0.67
sklearn KNN	0.60

We use a multilinear regression for the multiclass classification task. After calculating mutual information to determine feature importance we implement the model on top selected features. The class is defined using cross entropy as the loss function and a gradient descent loop to maximize accuracy in predictions. The gradient descent loop is coded to terminate after either a maximum number of iterations (5000 in this test) is reached or convergence is met between the training and validation sets such that the calculated loss exceeds that of the prior iteration. We

then checked the gradient with a small perturbation test which was calculated to be **5.328255059620041e-13** while the best validation loss from gradient descent was measured at **0.4887632494975844**.

3.1 Small Perturbation Test

We check our logistic regression gradient descent with a small perturbation test. We can see the results of the relative error are in line with what we would expect (a very small number, in this case, 4.752217356242678e-14) where the norm of gradient descent at the final iteration was 4.159e-03.

3.1.1 Small Perturbation: Logistic Regression

Test Type	Result
Analytical Gradient	-520681.094165622
Approximated Gradient	-520681.0823179607
Relative Error	1.2943796138003165e-16
Gradient Descent Norm (Final Iteration)	0.004159

Following that we implement the small perturbation test for the multinomial regression, finding a differential error of **5.328255059620041e-13**

4 Run Experiments

Method comparison using ROC curve

For our logistic regression, we determine the top 10 features contributing to both positive and negative sentiment, similar to the linear regression above. However, we return features with much more perceived correlation to their respective sentiments and less noise than in the simple linear regression implementation.

4.1 Logistic Regression Feature Importance

4.1.1 Top 10 features contributing to positive and negative reviews:

Positive Review

No.	Feature	Coefficient
1	excellent	1.0271664477901652
2	favorite	0.9616220483467965
3	wonderfully	0.9125637572881781
4	7	0.8733444879391511
5	superb	0.8682849728703693
6	perfect	0.845304863947304
7	amazing	0.7902154380366126
8	highly	0.7760788093888727
9	rare	0.7455531769730436
10	loved	0.7392080478277737

Negative Review

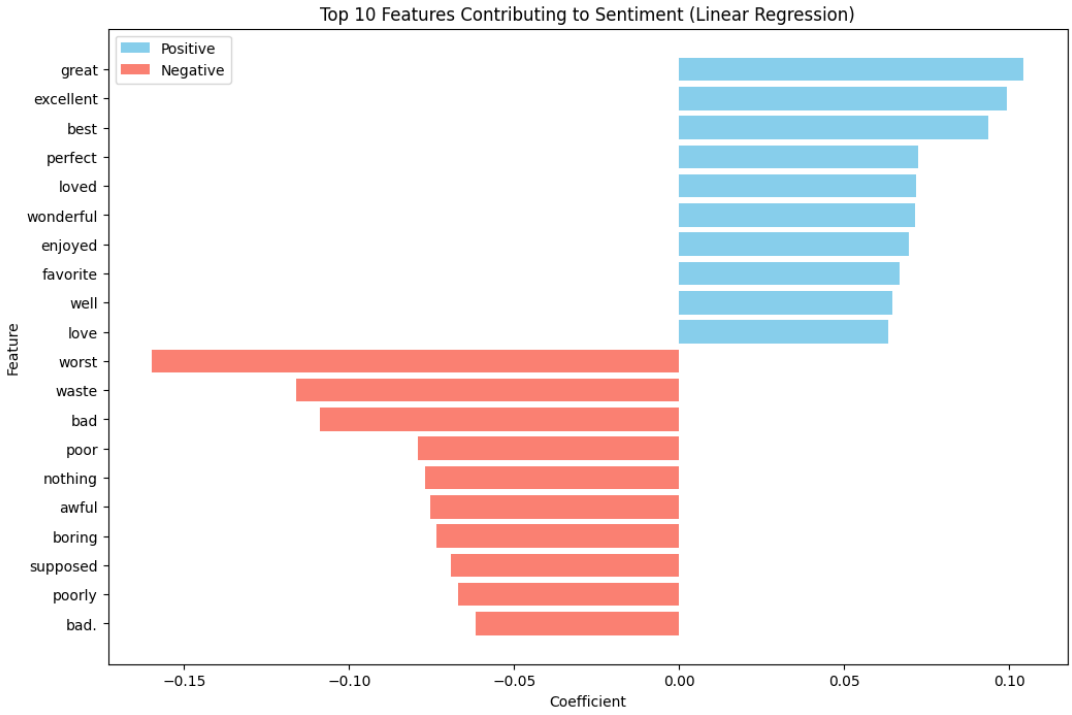
No.	Feature	Coefficient
1	worst	-1.921883624575095
2	waste	-1.740160633924712
3	poorly	-1.3568880474743863
4	awful	-1.1789621577242535
5	dull	-1.109172507766721
6	awful.	-1.0934833178474992
7	fails	-1.0664330824307509
8	boring	-1.0190557483285276
9	lame	-0.9763050921542816
10	badly	-0.9652634891564819

