COMP 551 - Assignment 2 - Group 11

Yian Bian
yian.bian@mail.mcgill.ca

Yijia Jing
yijia.jing@mail.mcgill.ca

Zeyu Li
zeyu.li2@mail.mcgill.ca

Abstract

In this project, we investigated the performance of linear classification models on two benchmark datasets, the "IMDB review" and "20-news-data". Through the evaluation of these two data sets, we apply the logistic regression and multiclass regression models to them and did some statistics for the results. With some optimization for the model itself, such as hyperparameter tuning, graphs generated for the results, and regulation, we finally found that both the logistic regression and multiclass regression approach achieved better accuracy than the KNN method provided by the sklearn library and was significantly slower to train. More specific details will be discussed in the following paragraphs.

1 Introduction

In this project, we deal with two groups of data. The first dataset is the IMDB Review. This dataset contains movie reviews along with their associated binary sentiment polarity labels. Many people have analyzed this dataset before, Qaisar[1] used the LSTM classifier which is based on the RNN algorithm to analyze the sentiments of the reviews; Pang and Lee[2] applied a meta-algorithm, based on a metric labeling formulation of the problem. In the paper, we will utilize the continuous rating score for feature selection, and then run the logistic regression for the binary classification of this data, using the feature selected to predict the sentiment of the review.

The second dataset is the 20 Newsgroups data set. This dataset is a collection of approximately 20,000 newsgroup documents, partitioned evenly across 20 different newsgroups, each corresponding to a different topic. This dataset is often used to conduct experiments in text applications such as text classification and text clustering, for example, Akef and Arango[3] used Latent Dirichlet Allocation to perform topic modeling and discover the semantic structure of the provided dataset, by examining word statistical co-occurrence patterns. In this report, we will use multiclass regression to perform the text classification based on the feature we selected through mutual information score.

By conducting these two experiments, we find that the logistic regression and multiclass classification models perform much better than KNN in terms of AUROC and accuracy. And the training size for these two models will impact the performance of the prediction, generally, more data is better, but it's not always the case. Also, regularization sometimes may improve the performance, but it also can drag down the accuracy, which may depend on the situation.

2 Datasets

2.1 IMDB Reviews

This dataset contains 50,000 movie reviews which consist of 25,000 highly polar movie reviews for training, and 25,000 for testing, labeled by sentiment where 0 represents negative and 1 represents positive.

Since the vocabulary size is large, we first counted the frequency of occurrence of each word across documents and filtered out stopwords, which appeared in more than 50% of the documents, and rare words, which appeared in less than 1% of the documents. Then, we calculated the z-score associations with the rating using Simple Linear Regression Hypothesis Testing. Based on how large the absolute z-score is, i.e., how significant a feature is, we picked the top 150 features to train our models.

By parsing the already-tokenized bag of words (BoW) features provided, we obtained the X matrix used to fit our model, where each row contains the frequency of occurrence of each word in a specific review. The response vector y is the binary sentiment class. We have 50% of the data predetermined for training and the other 50% for testing. We further split the training set into 70% for training and 30% for validation to find the best learning rate.

2.2 20 Newsgroups

This dataset contains 18,000 newsgroups posts on 20 topics of which 2,377 posts on four distinct topics are chosen: *comp.graphics*, *rec.sport.hockey*, *sci.med* and *soc.religion.christian*.

To filter out noisy features among all 27,463 features, we again excluded the stopwords and rare words using 50% and 1% as the thresholds of frequency, respectively. We calculated Mutual Information (MI) and selected the top 100 feature words for each class based on their importance measured by MI. In the end, we used a total of 297 features to train our models.

We utilized CountVectorizer to count the occurrence of each word in each document, which is transformed into the input matrix X. The response vector y, which represents one of the 4 topics numbered from 0 to 3, was one-hot encoded. We split the dataset into 42% for training (70% of the original training set), 18% for validation (30% of the original training set), and 40% predetermined for testing.

3 Results

3.1 IMDB Reviews

3.1.1 Hyperparameter Tuning

The learning rate α determines how rapidly we update the parameters. A too-small learning rate leads to an excessive number of iterations while one that is too large causes overshooting. We compared the accuracies using different learning rates on the validation set (shown in Table 1). Surprisingly, we found that for this specific dataset, a change in α does not cause a drastic change in performance. Since α=0.1 yields the best accuracy of 84%, we chose 0.1 as our final learning rate.

