# COMP 551: Assignment 2 Report

Authors: Yiran Shi, Alina Tan, Ge Gao

Due date: November 8th, 2022

## Contents

1 Abstract 1

2 Introduction 1

3 Datasets 2

4 Results 3

5 Discussion and Conclusions 4

6 Statement of Contributions 5

# 1 Abstract

In this assignment, our group coded the logistic regression and the multiclass regression learning models for data classification tasks and applied these algorithms on two representative datasets previously collected from IMDB movie comments and newsgroup posts. We did intense preprocessing on these datasets to convert verbal information to mathematical features and labels and tested for the best parameter settings for the learning models. We also compared the performance of these linear methods to previous methods like K-Nearest Neighbors and Decision Tree. Eventually, we found out that these regression methods greatly outperformed the KNN and DT models within a reasonable time if we carefully control the number of features used for regression on these datasets, and they will easily generate plots that help us approach features used for classification in an intuitive way. On the other hand, regression methods could consume a huge amount of time compared to simpler methods or perform relatively worse if parameters such as learning rate, stopping criteria or feature amounts were inappropriately tuned.

# 2 Introduction

The basic tasks of this assignment are to acquire and preprocess two datasets, to implement linear regression and multiclass regression data classification models that would effectively classify these two datasets, and eventually to find out the best-performing parameter settings on these models relative to AUROC or classification accuracy on the datasets.

Both datasets are in the form of paragraphs of words upon obtained. The first dataset is a set of a total of 50,000 highly polar movie reviews collected by a group from Stanford University, split into 25,000 samples for training and 25,000 for testing. Each movie review comes with a score, where a high (7+) score is regarded as a positive rating and a low (4-) score means a negative rating. It is explained, according to the source of data, that the overall distribution of positive/negative labels is balanced. Furthermore, to reduce correlated ratings, no more than 30 reviews are allowed for the same movie. Vocabulary of words used in these reviews and "bag of words" for each review were included with the original texts, so that easier vectorization of the textual data could be done. Collectors of this data used it in their own research of a model that classified these vectorized words using semantic similarities and sentiment implications. Their model had a goal to evaluate semantic and sentimental scores of words and reviews and eventually use these scores to maximize the probability of document labels given words in the document. Eventually, their model outperformed older methods like LDA and LSA, and were competitive with recent methods on classifying positivity of movie reviews when used on several movie review datasets (Andrew L. Maas et al., 3-8). The second dataset is a built-in dataset of scikit-learn (sklearn) that comprises around 18000 newsgroups posts on 20 topics split in training and testing subsets. We were instructed to choose only 4 topics among the 20 topics, and the goal was to first convert verbal data into numeric features with the built-in CountVectorizer of sklearn, then classify the topic that each newsgroup post belonged to.

Our logistic and multiclass regression models respectively applied to these data sets gave multiple findings. Firstly, a greater number of features would significantly increase the amount of time required by these models to fit train datasets, but the increase in AUROC or classification accuracy were often minimal unless the number of features was too small to begin with. A small learning rate had similar effects, but a large learning rate would cause training loss at each iteration to oscillate around a relatively small value according to the convergence plot, showing that we might have missed the optimal point where the smallest training loss could be achieved. Secondly, these regression methods performed significantly better compared to K-Nearest Neighbor or Decision Tree classifiers even if we use a rather relaxed number for features or learning rates so that regression methods terminate within a reasonable amount of time. We also tried LASSO and Ridge regression models on these datasets, but apart from the Ridge model on the imdb data giving satisfying AUROC, none of the others gave compatible performance compared to logistic and multiclass regression models. Next, when using a fraction of the whole training dataset to fit the model, logistic and multiclass classifiers had performance that deteriorated much slower compared to KNN or DT when we decrease the fraction of training data. Lastly, by visualizing the regression coefficients associated with features, we can easily identify the features (words) that were significantly meaningful for determining whether a movie review is of positive or negative attitude, or for determining the topic that a newsgroup post belongs to.