Then we have the implementation of the multilinear regression model after adjusting for mutual information score. We find the accuracy to be substantially higher than the Scikit-learn decision tree model with no hyper-parameter tuning, reflecting the benefit of the multilinear regression model along with the capabilities of gradient descent in minimizing cross-entropy loss.

Model	Accuracy
Multilinear Regression	0.8372933884297521
Decision Tree	0.75

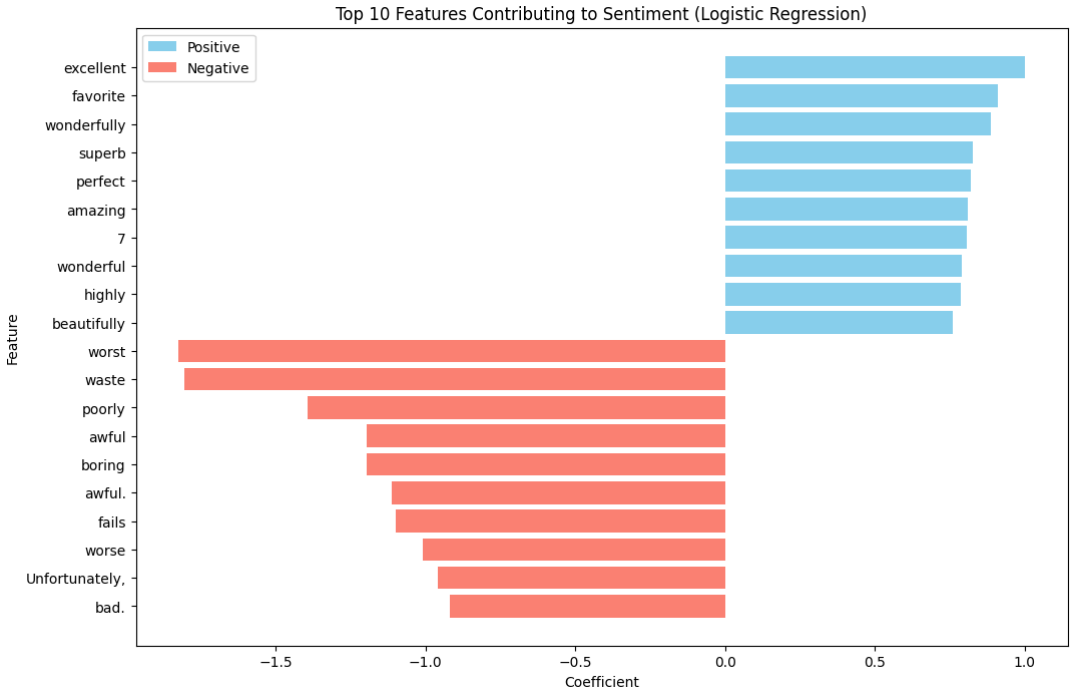
5 Results

Here we can see two bar graphs, one showing the top 10 features from the simple linear regression, and the next showing the top 10 features from logistic regression. We can see a notable difference in the feature selection done by both models, but both models show intuitive top features for both positive and negative sentiments and are clearly functioning effectively.



