Classification of Textual Data

Ben Hepditch, Rudolf Kischer, Maxx Railton

February 26, 2024

Github Link

Abstract

In this project, we investigated the performance of linear classification models on two different benchmark text datasets. The first dataset was used to benchmark binary classification to predict whether words from IMDB reviews were used in a positive or negative movie review. The second was used to benchmark multi-class classification by predicting the genre of news articles. Our experiments revealed that our Logistic Regression implementation achieved superior performance on the binary classification task compared to Decision Trees and K Nearest Neighbors. On the second dataset we experimented with using a multi-class logistic regression (MCLR) model for a classification task on 5 classes of text data. We compared against KNN and Decision tree classification techniques and found that the MCLR outperformed both techniques on every category.

1 Introduction

The goal of this project is to explore the different forms of linear classification models and the effects that selecting features by their linear coefficients can have on model performance. Linear classification is a broad term for machine learning techniques that separate data points into different classes using a linear decision boundary. Models of this nature can generally be divided into two main classes: binary classification and multi-class classification. For binary classification, we evaluated the performance of a custom logistic regression model on a dataset of IMDB reviews and compared its performance to other classifiers using AUROC. For multi-class classification, we chose to evaluate the performance of a custom multi-class regression model on a dataset of 20 news categories (groups) and compared their performance by their accuracy on the test set.

Logistic regression is a statistical method used to model the probability of a binary outcome based on one or more predictor variables, applying a logistic function to predict the likelihood of an event occurring[8]. In binary classification, logistic regression separates data into two classes by fitting a logistic curve to the observed data points, assigning probabilities to each class, and then making predictions based on a threshold probability. Multiclass regression is a machine learning technique used to predict categorical outcomes with more than two possible classes, often employing methods like multinomial logistic regression or softmax regression to model the probabilities of each class. In multi-class classification, multiclass regression extends logistic regression to handle scenarios where the target variable has multiple categories, allowing the model to predict the probability of each class and assign the observation to the most likely category[10].

The models we used for our experiments were a custom logistic regression and a custom multi-class regression model. The former involves methods for computing loss, calculating gradients using cross-entropy loss, and model training through gradient descent optimization. The latter includes methods for softmax calculation, loss, and gradient computation using cross-entropy loss, and model training using gradient descent optimization. To benchmark both of our models we used two different labeled datasets. These datasets have been widely used in the field of machine learning to evaluate the performance of different models, including large language models, adversarial neural networks, and clustering algorithms on their ability to predict the sentiment of a text corpus[9][2][7].

2 Datasets

We used two text-based datasets for evaluating different linear classification models. The first dataset was The Large Movie Review Dataset v1.0[4] consists of 50,000 movie reviews. The bag of words (BoW) feature files (.feat) contain tokenized representations of reviews stored in LIBSVM format, where each line corresponds to a review and features indicate the frequency of specific words. The accompanying vocab file (imdb.vocab) maps feature indices to their respective text tokens, aiding in the interpretation of the BoW features by providing the word corresponding to each feature index. We used this file to extract the frequencies of 89526 features (words) to predict the sentiment (binary label) of the word based on the rating of the movie reviews it is used in. Any rating with a label above 5 was 'positive' and all other ratings were labeled as 'negative'. Previous work with the same dataset has indicated that encoding the reviews as word embeddings instead of a BoW representation is more memory-efficient[3]. However, given that the models we selected for this dataset

have relatively simple architectures compared to multi-layer neural networks, we used the provided representation for simplicity of implementation.

We further reduced the dataset based on the distribution of word frequencies shown in Figure 1, to remove rare words and stopwords. That is words that appear in less than 1% of the documents and words that appear in more than 50% of the documents. This filtering reduced the dataset to around 1700 features. To make the feature size more practical for training our models, we applied simple linear regression to the reduced feature set and sorted the features by the absolute values of their regression coefficients, of which the top 1000 features were used to create the training set.