Table 1: Accuracies with different learning rates

α	0.05	0.1	0.25	0.5
Accuracy (%)	83.74	84	83.91	83.95

3.1.2 Model Evaluation

Using the best learning rate of 0.1, we evaluated the accuracy on both the training and testing set, which are 84.33% and 84.12%, respectively. This extreme similarity indicates that there is no sign of overfitting.

We checked the gradient using the small perturbations technique and obtained the result of 1.33e-17, which is a very small difference and we thus concluded that our cost function and gradient worked as expected.

The convergence plot shown in Figure 1 helped us better understand how the logistic regression model with the α=0.1 converges throughout iterations. From the plot, we can see that in the first 1,000 iterations, the cross entropy cost drops drastically in each iteration, while the rate gets much slower from iteration 1,000 to 4,000. The cost barely changes after 6,000 iterations, indicating that we are very close to convergence.

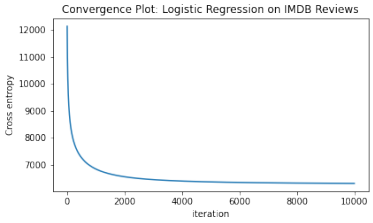

Figure 1: Convergence Plot for Logistic Regression

To understand the difference between our Logistic Regression model and the K-NN model better, we used the Receiver Operating Characteristic (ROC) Curve shown in Figure 2 to make comparisons. This is a graphical plot that is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). The higher the Area Under the Curve (AUROC), the better the model. We observed that our logistic regression model has an AUROC of 0.84 while the KNN model has only 0.67. We can therefore conclude that our model has a better performance than the sklearn KNN when applied to the IMDB Review dataset. The possible reason is that given a large number of features (150 in our case), underfitting occurs for the K-NN model due to the curse of dimensionality.

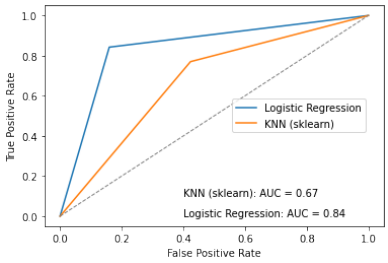

Figure 2: ROC for Logistic Regression and KNN

3.1.3 LASSO and Ridge

In Table 2 we have two models that do not use regularization, one of them does not use any validation data, and the other uses cross-validation to train the model. From the result, it's quite clear that Cross-validation performs much better than the original one since it gives your model the opportunity to train on multiple train-test splits, which gives you a better indication of how well your model will perform on unseen

Table 2: AUROC score of different models

Regularization	AUROC Score (%)
original (no validation)	74
validation (cross validation)	91.82
Ridge	91.56
LASSO	91.83

data. Since the models using regularization are both using cross-validation, so we will compare them with the regular model with cross-validation. By comparing the AUCROC score, we observe that LASSO regularization does better than Ridge since Lasso Regression has the ability to nullify the impact of an irrelevant feature in the data, meaning that it can reduce the coefficient of a feature to zero thus completely eliminating it and hence is better at reducing the variance when the data consists of many insignificant features, whereas Ridge can only shrink the size. Also, we can observe that LASSO regularization improves the performance while Ridge drags down the performance.

3.1.4 Top Features

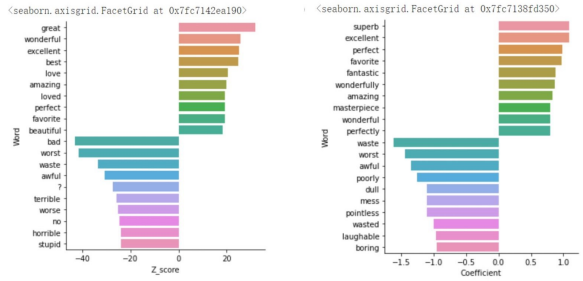

Figure 3: Top feature selected using z-scores and regression coefficients

Figure 3 above shows the top feature selected by the top z-scores which is calculated by running a linear regression(on the left side) and the top coefficients of the logistic regression(on the right side). By comparing the words on the y-axis, we can find that most of the words selected by SLR make sense, however, there are still words that can not represent the sentiment of the review like "?" and "no". But the words picked by logistic regression are all very sentimentally strong. Also, there are words that are selected by both approaches, like "wonder", "perfect", "amazing" etc, but they give them different weights, for example, the linear regression model thinks "amazing" is more related to a positive review than "perfect" and " favorite" do, whereas the logistic regression does the opposite. Given all the information these two plots give, we think logistic regression classifies more accurately than simple linear regression.

