Assignment 2: Classification of Textual Data
COMP 551

Sophia O., Jayden M., Negin E.
McGill University

February 26th, 2024

1 Abstract

In this assignment we were asked to implement both a Logistic Regression classifier and a Multi-Class Classifier. The Logistic Classifier was used on an IMDB data set to perform binary sentiment analysis. The multi-class classifier was used to classify news group data into one of five news topics. We then compared the performance of our models as compared to a decision tree classifier. For the Logistic Regression classifier we found that it outperforms the decision tree on both test and validation sets. The multi-class classifier had a higher accuracy on test sets compared to the decision tree but, was slightly worse when compared to LASSO and Ridge Regression models.

2 Introduction

For this assignment we implemented two machine learning models from scratch, Logistic Regression and Multi-class Regression. We then had to perform classification tasks on text data from IMDB movie reviews data set and the 20-news group data set respectively. Both data sets provided textual input data which required conversion into input vectors and a significant amount of feature reduction to filter out words unrelated to the classification task and improve computational load. After implementing both models our primary tasks were to assess the performance of these models to determine optimal hyperparameters such as learning rate of gradient descent, and then do a model comparison between the regression classifiers we implemented and a decision tree classifier. We found that both models out performed the decision tree in terms of prediction accuracy. Finally we compared the accuracy of the two models as a function of the size of the data set. We found that for the IMDB data set, as little as 60% of the data could be used without affecting the AUROC on the test or validation set. The IMDB data set is commonly used introductory set used for model comparison investigations. Different machine learning models are compared based on how well they can perform text based sentiment analysis as we did here. A study by B.S Harish, Keerthi Kumar and HK Darshan investigates how a hybridized type of feature concatenating machine learning features with lexicon features can increase the performance machine learning models performing sentiment analysis [**2**]. The 20 News Group data is another popular set for performing experiments on a variety of possible models such as classification and clustering tests [**1**].

3 Methods

3.1 Logistic Regression

Logistic Regression is a statistical method used for binary classification tasks, predicting the probability of an instance belonging to a particular class. It models the relationship between the independent variables and the binary outcome using the logistic function. The logistic function, also known as the Sigmoid function, maps any real-valued number to the range [0, 1]. In this assignment we implemented this logistic algorithm for binary classification as a class object effectively considering the logistic mapping expressed as:

$$P(Y = 1) = \frac{1}{1 + e^{-XW}}$$

We used the hand calculated gradient function for the sake of speed to learn the weights gradually through iterations using the gradient decent method to fit the model to the training data. This allowed us to find the optimum weight to predict the test data in the testing phase. We used two stopping criteria, maximum iterations and small gradient L2 norm. The motivation for the first criteria is prevent huge computational cost for no reason, particularly in a case where the weights fail to converge. The second criteria is an indicator that the value of the weights has converged. When L2 norm is small it shows that numerically very little changes in coefficients are applied at the given iteration. We checked the gradient calculation using numerically calculated gradient with small perturbation in the loss function and the results can be seen in the Figure 1 which ensures us about the gradient well-computation. Also, the figure shows the decrease in loss function, cross entropy, as we go further through iterations which is a sign that we are in line with our goal, which is iteratively changing the features weights in a way that minimizes the loss function.

3.2 Multi-class Regression

Multi-class Regression is a statistical method that is similar to Logistic Regression except that it is able to perform classification tasks when the number of classes is greater than two. Multi-class Regression uses the softmax function to map real values to probability values between [0,1], it is similar to the Sigmoid function but over a greater number of classes. The weights for Multi-class Regression are learned using gradient descent with the gradient being calculated as the partial derivative of the cross entropy loss function with respect to the weight matrix. We set a maximum number of iterations to minimize computational cost when the weight is not converging.

3.3 Datasets

3.3.1 IMDB Reviews

The Large Movie Review Dataset is a comprehensive collection designed for binary sentiment classification, intended to serve as a benchmark for sentiment analysis tasks. With a total of 50,000 movie reviews, evenly split into 25,000 for training and 25,000 for testing, the dataset offers a balanced distribution of positive (score >= 7 out of 10) and negative (score of <= 4 out of 10) labels. The dataset includes bag of words (BoW) features in LIBSVM format, and expected ratings for tokens. For the purpose of this assignment we only used the BoW as features and their number of appearance times in labeled reviews. However, since the Bag of Words (BoW) contains 89,526 words, many of which are irrelevant to sentiment (such as rare and stop-words), we tried to filter out these noise features to enhance our model in terms of both speed and accuracy. In order to to this, we filtered out the words that appear in less than 1% of the documents and words that appear in more than 50% of the documents which resulted in 1744 features. Then, again for the purpose of removing noisy features and reducing the computational cost, we ran a Linear Regression model and assess the weights for each feature when predicting the ratings related to each review and selected the top 500 features. In Figure 2 you can see how the Linear Regression weight magnitude is related to the sentiment we get from 10 top words.