# 3 Datasets

For the IMDB reviews data, class 0 (negative) and class 1 (positive) are equally distributed. One thing to notice is that we processed all original txt files containing movie reviews locally into a csv file, then we imported the csv file back into Google Drive to work with it, because such processes would take a long time such that Google Colab automatically terminated the run. We used the tokenized bag of words to obtain the initial features data. Since the data had 25,000 samples each for training and testing, and the vocabulary size is 89526, we had to filter out many words in several steps. First, we counted the appearances of each word in all of the reviews and removed the words that appeared in either less than 1% or more than 50% of the reviews, because these words are either too rare or too general for the purpose of determining the positivity of the review. Then, we standardized each column of the X matrix and the array containing ratings associated with the review to calculate the z-scores between each feature and the ratings. If we examine the words with the highest and the lowest z-scores at this step, we can see that they are words commonly used to express strong emotions about movies, such as "beautiful", "favorite" or "bad", "awful". And so, they make sense as negative scores indicated bad mood and positive scores indicated features with good meaning. Using these z-scores, we performed two-tailed hypothesis testing with a significance level of 0.0000001. This would leave us with 519 features. We have also tried other significance levels, but they did not improve the AUROC, so we keep the 519 features to save the runtime. Finally, we shuffled the datasets.

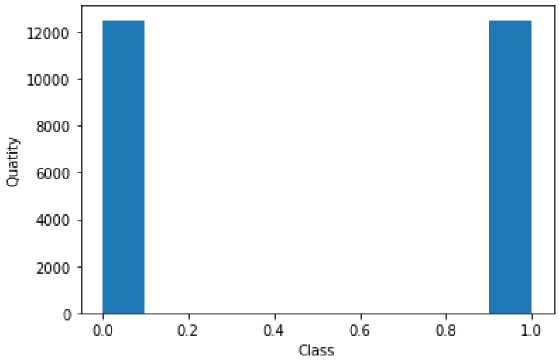

Figure 1: class distribution of training set of IMDB dataset.

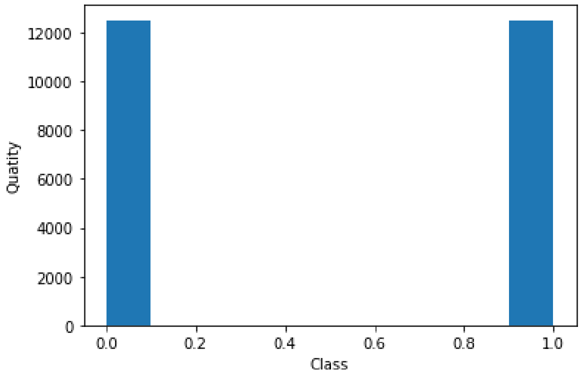

Figure 2: class distribution of test set of IMDB dataset.

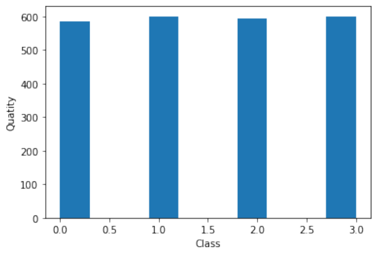

Figure 3: class distribution of training set of 20 news groups dataset.

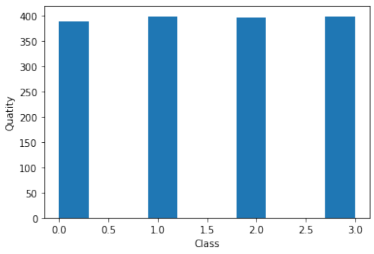

Figure 4: class distribution of test set of 20 news groups dataset.