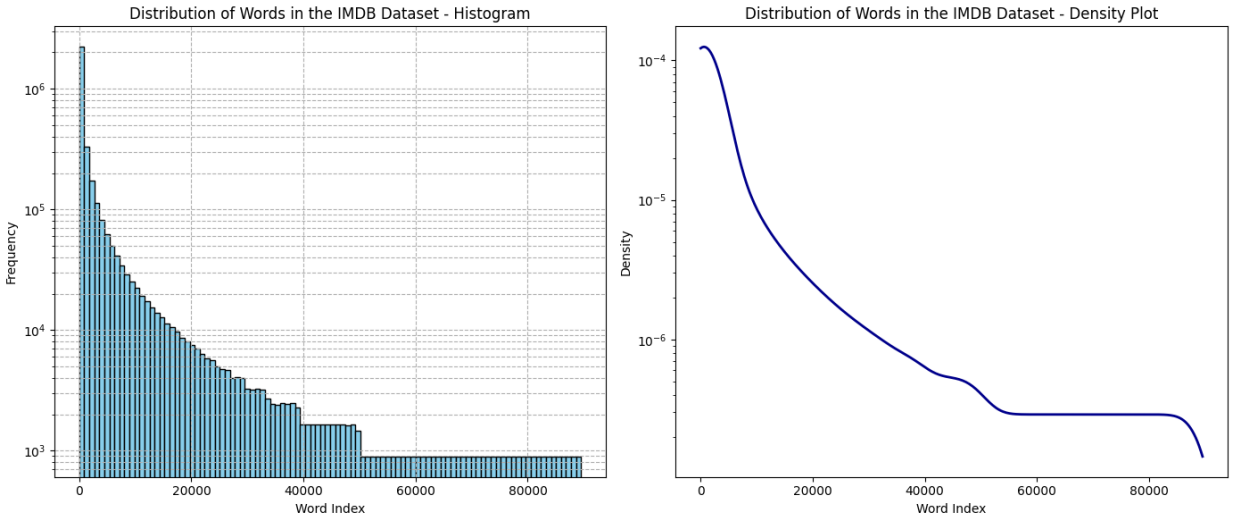

Figure 1: Distribution of Word Frequencies in the IMDB Dataset

The second dataset, The 20 Newsgroups[6], contains roughly 20,000 unique news articles split evenly across 20 different newsgroups. We used Scikit-learn's Real-world datasets API to access the aforementioned dataset[5]. The dataset is made up of a list of raw text that were fed into feature extractors to extract the feature vectors. CountVectorizer[5] was used to preprocess, tokenize and filter stopwords, building a dictionary of features and transforming documents to feature vectors. We used the parameters 'stop_words', 'max_df', 'min_df', and 'ngram_range' to choose what words to ignore and whether to consider unigrams or bigrams for features. After using Countvectorizer we used the TfidtTransformer[5] to take the matrix of token counts and produce a matrix of TF-IDF features. The rows of this matrix are normalized so that they have unit Euclidean norms. This is done to downscale the impact of tokens that appear very frequently, making them less informative than features that occur in a smaller fraction of the data. The final part of the selection process was the use of SelectKBest with mutual_info_classif[5], which calculated the top k features that had the

highest mutual information with the class labels. After all these steps were completed, we were left with the features that most significantly reflected each class.

3 Results

3.1 Experiment 1: Top 10 Positive and Negative Features in the IMDB Dataset

As mentioned previously, we filtered the training data by the absolute regression coefficient of each feature to reduce the dimensionality. This technique has been extensively shown to be an effective pre-processing technique when using linear models[1]. Figure 2 shows the top 10 negative and positive coefficients which can be conventionally thought of as a reflection of each word's sentiment. Interestingly, the top coefficients from logistic regression are similar to those from linear regression, with a few notable differences that stem from their distinct methodologies and underlying assumptions. Linear regression assumes a linear relationship between features and the target variable, aiming to minimize squared differences, while logistic regression models the probability of binary outcomes using the logistic function. In sentiment analysis, logistic regression may assign higher coefficients to words influencing the likelihood of sentiment due to its probabilistic nature, leading to differences in coefficient magnitudes compared to linear regression.