3.3.2 20 News Groups

The 20 News Groups is a scikit-learn data set with 20 possible output classes representing news topics. There are 18,864 samples of text data in this data set. For this assignment we have filtered the set to only work with 5 of the 20 categories, reducing the number training points to 2,918 as we arbitrarily decided to work with "rec.sports.hockey", "sci.space", "sci.med", "misc.forsale" and "talk.politics.guns". To reduce the vocabulary to only relevant works for determining the class, as in Logistic Regression we used a CountVectorizer to filter out words that appeared in less than 1% or greater than 50% of documents, leaving us with a vocabulary of 1,520 words. To calculate the top features for each class we calculated the Mutual Information score for each feature relative to each of our target labels. For each label we took the top 80 most important features. When compiling the list of top feature words, each feature was only counted once. The final count of unique top features is 324. The class data is relatively evenly distributed with all classes corresponding to approximately 20% of the data points

4 Results

4.1 Logistic Regression

4.1.1 Checking the model performance and choose the best coefficients

Training data was split into 70% for training and 30% for evaluation. We fit the model on the training data, then check the gradient and monitored the training loss as well as evaluation loss during 1000 iterations using different learning rates. Figure 1, shows the results of the Cross Entropy monitoring. As it can be seen, convergence occurred for all chosen learning rates within the chosen iteration range except for the learning rate equal to 0.01. Also, the evaluation loss increasing after a certain iteration is a sign of over-fitting of the model. Therefore, we picked the coefficients that led to the minimum amount of evaluation loss and since we had faster convergence with learning rate of 0.5, we picked this as our alpha for our further investigations.

4.1.2 Effect of learning rate on convergence speed

Figure 3 clarifies how different learning rates and stopping criteria affected the convergence speed of fitting this data set to our logistic model. Typically, as the learning rate increases the model takes bigger steps towards convergence and the maximum iterations will no longer be the limitation for the gradient descent. However, when the learning rates increases too much, the model may oscillate around the minimum loss coefficients and this way taking more time to reach the optimal point.

4.1.3 Highly weighted features in Logistic Regression

The 10 features which had the highest positive weights and the lowest negative weights after model fitting and coefficients selection were selected. Figure 4 represents these features as words and their corresponding weights in descending order.

4.1.4 Comparison of the evaluation metrics between logistic regression and DT

A decision tree was also fit to training data and we used the evaluation data to select an appropriate tree depth in order to prevent over-fitting. Figure 5 show the performance of the DT on predicting the training and validation data. Then the performance of the two models on training, validation and testing data was compared in terms of accuracy (with the threshold of 0.5), ROC curve, and the AUROC in Figure 6. It can be seen that none of the models are over-fitted but from Figure 7 we can see that Logistic Regression clearly outperforms DT on unseen data.

4.1.5 Effect of training size on AUROC

In this section we tried using 10, 20, 40, 60, 80, and 100% of the training data to see how the AUROC changes if we have different training dataset sizes. Figure 8 is the result of this experiment. As it can be seen, using larger sets of data for training improves the AUROC on test data but since we already have an abundant amount of data, the effect is not very visible in larger sizes. We find an opposite effect on the training data since the training variance is increasing and we face larger training errors with more data points. Finally, on the validation set we can see the results similar to the test set, except for the decrease after the evaluation set size itself gets too small to correctly estimate the AUROC.

4.2 Multi-class Regression

4.2.1 Checking the model performance and choose the best coefficients

The 20newsgroups dataset was split into a 33% test set and a 67% training set which was further split 50-50 into a validation set. So each set consisted of roughly 1/3 of the data points. The model was fit on training data, then the gradient was checked, and monitored the cross-entropy over 500 iterations at various learning rates.