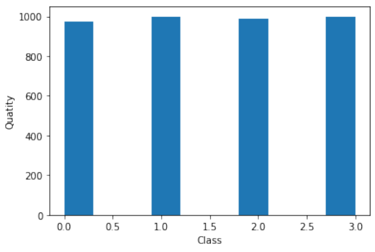

Figure 5: class distribution of the combination of train set and test set of 20 news groups dataset.

For the newsgroups posts, we selected 4 different classes (comp.graphics, rec.sport.hockey, sci.med, soc.religion.christian) from the 20 news group. The four class distributions are shown above and they are approximately equally distributed (Fig.5). To pre-process the data, we first use the CountVectorizer built into scikit-learn to obtain a numerical representation of feature values. The training set has 2377 samples and the testing set has 1582 samples. Then, we find the set of common features that is in both the training set and the testing set to prevent situations like selected features from the training set do not exist in the testing set. Next, we use one-hot encoding to convert numerical labels ranging from 0 to 3 into binary labels for 4 classes, so that the original y of n samples becomes an n time 4 matrix Y. Due to the fact that these sets also have vocabulary sizes between 32,000 and 38,000, we still need to perform feature selection. Thus, we use the same way as for the IMDB review data to eliminate rare and stopwords features, then instead of z-scores and two-tailed hypothesis testing, we calculate mutual information scores between each feature and each class label for the whole dataset. For each class label, we generated the list of the top 50 features with the highest MI scores, and eventually, we took the union of these lists to get the final set of features, which had a size of 166.

# 4 Results

After we pre-processed the datasets and implemented logistic regression and multiclass regression, we run some experiments to test how well the algorithm performed. There were different top 20 features were selected in IMDB datasets depending on which parameter we used (Fig.6). In Logistic regression with the IMDB dataset, the small perturbation showed 5.901e-11, and when we monitored the cross entropy at each iteration, the convergence plot all converged, which approached zero (Fig.7.A). The convergence plot showed that the learning rate of 0.1 computed smooth curve and the cross entropy loss decreased faster. Therefore, we would like to choose learning rate = 0.1 for further investigation (Fig.7.A). With the multiclass regression with 20 news groups, the small perturbation test showed the result around 1.321e-13 and the convergence plot showed the cross entropy keep decreasing as iteration increased (Fig.7.B). The convergence plot indicated that both the learning rate (lr) of 0.005 and the lr of 0.0001 could generate a smooth curve. We would like to choose learning rate = 0.005 for further investigation since it takes less running time (Fig.7.B). Next, we compared the AUROC score of logistic regression, K-nearest neighbours (KNN) (sklearn) and decision tree (sklearn), which were s 0.94, 0.7, and 0.73, respectively (Fig.8). In this case, logistic regression gave us the highest AUC score which means the most accurate among three. In addition, the AUROC of logistic regression on the test data remained the highest compared to KNN and decision tree across all different sizes of training data (Fig.10.A). Similarly, the accuracy of predicting classes with multiclass regression was the highest among the three models in all different sizes of training data (Fig.10.B). For the 20 news groups with multiclass regression, the heatmap indicated that different classes correlate with different features (Fig.9).

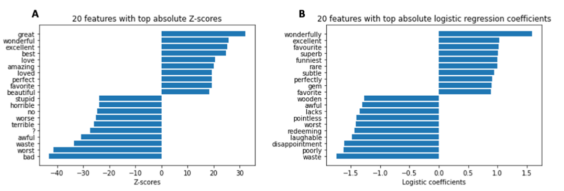

Figure 6: The bar plots were the top 20 features top 10 most positive and top 10 most negative) based on the (A) absolute z score and (B) absolute logistic regression coefficients of the IMDB datasets.

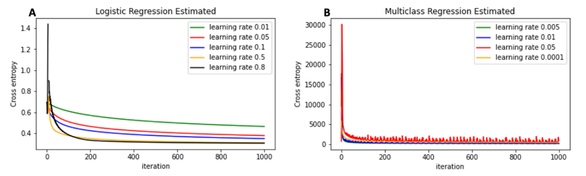

Figure 7: (A) was the convergence plot of logistic regression with different learning rates in the IMDB dataset. (B) was the convergence plot of multiclass regression with different learning rates. The x-axis is the number of iterations, and the y-axis is cross entropy at each iteration in the 20 news groups dataset.