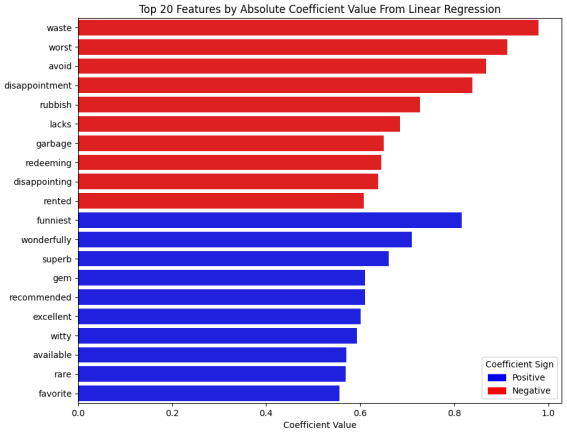

Figure 2: Top 20, Linear Regression

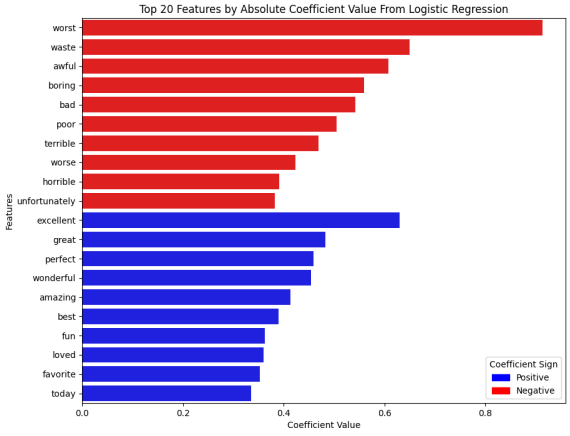

Figure 3: Top 20, Logistic Regression

3.2 Experiment 2: Classification Results

To evaluate the performance of our models we benchmarked them against their respective sckit-learn models as well as scikit-learn's Decision Tree and K Nearest Neighbors. Logistic regression for binary classification was performed on the first dataset and multi-class regression for multi-class classification was performed on the second dataset.

Hyperparameter	**Value**
Learning Rate	0.01
Epsilon	1e-5
Max Iterations	1e5
Add Bias	True

Table 1: Hyperparameters of the Logistic Regression model

Within the scope of the IMDB dataset binary classification task, the Custom Logistic Regression model achieved the highest AUROC value of 0.93. The hyperparameters for this model are shown in Table 1. This was followed by the standard Logistic Regression model with an AUROC of 0.86. The Decision Tree and K Neighbors classifiers presented lower AUROC values of 0.71 and 0.69, respectively. As seen in Figure 6, these findings indicate a quantitative advantage in classification performance for logistic regression models over the Decision Tree and K Neighbors models in this dataset.

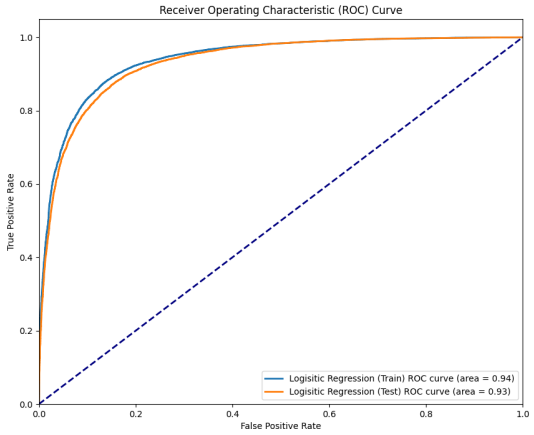

Figure 4: ROC curve results of binary classification using our custom logistic regression model

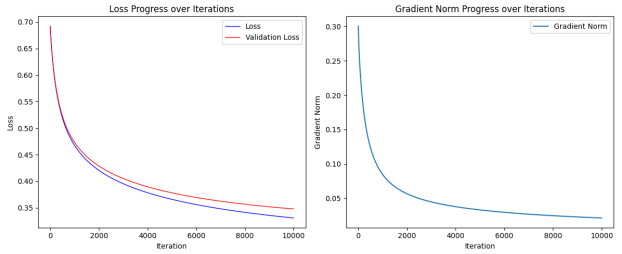

Figure 5: Loss and gradient progress of our custom logistic regression model

3.2.1 Comparison of Binary Classification Models on the IMDB Dataset

The ROC curve analysis for binary classification models on the IMDB dataset indicates that the custom logistic regression model outperforms the other models, with an AUC of 0.93, demonstrating excellent classification ability. The standard logistic regression model follows with an AUC of 0.86, indicating good performance. The decision tree classifier and K Nearest Neighbors (KNN) model show moderate performance, with AUCs of 0.71 and 0.69, respectively, suggesting they are less effective at distinguishing between positive and negative classes in this dataset.