4.2.2 Performing gradient check and monitoring convergence for various learning rates

We performed a gradient check to ensure that to ensure that the model implementation was working as expected. We did this by comparing the gradient computed by our model to a numerical computation and found only minor differences. Figure 9 demonstrates how different learning rates affect convergence when fitting the data. We tested 0.0001, 0.001, 0.005, 0.01 and 0.05. 0.0001 did not converge within the number of iterations we tested, both 0.001 and 0.005 converged nicely. For 0.01 and 0.05 the figures show oscillation of the CE value for the first couple hundred iterations. This is to be expected as larger learning rates are prone to skipping past the ideal value, preventing convergence.

4.2.3 Accuracy comparison with DT, LASSO, and Ridge

The sk-learn Decision Tree, LASSO, and Ridge models were implemented for further comparison with the Multi-class Regression results. All four were trained then the validation set was used to determine optimal hyperparameters for each case. The table below shows the best accuracy for each model when using its ideal hyperparameters. The Decision Tree was tested at a depth of 10, LASSO with an alpha of 0.001, and Ridge with an alpha of 10. Ridge performed the best with a slight margin over LASSO and followed closely behind by multi-class. Decision Tree performed slightly worse than the other three.

Table 1: Accuracy scores for various models

Model	Accuracy	Hyperparameter Value
Multi-class	0.75493	-
Decision Tree	0.60955	Depth = 10
LASSO	0.76012	Alpha = 0.001
Ridge	0.76116	Alpha = 10

Since the dataset was tending to overfit when used with other models, Ridge was used as a comparison since it uses a penalty term to reduce these effects. Whereas LASSO can be a beneficial comparison tool as it can further aid with feature selection.

4.2.4 Effect of training size on accuracy

Similar to Section 4.1.5, the training data was broken down into smaller subgroups to determine if the accuracy scores change when using various training set sizes. Figure 10 shows the results. When accuracy scores are predicted with the test data, we can see that a larger training set leads to higher accuracy scores for both multi-class regression and the Decision Tree model.

4.2.5 Top features for each class

For further analysis of the top features, a heatmap (Figure 11) was created. This highlights the effectiveness of the feature selection process. The top features for each category have higher mutual information scores for their respective class compared to their mutual information scores when they appear in documents from other classes. Thus showing

that there is a strong mapping between the majority of the feature words and their respective categories. Words related to buying and selling appear in misc.forsale, hockey related words in rec.sport.hockey, space related words in sci.space, and words related to guns and the law in talk.politics. guns. The exception would be sci.med, as the feature words are seemingly less connected yet, the lower mutual importance scores for these words accurately reflect that observation.

5 Discussion and Conclusion

The main take away from these experiments is that both logistic and multi-class regression classifiers perform better than a decision tree model for classification tasks. When using AUROC for logistic classification and prediction accuracy for multi-class regression, the decision tree model was unable to perform as well as the regression classifiers. It is notable that this relationship held up even when the size of the training set was reduced to just 10% of its original size. Decision trees make much more course grained boundary lines compared to logistic regression. As such, logistic regression has similar performance on seen and unseen data, while a decision tree model has notably lower performance on unseen data. Regression Classifiers are generally less prone to overfitting when dealing with datasets that have many irrelevant features which was our current case. They can handle a large number of features without the risk of overfitting, unlike Decision Trees. Our results also made clear the importance of choosing a good learning rate for gradient descent as if too large of a learning rate is used an optimal value may never be achieved due to overshooting, and too small of a value will be more computationally expensive to arrive at convergence. Going forward it would be interesting to perform more tests where the number of features varies or where we look at different methods of determining feature importance to further fine tune our model performance. In addition, we encountered noteworthy results when comparing the highly weighted features using Linear Regression and Logistic Regression. Although we didn't evaluate the performance of Linear Regression in this problem, it turns out that the final most important features identified by Logistic Regression make much less sense than those identified by Linear Regression. One main reason for this could be that in Linear Regression, we used ratings as labels, where higher ratings and very low ratings tend to give more polarized weights to the words used. This is not the case when we treat the ratings as binary values.

6 Statement of Contributions

Jayden performed the feature section for the 20 newsgroup data set, implemented the multi-class regression and executed experiments related to that class and dataset. Sophia performed test experiments, worked on feature selection for IMDB data set, worked on writing the report and debugged errors throughout code notebook. Negin perfomred the feature selection for the IMDB data set, implemented logistic regression class, executed experiments related to the logistic class and wrote relevant sections of the report.