[!htp]

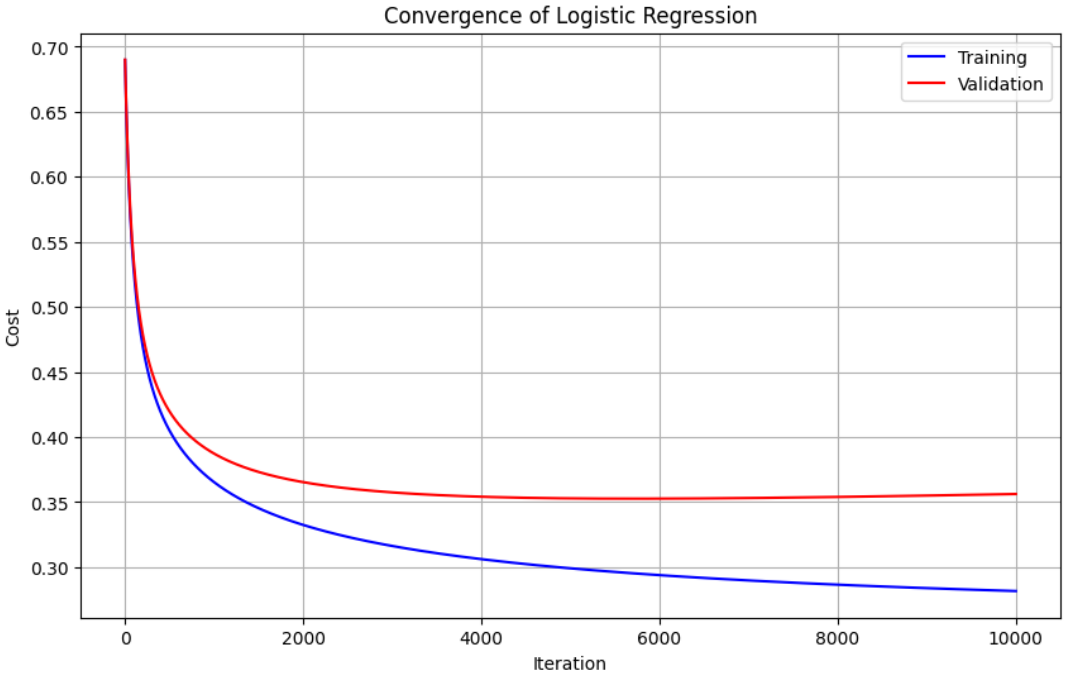
Figure 1: Top Features from Linear Regression



[!htp]

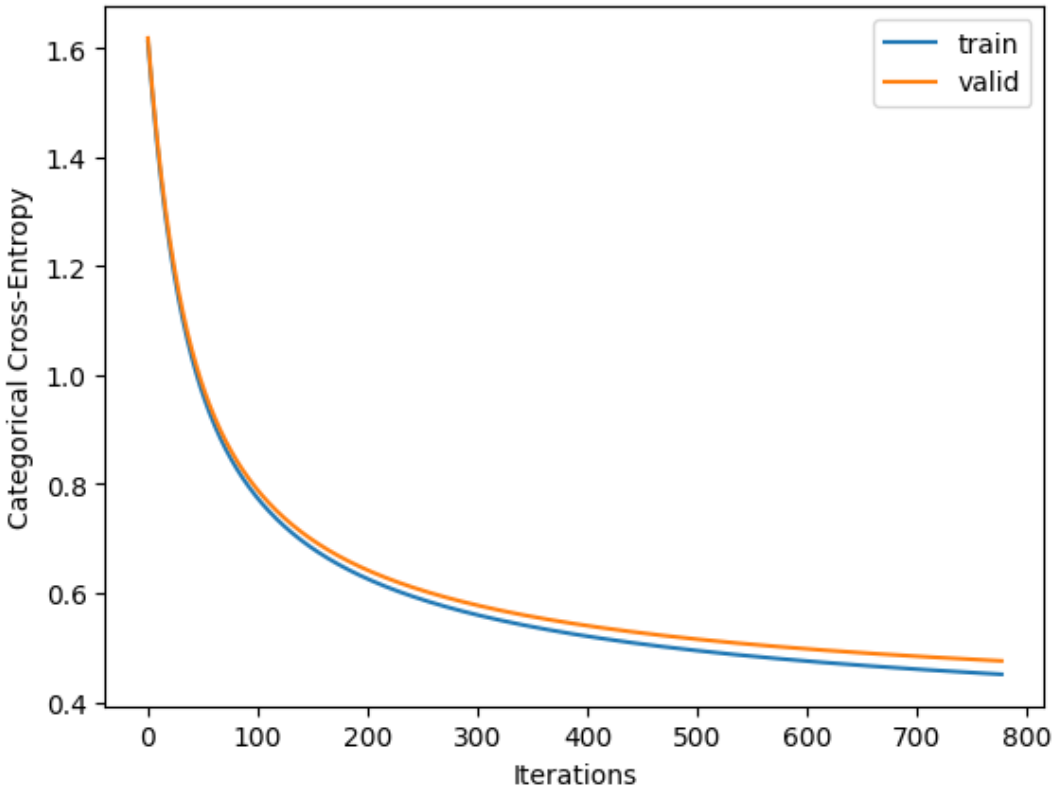
Figure 2: Top Features from Logistic Regression

Tests for convergence were also completed for both models and shown below. In the logistic regression, we observe slight overfitting that begins to happen around the 6000th iteration, observable by an increase in cross entropy loss in the validation set. With less than 1000 iterations, the multilinear regression shows no signs of overfitting.