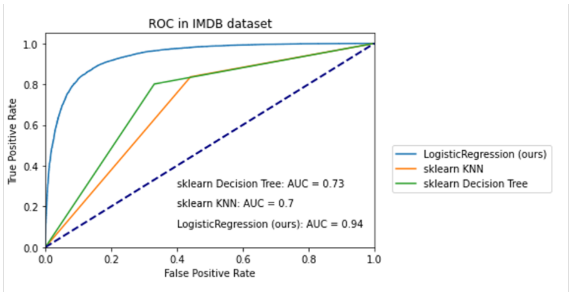

Figure 8: This plot contained receiver operating characteristic curve (ROC) curves for logistic regression, K-nearest neighbor (sklearn) and decision tree (sklearn) on the IMDB test dataset. The AUC stands for area under the curve.

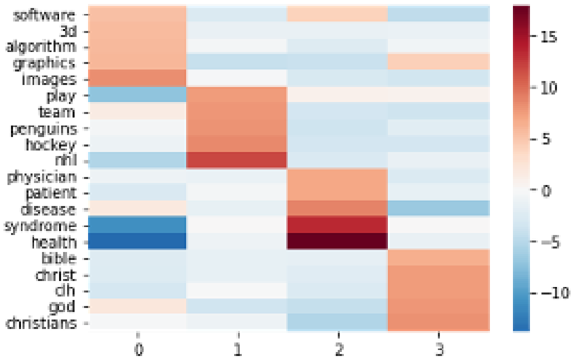

Figure 9: This was a heatmap showing the top 5 most positive features as rows for each class as columns in the multi-class classification on comp.graphics, rec.sport.hockey, sci.med, soc.religion.christian from the 20-news groups dataset

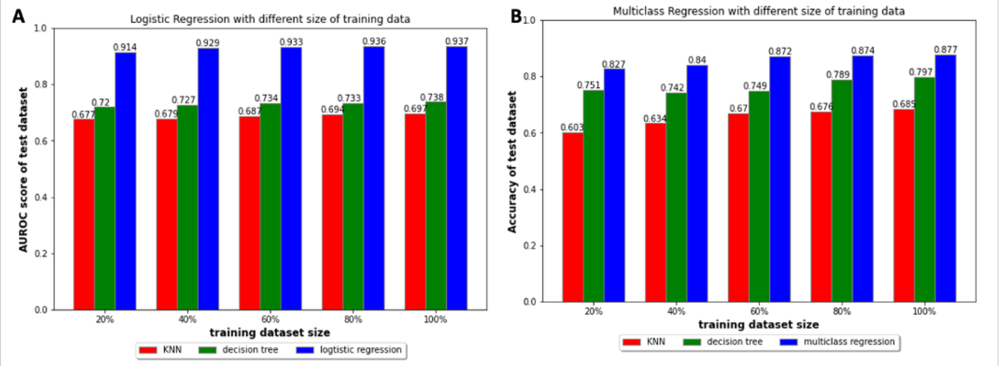

Figure 10: (A) A bar plot showed the AUROC of KNN (red), decision tree (green) and logistic regression (blue) on the test data (y-axis) as a function of the 20%, 40%, 60%, 80%, and 100% training data (x-axis). (B) A bar plot showed the accuracy of KNN (red), decision tree (green) and multiclass regression (blue) on the test data (y-axis) as a function of different size of training data (x-axis).