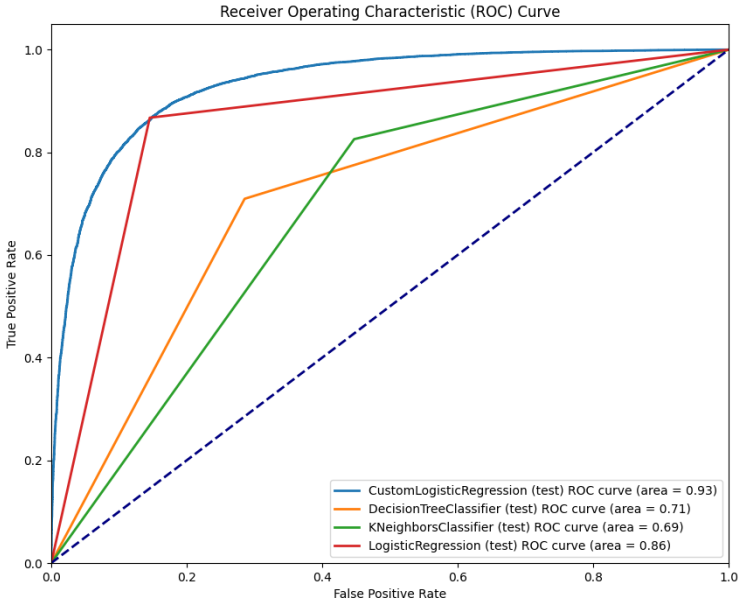

Figure 6: Performance of different linear models on the binary classification task

3.2.2 Comparison of Multi-class Classification Algorithms on the News Dataset

Regarding the 20 news categories dataset, the loss versus iterations plot for the MultiClassRegression model shows a healthy learning curve, where the training and validation losses converge without significant divergence, showing that the model generalizes well without any overfitting. In terms of accuracy, the MultiClassRegression model was the most effective, achieving the highest average accuracy, particularly in 'comp.graphics' and 'misc.forsale' classes, with an accuracy of 0.87. This can be seen in both Figure 9 and Table 2. The LogisticRegression also performed consistently, outperforming the MultiClassRegression model in the 'rec.sports.baseball' class with an accuracy of 0.82. The KNeighborsClassifier and DecisionTreeClassifier were significantly less accurate, exhibiting significantly lower accuracy in certain classes like 'sci.med' and 'talk.politics.guns'.