[htp]

Figure 3: Logistic Regression Convergence



[htp]

Figure 4: Multilinear Regression Convergence

Below we plot the area under the receiver operator characteristics curve (AUROC) for our Logistic Regression model and compare it to three models from the sklearn toolkit: KNN, Decision Tree, and Logistic Regression. Our model is notably the highest achiever, significantly outperforming the sklearn KNN and Decision Tree models, and marginally outperforming the sklearn Logistic Regression model.

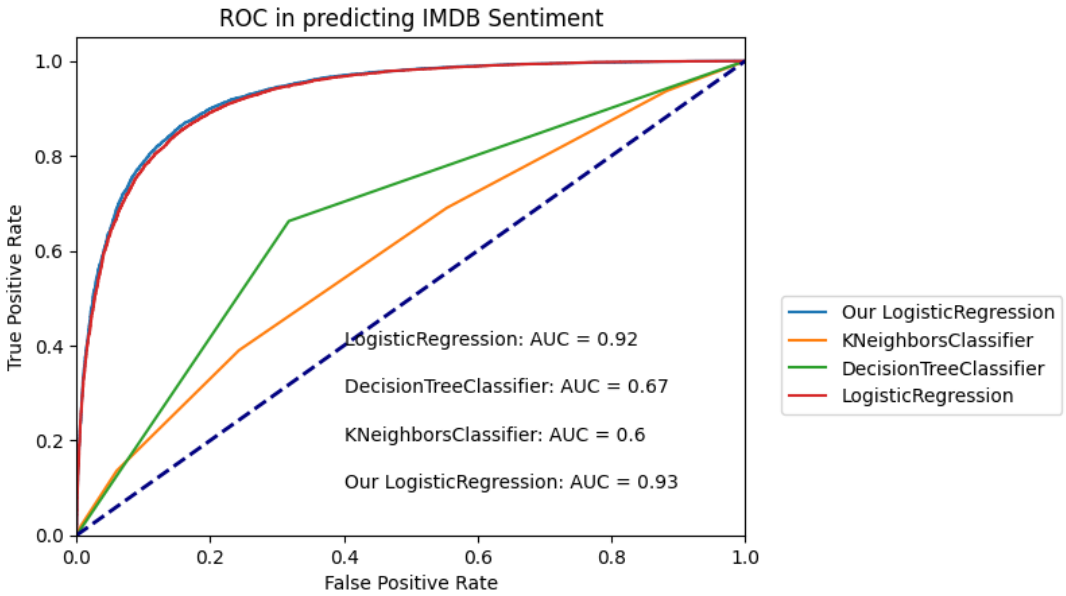


Figure 5: ROC Logistic Regression vs. Scikit-learn Models

Here we compare the area under the receiver operator characteristic curve (AUROC) when implementing our Logistic Regression model with a maximum iteration of 10^4 on various training dataset sizes. We test with random selections of 20%, 40%, 60%, 80%, and 100% of the original training set size to see how this affects results. We

can see a slight but notable increase in accuracy as we increase our training data size, which is congruent with the intuition that more training data allows a model to make more accurate predictions.