References

[1] Chris Crawford. (2017). *20 Newsgroups. Kaggle. https://www.kaggle.com/datasets/crawford/20-newsgroups*

[2] Harish, B. S., Kumar, K., & Darshan, H. K. (2019, June). *Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method. International Journal of Interactive Multimedia and Artificial Intelligence (IJI-MAI).http://doi.org/10.9781/ijimai.2018.12.005*

[3] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning.*

[4] Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression.*

[5] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). *Learning Word Vectors for Sentiment Analysis.*

[6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12*, 2825–2830.

7 Appendix

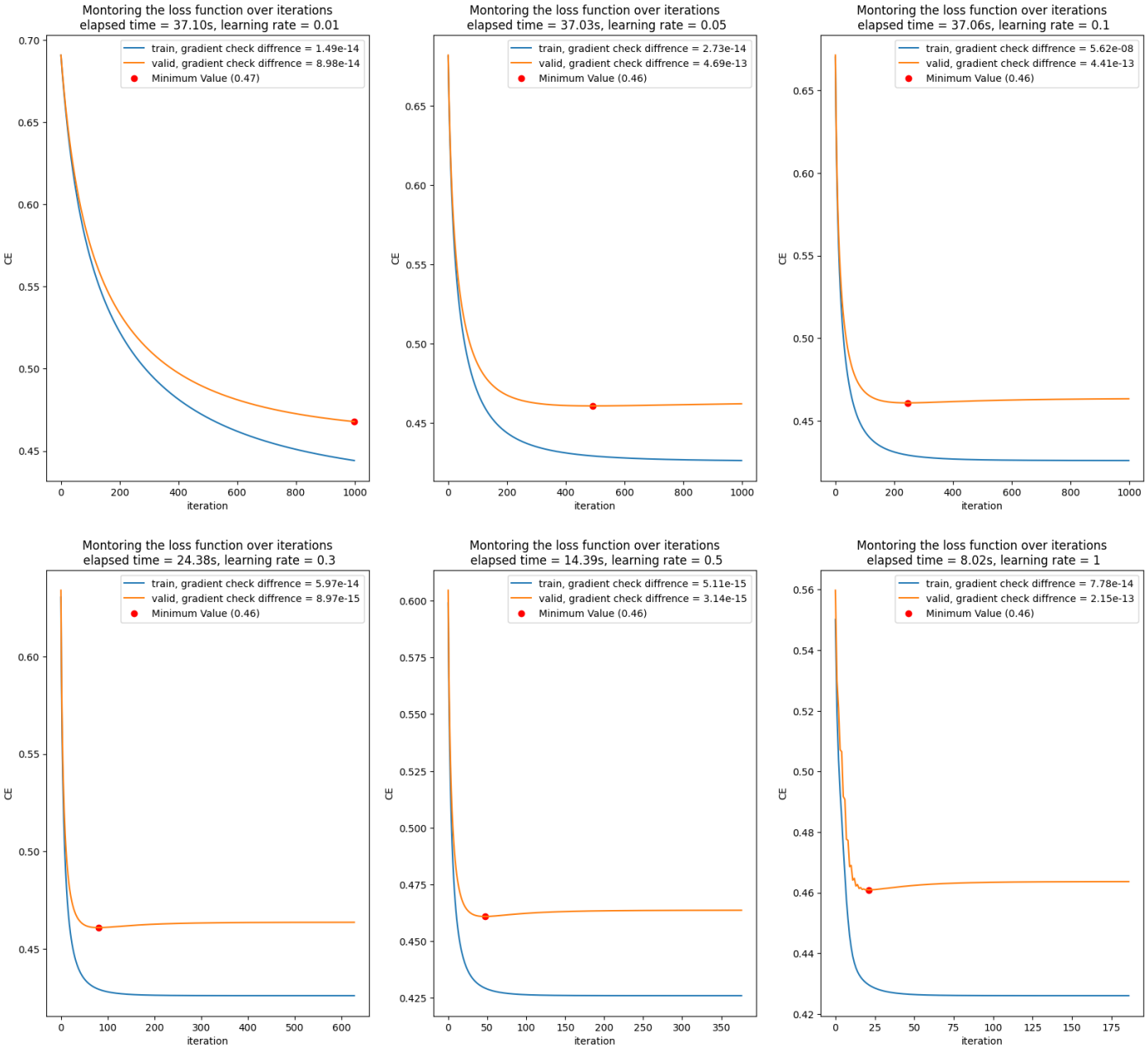

Figure 1: Logistic Regression loss function (cross entropy) and gradient monitoring for different learning rates