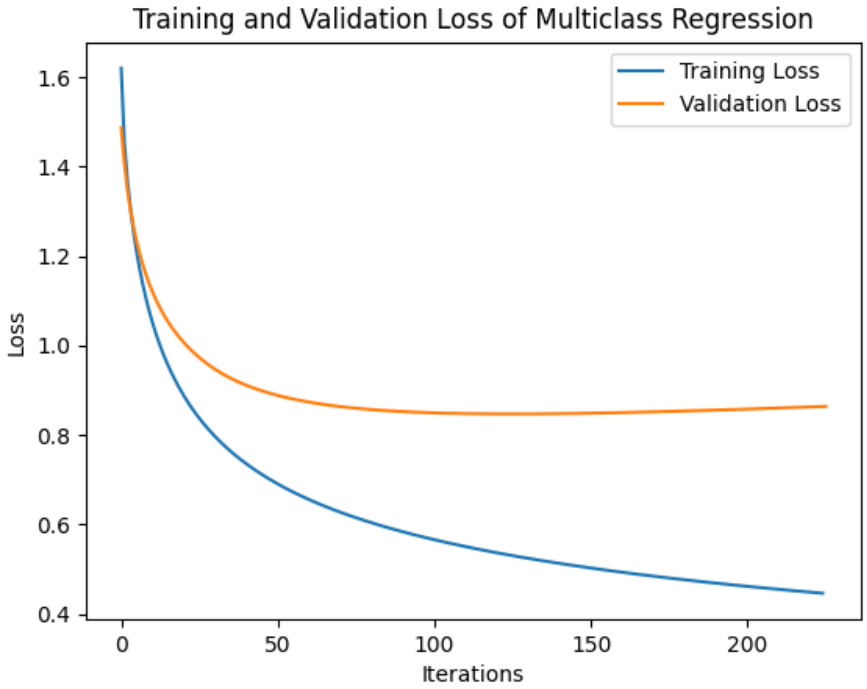

Figure 7: Multi-class loss versus iterations

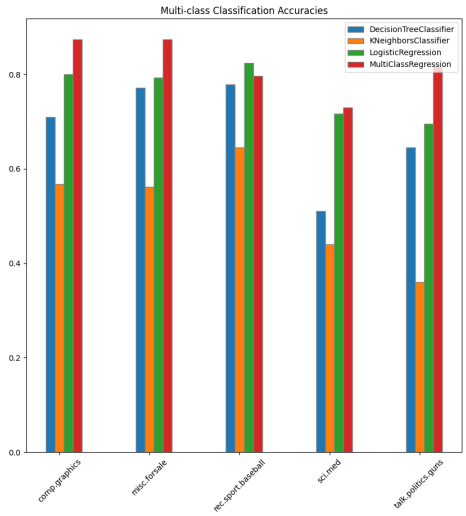

Figure 8: Comparison of model class accuracies

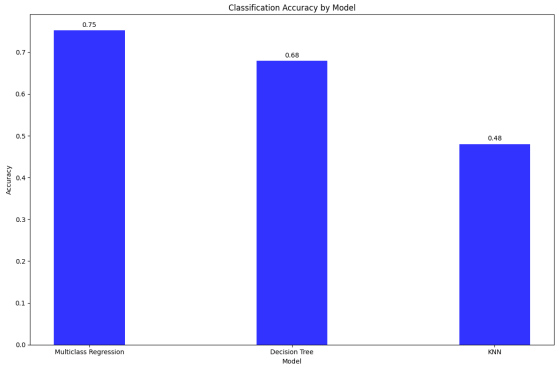

Figure 9: Comparison of classification accuracies of models on the news dataset

3.3 Experiment 3: Effectiveness of DT and Multiclass and Logistic Regression With Limited Training Data

In evaluating the performance of Logistic and Multiclass Regression and Decision Tree classifiers across varying sizes of training data, our study revealed that Logistic Regression consistently outperformed the Decision Tree model. The Logistic Regression maintained an AUROC of approximately 0.93 across all training set sizes, demonstrating robustness to the amount of training data. In contrast, the Decision Tree's performance was markedly lower, with an AUROC in the range of 0.70 to 0.71. Notably, both models showed little improvement as the training set size increased. In terms of accuracy, the Multiclass Regression model also constantly outperformed the Decision Tree, with an average accuracy of 0.744. The Decision Tree performed notably lower, with an average accuracy of 0.636. The Multiclass Regression's accuracy was also more consistent having a maximum fluctuation of only 0.02. On the other hand, the maximum fluctuation for the Decision tree was 0.1. These findings suggest that Logistic and Multiclass Regressions are more suited to the dataset in question, potentially due to their ability to handle linear relationships and interactions more effectively than the Decision Tree model on the selected dataset.