Besides, we also tried different ways of feature selection. For the IMDB dataset, we changed the significance level from 1e-6 to 1e-5, 1e-4 and1e-3. The selected features increased from 520 to 596, 685 and 779 features, respectively. However, the different number of features did not change the AUROC of the logistic regression model, all of them resulted in an AUROC score of 0.94. Reducing the number of features of the training dataset in 20 news groups did not mean decreasing the accuracy of the multiclass regression as with 166 features the accuracy was 0.874, with 100 features the accuracy was 0.882 and with 50 features 0.743. In addition, we used linear regression to predict the ratings of IMDB data. The mean squared error (MSE) was 11.87 which was a large number. We tried regularization for both datasets. For IMDB, the AUROC for Lasso (L1) Regression was 0.673 and for Ridge (L2) was 0.934. There was no significant improvement compared to the not regularized one, 0.940. Similarly, for 20 news groups, the accuracy for Lasso (L1) Regression was 0.598 and for Ridge (L2) was 0.731. Both were lower than the non-regularized multiclass regression which was 0.877.

## 5 Discussion and Conclusions

In this assignment, our goal is to select the best model to classify IMDB dataset and newsgroup dataset. We fitted logistic regression on IMDB dataset and multiclass logistic regression on newsgroup dataset. We did not use validation set for both two datasets since we found that different learning rates would not improve AUROC score and test accuracy significantly. And so, we choose the learning rate of 0.1

to fit for logistic regression and the learning rate of 0.005 to fit for multiclass logistic regression.

For IMDB dataset, we first fit the logistic regression. By calculating the small perturbation and monitoring the cross-entropy as a function of iteration, we show a very small perturbation and the convergence plot shows that the cross-entropy cost converges to 0. These two evidences show that our gradient is computed correctly. Then we compared the AUROC score of logistic regression, KNN and Decision Tree using 20%, 40%, 60%, 80%, and 100% respectively. As the barplot shows, the AUROC scores of logistic regression are always the highest. This shows that the logistic regression has the best performance among the three models. Comparatively, KNN always has the lowest AUROC score. We think the reason of this low AUROC score is that we used 519 features without standardization to fit for the model, and KNN is sensitive to feature scaling and high-dimensional data. This might lead to the low AUROC score. In addition, we have also tried lasso regression and ridge regression, but the AUROC scores are still lower than logistic regression's, and so we decided not to use these two models. Lastly, we fit the linear regression to predict the ratings. The calculated MSE is quite large which indicate that the predictions are not good. To further check the performance of the predictions, we plot the predicted ratings distribution and the true rating distributions. We found the true ratings are distributed at the two tails that is biased, but the predicted ratings are distributed in the center. Linear regression refused to predict data with larger bias, we need to use a more complex model instead to predict the ratings.

For the newsgroup data, we first fit the multiclass logistics regression model. The small perturbation and the convergence plot both show that our gradient calculation is correct. The cross-entropy cost is converging. Then we compared the test accuracy of multiclass regression, KNN, and Decision tree. We generated the similar pattern as the logistic model comparison. Multiclass regression always has the highest test accuracy and KNN always has the lowest test accuracy. The reason of the low accuracy of KNN is that KNN is sensitive to feature scaling and high-dimensional data. Additionally, we also try to fit lasso regression and ridge regression. The test accuracies of both models are lower than the test accuracy of multiclass regression. And so, we would not use lasso and ridge to classify the newgroups data.

In conclusion, by comparing the performance of different models, we finally found that the logistic regression model classified the IMDB dataset the best, and the multiclass logistic regression classified the newsgroup dataset the best. But since we did not standardize both datasets, the performance of KNN might be underestimated. To further improve KNN, we could select less significant features and standardize the dataset. In addition, the simple linear regression model did not predict the ratings of IMDB very well, we could also try to use regulariztion model to predict the ratings for further investigation.

# 6 Statement of Contributions

All three of us wrote the implementation of the Logistic Regression and the Multiclass Regression algorithm. We did the experiments and datasets together. For the report, Yiran Shi is responsible for the abstract, introduction and datasets; Ge Gao is responsible for the discussion and conclusion, and Alina Tan is responsible for the result and statement of contributions.

# References

[1] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).