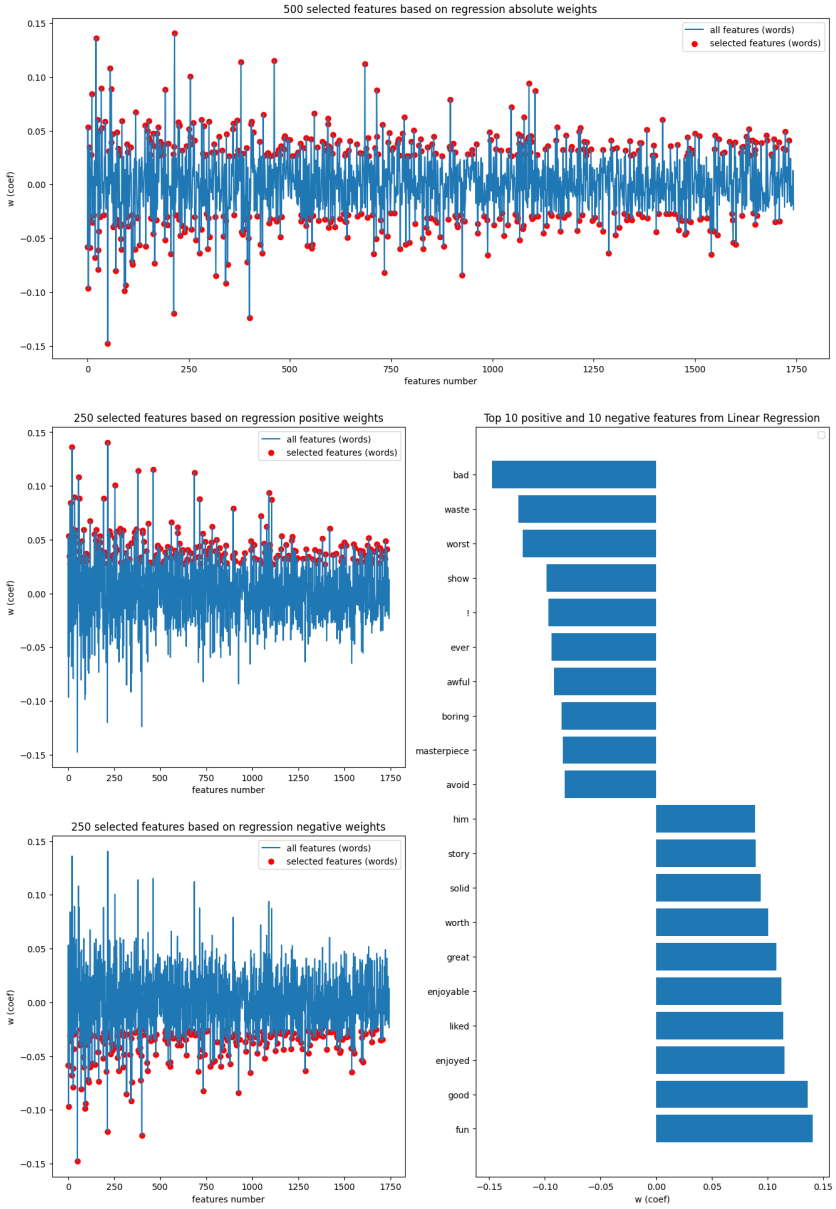

Figure 2: Feature selection using Linear Regression on IMDB dataset. the plots show highly positive and highly negative coefficients which were selected. 20 highest ones showed in the bar plot

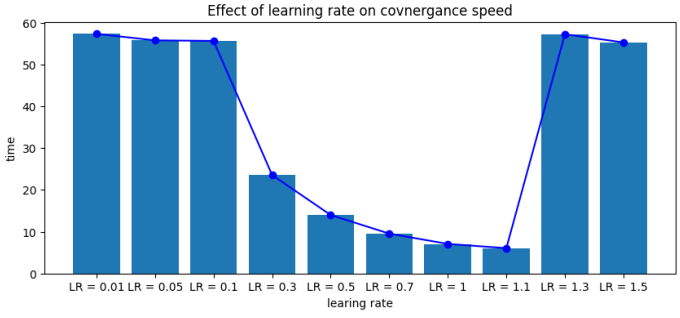

Figure 3: Effect of changing learning rate on the Logistic Regression convergence time