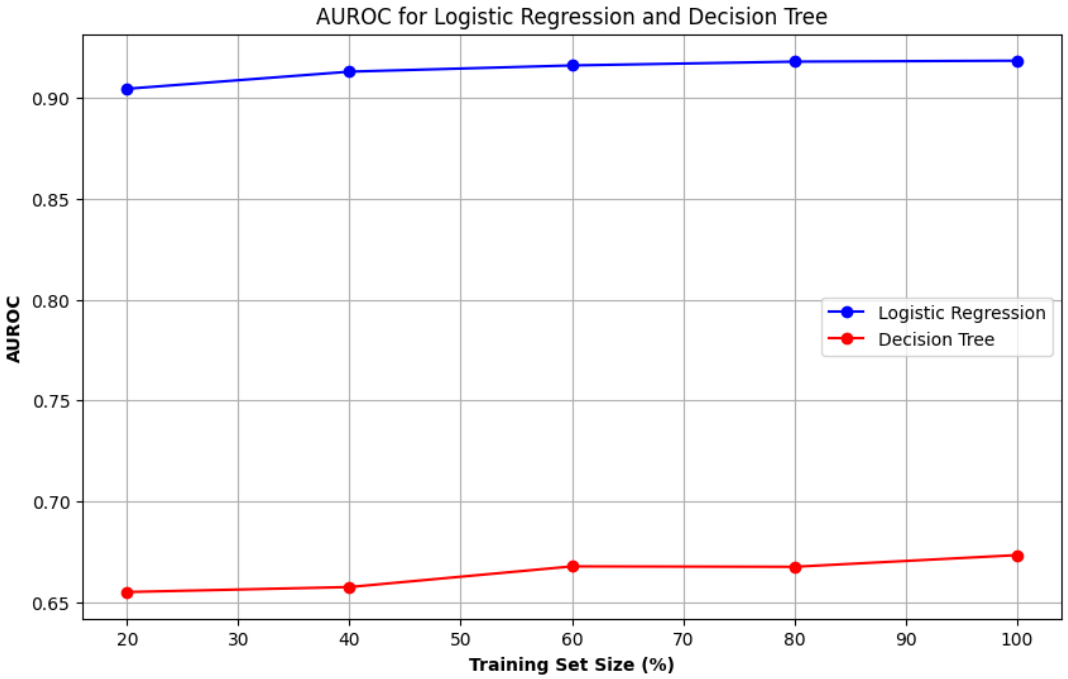


Figure 6: Logistic Regression Accuracy vs. Scikit-learn Decision Tree (Varying Training Data Sizes)

Here we see linear plots for our multilinear regression compared against the sklearn decision tree with no hyperparameter tuning.

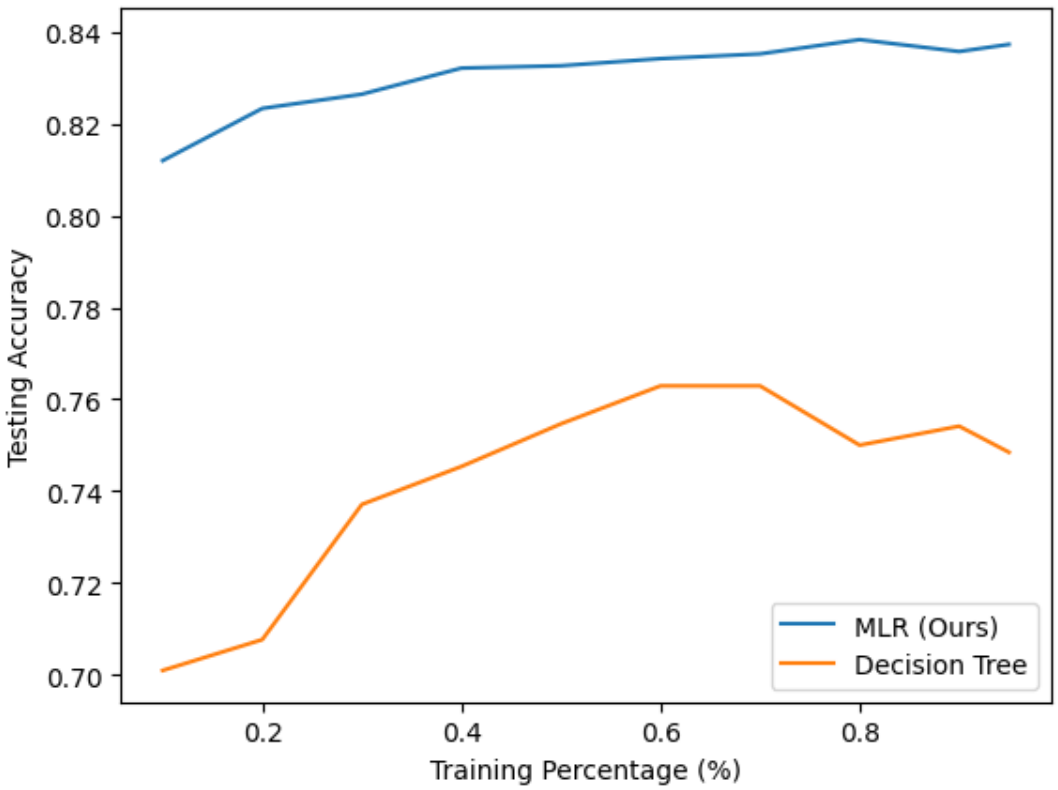


Figure 7: Multilinear Regression Accuracy vs. Scikit-learn Decision Tree (Varying Training Data Sizes)