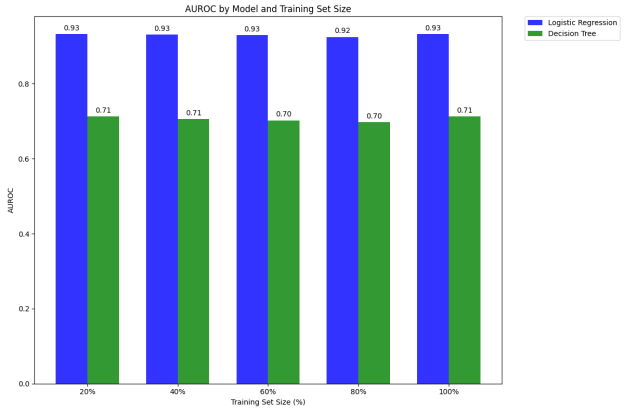

Figure 10: AUROC on IMDB data

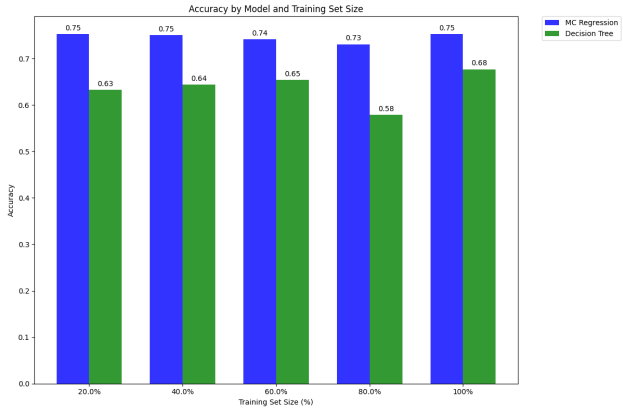

Figure 11: Accuracies on news data

3.4 Experiment 4: Effectiveness of Logistic Regression and MultiClass Regression with Different Learning Rates

In this experiment, we evaluated the effectiveness of the logistic regression model across four different learning rates (LRs): 0.01, 0.001, 0.0001, 1e5. The gradient norm and training loss results indicated that higher LRs led to faster convergence but risked overshooting the optimal solution, as evidenced by the plateauing of the gradient norm for LR=0.01. Conversely, the lowest LR (1e5) showed very little decline in both gradient norm and loss, suggesting that it might be too conservative, resulting in slow convergence and potential underfitting. Interestingly, despite the aggressive learning approach, the LR=0.01 model achieved the highest Area Under the Curve (AUC) value of 0.93 on the Receiver Operating Characteristic (ROC) curve when evaluated against the test dataset, implying superior generalization ability. Meanwhile, the smallest LR yielded the lowest AUC of 0.85, affirming a decrease in discriminatory power. These results indicate that the optimal learning rate for logistic regression on this dataset appears to be 0.01.

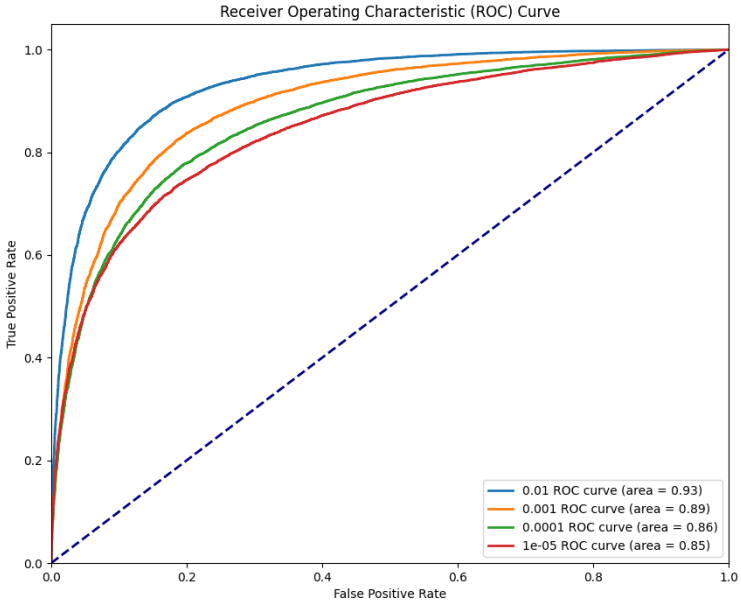

Figure 12: ROC curve for different learning rates

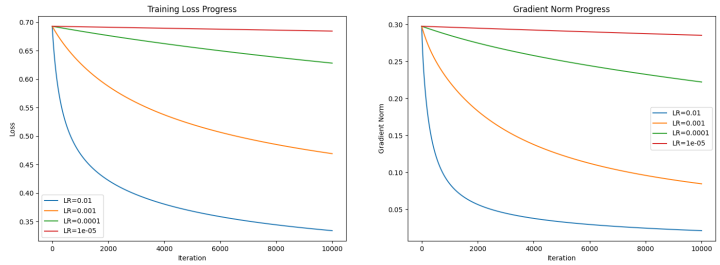

Figure 13: Loss and gradient progress

3.5 Experiment 5: Top 5 Most Positive Features for 4 Classes in Multi-Class Classification from the 20-News Group Dataset.