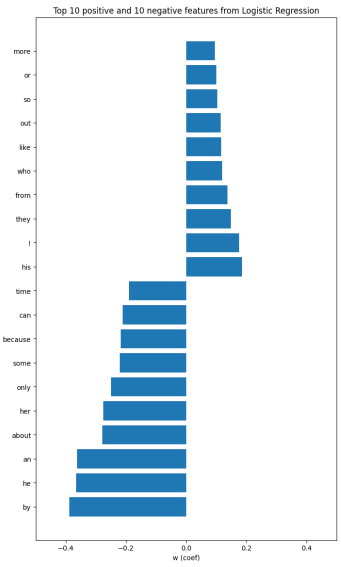

Figure 4: 10 highly negatively and 10 highly positively weighted features in Logistic Regression

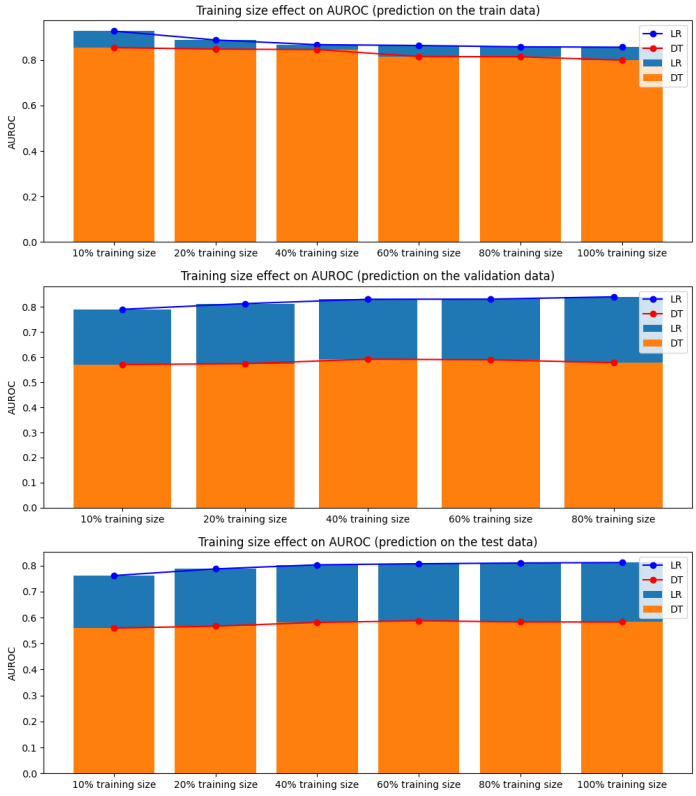

Figure 8: Logistic Regression: Effect of training set size on AUROC

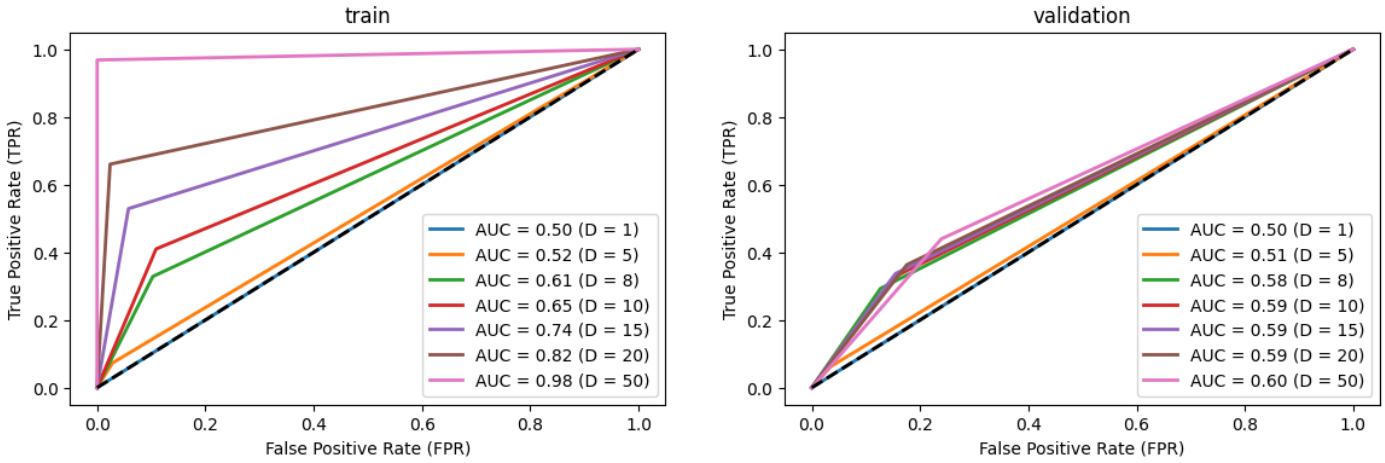