3.1.5 Effect of Training Set Size

We would also like to investigate the effect of training set size on the model performance. We took subsets of the training set, which are 20%, 40%, 60%, 80%, and 100% of the original training data, to fit both Logistic Regression and KNN models. We then evaluated the accuracy on the same test set. The AUROC of logistic regression is consistently higher than that of KNN regardless of the size of the training data. The behaviors are surprisingly different for the two models according to Figure 4. For logistic regression, the peak is at 60% of training data with an AUROC of 0.8421 while for KNN, the accuracy increases as we increase the size of training data and reaches its peak at 100% of training data with an AUROC of 0.6734.

Therefore, it should not always be believed that the larger the size of training data, the better the performance. Noise and overfitting can play a role in lowering the accuracy. The suitable size should be examined on a case-by-case basis.

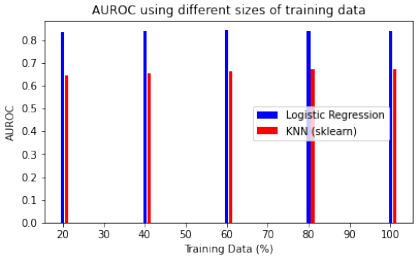

Figure 4: AUROC for Logistic Regression and KNN

3.2 20 Newsgroups

3.2.1 Model Evaluation

With a learning rate of 0.00025, we fit our multiclass logistic regression model with the training set and continuously monitored the cross-entropy loss on both the training set and the validation set as we go through more iterations. Figure 5 is a convergence plot showing the trend of such change in the loss. In the first 100 iterations, both the losses on training and validation sets drop by a lot in each iteration. However, after that, while the loss on the training set continues to decrease slowly, the loss on the validation set almost stops changing, suggesting a possibility of overfitting. We again utilized small perturbations to evaluate our loss and gradient implementations. After checking the gradient, we obtained a very small difference of 2.96e-14, suggesting that our cost function and gradient worked as expected.

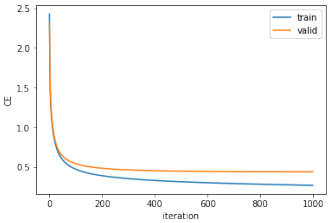

Figure 5: Convergence Plot for Multiclass Regression

3.2.2 Effect of Training Set Size

We also changed the size of the training set size, to see whether the size will have an impact on the accuracy of the prediction. So we took 5 subsets of the training data randomly, which are 20%, 40%, 60%, 80%, and 100% of the original training data, and then we fit these 5 groups with the Multi-class regression separately, making other parameters unchanged. By looking at the blue bar in Figure 6, we can observe that the accuracy increases as we contain more data in our training dataset. After investigating the impact of size on the accuracy, we also compare our multinominal regression with the K-Nearest Neighbor algorithm. The increasing accuracy of the KNN model again confirms that the size of the training data affects the accuracy of the test data. However, if we compare these two models, we can see that the accuracies generated by the KNN algorithm are nearly half of those generated by the multinominal regression. So we conclude that although the KNN algorithm is a good algorithm to classify data, in this example, the multi-class regression performs better.

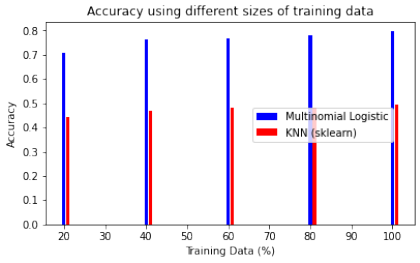

Figure 6: Accuracy for Logistic Regression and KNN

3.2.3 Top Features

The figure below shows the top 20 features selected by using mutual information and multi-class regression separately. For "comp.graphics" group, the words selected contain "graphics", "software", "code" etc; for "rec.sport.hockry" group, the words are "team", "game", "hockey" etc; for "sci.med" group, the words are "disease", "medical", "medicine" etc; and for 'soc.religion.christian" group, the words are "god", "jesus", "christ" etc, which are all related to each topic of the group. We can see that although most of the features selected by mutual information make sense, there are still unrelated words like "he", "who", "NHL" etc, which shows the limited ability of this method. Compared with the words selected by the mutual information, the features selected by the multi-class regression which is shown on the right side make much more sense, they hardly contain personal pronouns which appear on the left-hand side. We can find the obvious relationship between the words and their group. By analyzing the words, we conclude that multi-class regression has a better ability to select important features than mutual information does, however, mutual information has already done a good job. Figure 8