The heatmap analysis shows the top 5 features for 4 selected classes from the 20-News Group Dataset. They are ranked by their respective feature importance scores. In the 'comp.graphics' class, the feature 'graphics' had the highest score of 0.53, indicating its strong association with this class. For 'misc.forsale,' the feature 'sale' has the highest importance at 0.56. In the class 'rec.sport.baseball', both 'team' and 'year' share the highest significance, at 0.40. Lastly, the 'sci.med' class displays an equal score for the terms 'geb', 'intellect', and 'skepticism' at 0.30. All the top features clearly relate to their respective classes, except for 'sci.med'. These less obvious relationships between feature and class may result from CountVectorizer's parameter values during data preprocessing. This function may have

over-filtered vital terms that could have been top positive features by ignoring terms that appear more than our chosen frequency threshold. This frequency threshold can be easily adjusted by changing the values of min_df and max_df, allowing for the potential solution to this error.

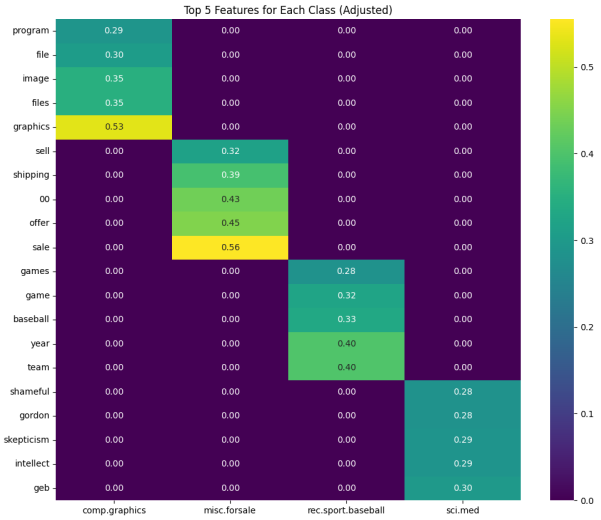

Figure 14: Heatmap of the Top 5 Most Positive Features in MultiClass Classification on the 20 News Group Datasets.

3.6 Experiment 6: Multi-Class Prediction Accuracy for increasing number of classes

In this experiment we re-evaluate the 20 news group dataset using the same MCLR model, but now instead of only 5 groups we increase the number of groups up the maximum of 20. We do so by starting with 3 random classes, and train the model, just the the data from those groups and evaluate its performance. We then repeat the process but add in another randomly selected group. Our experiment found that by increasing k there is a negative correlation with performance by some linear factor. Note that because we select the new group at random, this can lead to some variation depending on the ordering. To get more conclusive results you could average the performance over multiple different orderings but this becomes expensive quickly. Any jumps or dips in performance may be due to the individual difficulty of classification for the content of the group that is added.

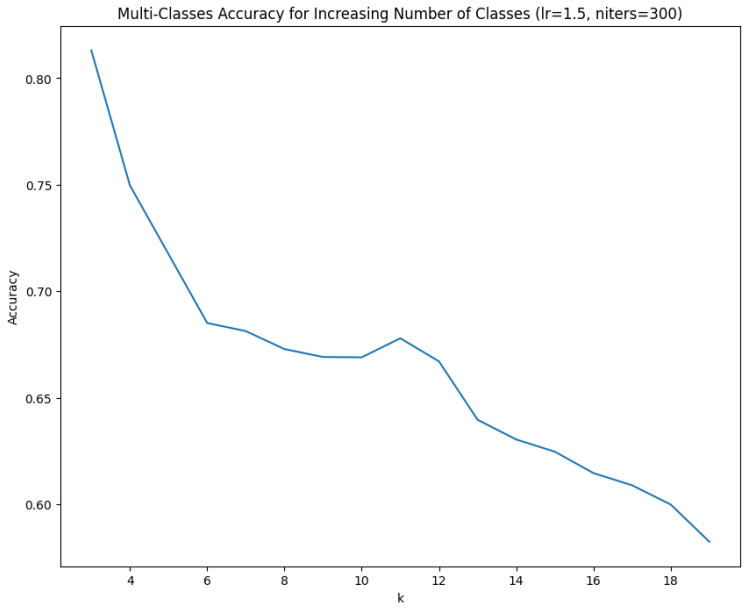

Figure 15: Accuracy for multi-class prediction for group sizes

4 Discussion

The results of our experiments shed light on several aspects of the classification models' performance on text data. Logistic Regression emerged as a superior model for binary classification on the IMDB dataset, as evidenced by the highest AUROC score. This suggests that logistic regression is particularly adept at capturing the linear boundaries between classes in high-dimensional text data, which aligns with the model's theoretical strengths.