Figure 5: ROC and AUROC for varied Tree Depths on IMDB Classification using DT

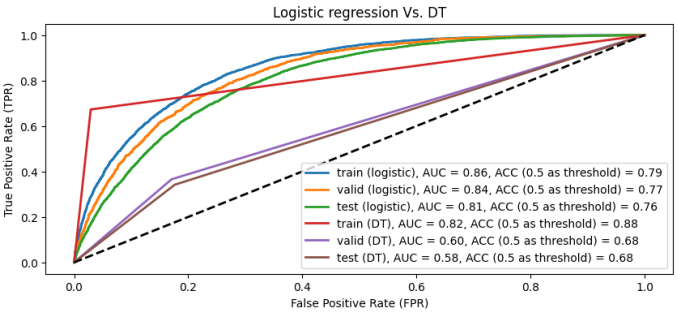

Figure 6: comparison of ROC and AUROC for Logistic Regression vs DT

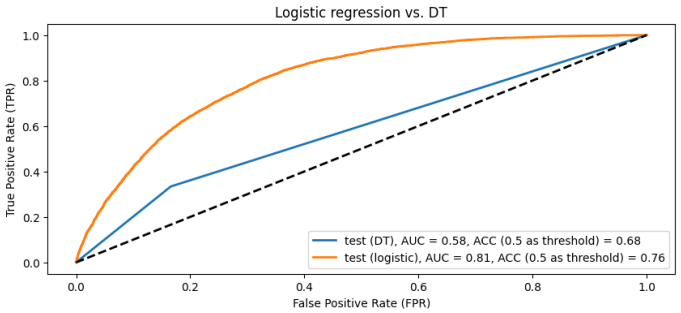

Figure 7: comparison of ROC and AUROC for Logistic Regression vs DT

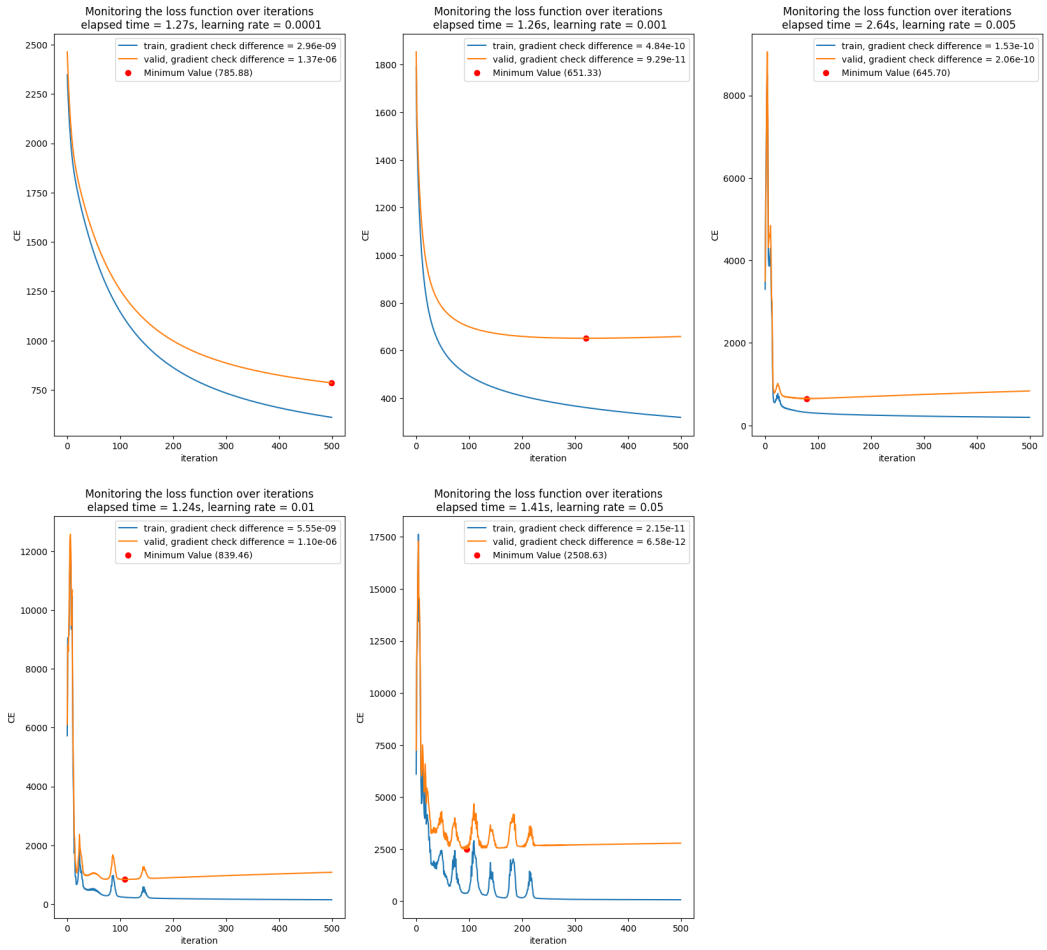