Finally, we show a heatmap of the top 5 most positive features for the multiclass regression.

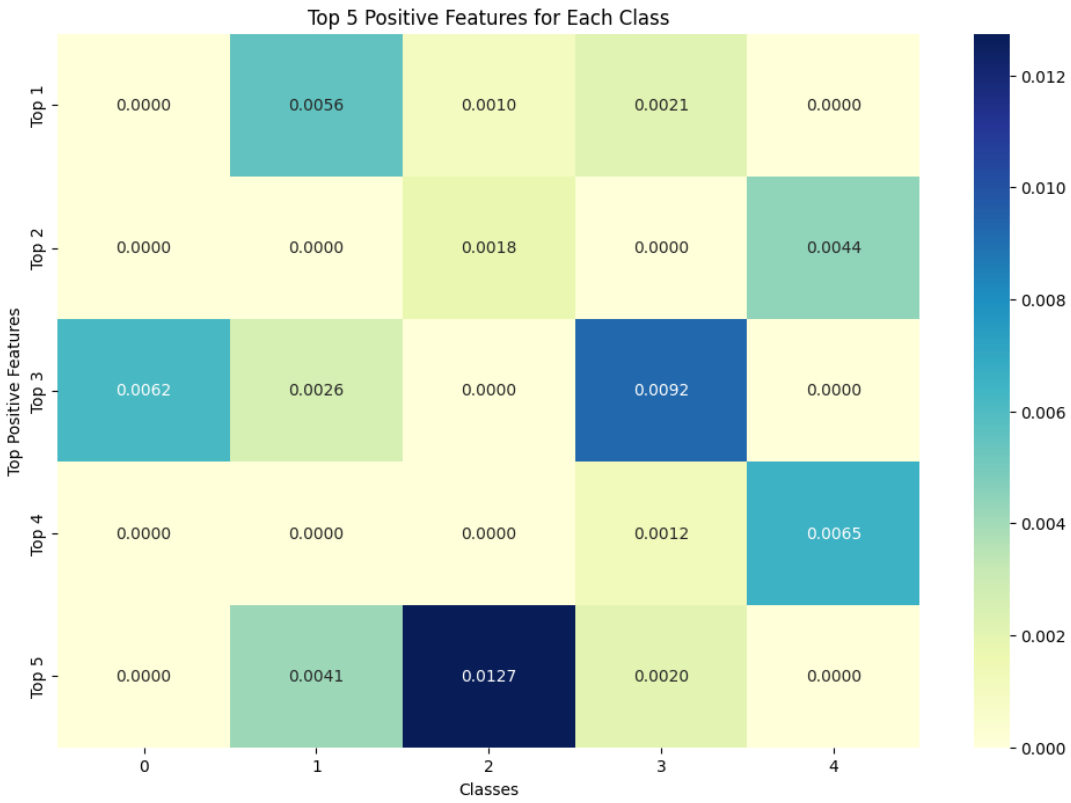


Figure 8: Multilinear Regression Top 5 Features Heatmap

6 Conclusion

In conclusion, we find the logistic regression model implementation to be very robust in predicting sentiment for the IMDB movie review data. The experimentation shows the benefit of strong data cleaning and preprocessing practices, as well as the effects of hyperparameters like feature selection, maximum iterations in gradient descent, and learning rate that contribute to accuracy and/or overfitting. The multilinear regression model was also highly successful in its multi-class classification objective. It highly outperformed the Scikit-learn decision tree model (with no hyperparameter tuning), and benefitted strongly from data preprocessing and feature selection, including the removal of noisy data such as punctuation.

7 Statement of Contributions

Logistic Regression, Multiclass Regression, Report - Mat Vallejo
Multiclass, Report - Bonaventure Dossou
Review and Report - Minjae Kim