The top features identified through regression analysis for the IMDB dataset align with intuitive expectations, with positive coefficients corresponding to words typically found in positive reviews, and negative coefficients with words in negative reviews. This validates the model's ability to discern sentiment-laden words effectively and confirms the usefulness of the regression coefficient-based feature selection approach. The learning rates tested in Experiment 4 provide critical insights into model training dynamics. Specifically, the significant out-performance of a model trained with a learning rate of 0.01 over lower learning rates suggests that the regression can find relatively clean boundaries between classes at the chosen learning rate which it fails to fit when that rate is too low. Moreover, this is indicative of clear separations between classes that follow a convex shape, with few local minima that would otherwise lead to overfitting if the test data did not follow a similar pattern. However, it is crucial to note that this does not necessarily indicate the model will be able to generalize enough to predict the sentiment of other corps. Lastly, despite the demonstrated robustness of the logistic regression model on varying sizes of training data, the results also highlight a plateau in performance gains. This plateau could suggest that beyond a certain data size, the quality of data and feature representation may play a more significant role than quantity alone.

5 Conclusion

In conclusion, our experiments highlight the effectiveness of logistic regression and multi class logistic regression in text classification tasks over other linear models like Decision Trees and K Nearest Neighbors. Logistic regression's capacity to model the probability of class membership makes it particularly suitable for binary classification problems such as sentiment analysis in text, as it can handle sparse data well and is robust to variations in training data size.

Future investigations could explore the integration of non-linear models and deep learning approaches, which may capture complex patterns in data that linear models cannot. The

model convergence times could also greatly benefit from adaptive learning rate methods such as momentum or stochastic gradient descent.

Additionally, experimenting with alternative text representations, such as word embeddings or transformer-based models, might provide further performance improvements. Finally, expanding the datasets and varying preprocessing steps could offer more generalizable insights into the models' performance across different text classification tasks.

6 Statement of Contributions

Ben Hepditch: Implemented the code for the logistic and linear regression models, and conducted the data cleaning, experiments, and write-up for the IMDB dataset. Also wrote part of the discussion/conclusion and abstract. Maxx Railton: Worked on multiclass implementation, experiments and write-up for results. Rudi Kischer: worked on multiclass experiments and plots for the write up.

References

[1] Andrea Bommert, Xudong Sun, Bernd Bischl, Jörg Rahnenführer, and Michel Lang. Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, 143:106839, 2020.

[2] Zhijie Gan and Zicheng Zhu. An empirical study on how emotion affects the probability of replies based bert. *Applied and Computational Engineering*, 41:148–152, 02 2024.

[3] Md. Rakibul Haque, Salma Akter Lima, and Sadia Zaman Mishu. Performance analysis of different neural networks for sentiment analysis on imdb movie reviews. In *2019 3rd International Conference on Electrical, Computer Telecommunication Engineering (ICECTE)*, pages 161–164, 2019.

[4] 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.

[5] 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.

[6] S. J. Phillips, R. P. Anderson, and R. E. Schapire. Maximum entropy modeling of species geographic distributions. *Ecological Modelling*, 190:231–259, 2006.

[7] Anshula Raj and Seba Susan. *Clustering Analysis for Newsgroup Classification*, pages 271–279. 07 2022.

[8] J.C. Stoltzfus. Logistic regression: A brief primer. *Academic Emergency Medicine*, 18:1099–1104, 2011.

[9] Abinash Tripathy, Utpal De, Bibhuti Dash, Sudhansu Patra, Ch Rao, and Trilok Pandey. Sentiment classification of reviews using combination of firefly algorithm and ann. 02 2024.

[10] Jingyu Wang, Hongmei Wang, Feiping Nie, and Xuelong Li. Feature selection with multi-class logistic regression. *Neurocomputing*, 543:126268, 2023.