Figure 9: Multiclass Regression Loss function and gradient monitoring for different learning rates

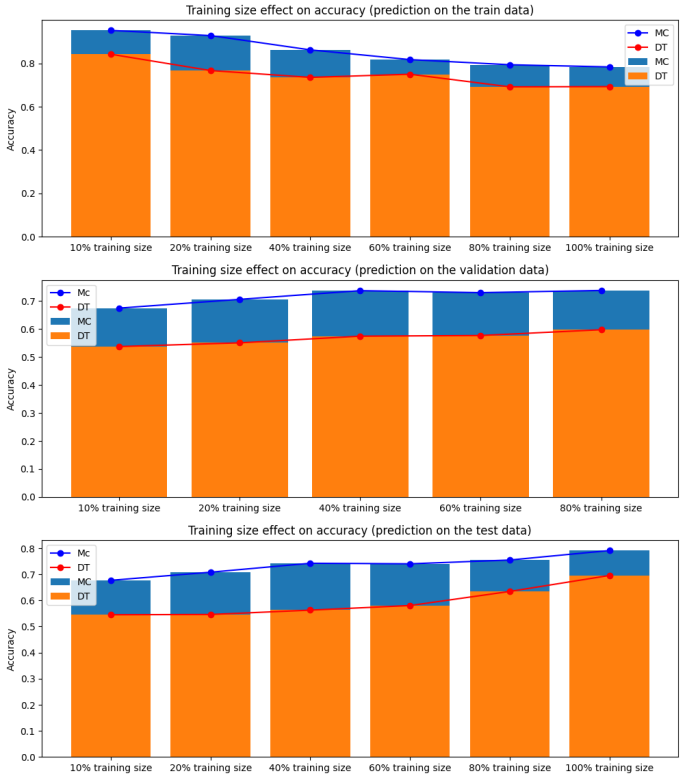

Figure 10: Multi-class: Effect of training set size on accuracy, DT comparison

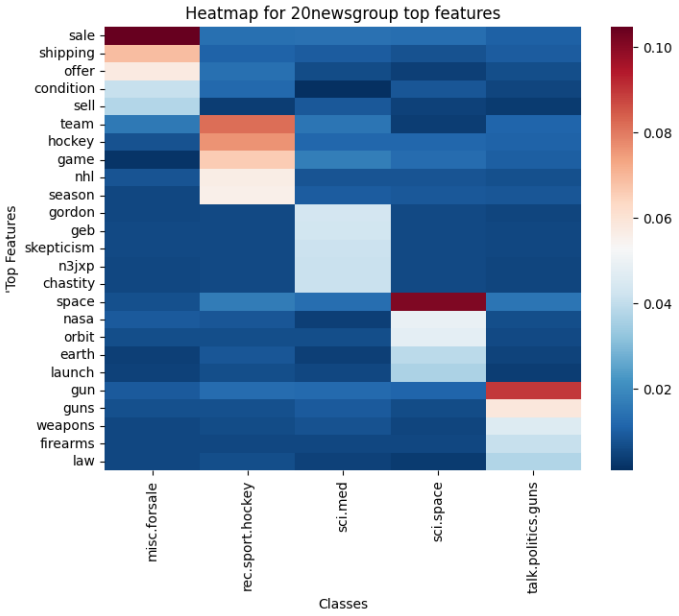

Figure 11: Heatmap for 20newsgroup top features