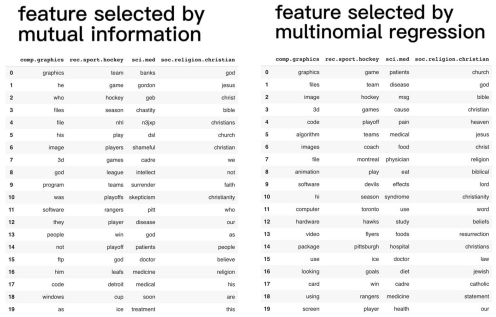

Figure 7: Top feature selected using mutual information and regression coefficients

shows the heatmap of regression coefficients of the top 5 words from each class. Each word has a large (light-colored) coefficient for the class that it correlates to the most while having small (dark-colored) coefficients for the other three classes. We observed that there are four noticeable light-colored blocks in the diagonal. This demonstrates that a good classification model identifies the association of a class and an important feature well, and gives small weights to a feature that is not very related to classes. For example, "graphics" is highly correlated to *comp.graphics* with a weight of 2.4 while it is not significant at all for all three other classes with weights all less than 0.3.

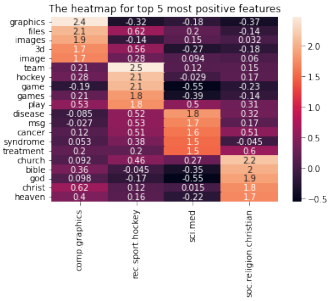

Figure 8: Heatmap for top 5 most positive features

3.2.4 LASSO and Ridge

Table 3: Accuracy of different models

Regularization	Accuracy (%)
original (without validation)	79.58
validation (with cross validation)	76.74
Ridge	64.79
LASSO	74.97

The differences between these four models are the same as the IMDB dataset. However, the results are quite different. Here, the original model has the highest accuracy, whereas the model without regularization but with cross-validation is the second, and the model with LASSO regularization is the third. The reason why the original model is the highest may be because we only focus on the accuracy and ignore the loss. Since the accuracy is quite near for the original model, validation model, and LASSO model, the latter two may have lower losses given that we have improved our model. Another reason may be the "k" for the cross-validation, here we randomly choose 5, but it may not be the best one for the K-fold-cross-validation for this dataset, we can further tune the hyperparameter for both regularization and cross-validation.

4 Discussion and Conclusion

4.1 Key Takeaways

For our two datasets which involve text classifications that are intrinsically complicated, the logistic regression and multiclass classification models perform much better than KNN in terms of AUROC and accuracy. Because there is no such thing as the best classifier, we could evaluate the performance of different models and choose carefully based on different metrics.

Techniques such as z-scores from linear regression and mutual information gave us a nice intuition on the most important words, and filtering based on these pieces of information played a vital role in improving the accuracy. The top features obtained from the regression weights of both the logistic regression and multiclass classification models generally make sense given the context of the dataset, which is also a good metric to evaluate the model in addition to accuracy and AUROC.

The size of the training data is one of the factors that impact the model performance. Generally speaking, providing more data helps with accuracy. However, it is not always the case that we could just improve the performance by blindly feeding in more data, as proved by our experiments. Moderately large datasets are ideal for training our Logistic and Multiclass Regression models.

Regularization with LASSO and Ridge can generally help with the performance, but in some cases, as experimented on our datasets, it does not contribute as much to the overall performance. Therefore, we should choose such techniques based on our needs and performance evaluation.

4.2 Future Investigation

We noticed that the top features we selected using both the two approaches - z-score/mutual information and regression coefficients - include stopwords that are not specific to a certain class. However, filtering out too many words with high frequency could lead to the accidental exclusion of important features. Therefore, we could experiment with different thresholds for stopword filtering and find the best one based on either accuracy or top feature observation, as 50% would certainly not be the best threshold in all different contexts.

We performed multiclass regression experiments on a partial 20 newsgroups dataset which only contains 4 classes. We would be interested to know whether the increase in the number of classes could have a noticeable negative impact on the accuracy. Therefore, we would like to also design experiments on a bigger dataset with 20 classes either with a similar number of features or more features as we see fit.

We performed cross-validation and regularization on both datasets, and the results are not quite satisfying. One reason may be the parameter we use since we randomly choose the lambda for the regularization and k for cross-validation. Therefore, if we tune our hyperparameter for our model in the future, we may get better results.

References

[1] Saeed Mian Qaisa, Effat Univeristy.Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory, November 2020

[2] Bo Pang and Lillian Lee, Cornell Univerisity. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales, 2005

[3] Islam. Akef and Suan S. Munoz Arango. Mallet vs GenSim: Topic modeling for 20 news groups report, April 23, 2016