Assignment 2: Classification of Textual Data

Terrance Yan, Yunjia Zheng, Annie Kang

COMP 551 Winter 2024, McGill University

Abstract

As we navigate through a world overflowing with digital text—from stock market forecasts to healthcare alerts—understanding this vast amount of information becomes crucial.[1] This project tackles the challenge of comparing advanced machine learning models to make sense of complex data, aiming to find more accurate ways to see through textual data. In this project, we mainly investigated the comparative analysis of linear models for classification problems —logistic regression and multiclass regression—against traditional decision trees (DT) on two benchmark datasets, with a significant emphasis on the role of feature engineering in model performance. Through data cleaning and feature selection methods, we enhanced the adaptability and effectiveness of our models for each dataset. For the IMDB dataset, we utilized multivariable linear regression for feature discernment, while for the second dataset, information gain ratios were employed to identify the most salient features. Our comprehensive investigation revealed that the logistic/multiclass regression models not only outperformed DT in terms of accuracy but also exhibited remarkable stability across different volumes of training data. Furthermore, additional experiments of comparing with Ridge Regression and predicting for more classes are conducted, indicating the models are also good enough for a vast range of classification scenarios. Although the models perform well, special attention should be paid on choosing proper learning rate. All the explorations above validate the effectiveness of logistic and multiclass regression in analyzing textual data, highlighting their ability to improve classification accuracy and adaptability when hyperparameters are chosen carefully.

1 Introduction

In this assignment, we use two datasets: the IMDB Reviews dataset for movie feedback analysis and the 20 Newsgroups dataset for sorting text into topics. These form the basis for our exploration into text classification using machine learning models.

The IMDB dataset [2] is a collection of movie reviews, designed for binary sentiment classification. It comprises 50,000 reviews split evenly into a training set and a test set, with balanced subsets of positive and negative reviews. This dataset, originating from the Internet Movie Database (IMDB), has become a standard benchmark in the field of natural language processing (NLP) for sentiment analysis tasks, offering rich text data for modeling and evaluation of various machine learning algorithms. A study applied sentiment analysis to the IMDB reviews dataset using four algorithms: Decision Trees, Random Forest, Gradient Boosting Classifier, and Support Vector Machines, finding that the Support Vector Machine with TF-IDF feature extraction achieved the highest accuracy of 89.55%.[3]

The 20 Newsgroups dataset[4] is a popular collection used for text classification and clustering, comprising around 20,000 newsgroup documents evenly distributed across 20 different newsgroups. It's accessible through the scikit-learn library, offering a diverse set of documents that range from sports discussions to technology reviews, making it an ideal benchmark for machine learning algorithms in natural language processing tasks. Research developed a probabilistic approach using a Naive Bayesian classifier for text classification, significantly improving the accuracy and efficiency of sorting text documents into categories on the 20 Newsgroups dataset.[5]

Following our exploration of the IMDB Reviews and 20 Newsgroups datasets, the next phase of our assignment involves the hands-on implementations of Logistic and Multiclass Regression models from scratch. This task is twofold: firstly, to construct and assess a Logistic Regression model using the IMDB dataset for binary sentiment classification, and secondly, to implement and evaluate a Multiclass Regression model on a subset of the 20 Newsgroups dataset, focusing on five distinct categories for classification. A key step in our task is checking our models' gradients with small changes to confirm that our gradient descent optimization method is accurate and reliable. The evaluation of our Logistic Regression model will be measured by its ROC curve and AUROC values, while the Multiclass Regression model's success will be gauged through classification accuracy. Furthermore, we aim to benchmark our models against Decision Trees to contextualize their performance.

To analyze the results of our exploration into linear classification, we undertake a series of experiments focusing on the IMDB and 20 Newsgroups datasets. We begin by identifying key features in the IMDB dataset using simple linear regression, spotlighting the top contributors to positive and negative sentiments. Subsequently, we implement binary and multiclass classification models from scratch, comparing their efficacy against Decision Trees through metrics like AUROC for binary classification and accuracy for multiclass scenarios. Further analysis includes evaluating model performance across different training dataset sizes to understand the impact on accuracy. This project's analysis reveals that logistic and multiclass regression models significantly outperform traditional decision trees in accuracy and stability across various training datasets, supported by advanced feature engineering techniques.

This assignment seeks to advance our comprehension and application of machine learning in text classification by developing and evaluating logistic and multiclass regression models against traditional benchmarks, highlighting the significance of feature engineering and model optimization in improving predictive accuracy.

2 Datasets

Our data process is mainly focused on feature engineering due to the intrinsic property of the data per se, whose validation is supported by the learning algorithm and the statistical techniques. First, we do the basic data cleaning by filtering out the intuitively undiscriminating features including the stop words and the rare words demonstrated useless for most times[6]. We followed the selecting criterion offered by the guidance such that the words with showing rates less than 0.01 and bigger than 0.5 ought to be eliminated. Such a procedure had been done for both dataset 1 and dataset 2. After the preliminary data cleaning, we implement two different techniques for feature selection on datasets 1 and 2 respectively.

For dataset 1, we constructed a multi-variable linear regression by batch gradient descent, acquiring all the regression coefficients and doing the selection based on that for feature selection. We plotted the result in two different spaces(graphs), the RC-RFA(Regression Coefficient - Relative Frequency of appearance) and the ARC-RFA(Absolute Regression Coefficient - Relative Frequency of appearance), in the scatter plot format. We first scrutinized the top and bottom 20 features by dying them on our spaces, raising up an interpretation bearing out our intuition that top features tend to have lower RFA since they provide more peculiarity and heterogeneity. Then, we took the mean of RC in RC-RFA as a measure of systemic error, which shows the selected features are balanced so no systemic error is expected. We further discussed the scaling(sizing) bias inherent in our selected features by overlapping the histogram and came up with a cautious conclusion that the menace from scaling bias cannot be denied and ignored. Thus, supported by the aforementioned discussion, the separating hyperplanes in the ARC-RFA space are utilized to acquire the data points by portioning out an acceptance domain leading into some features with lower ARC. We took all the points falling in the domain as the desired features amounting to around 300.

For dataset 2, we implemented the information gain ratios(adjusted mutual information/IGR) and used the corresponding operations to acquire the IGR of all the features. The reason for us to choose IGR is its informational significance and robustness confronting scaling bias.[7] Then, we plotted them in scatter. We then selected the top 200 features with the highest IGR in the training set. To justify our strategy, we used IQR-outlier analysis, demonstrating that our selection is statistically significant and has good properties. Since word bags differ in testing and training set, we then performed an intersection operation utilizing a set container to get the finally selected 153 features.

Corresponding reference features and label matrices were then created and denoted clearly in the reference section. Besides the aforementioned content, The data preprocessing also offered insight into the choice of performance measurement: AUROC or ROC should be selected as the desired measure.

3 Results

First we show the top 20 features on the IMDB data in Figure 1 and 2, where Figure 1 shows the top features from Simple Linear Regression(SLR) and Figure 2 shows the ones from Logistic Regression(LR). Although the features are not the same from the two methods, it makes sense because it was aimed at predicting ratings using SLR compared to finding the most relevant features for good/bad movies using LR. But in general, we can see clear correlation between the top features for both method with the label. For example "recommended, witty,excellent" are among the top 10 positive features for predicting ratings and "worst,bad,waste" are for predicting bad movies.

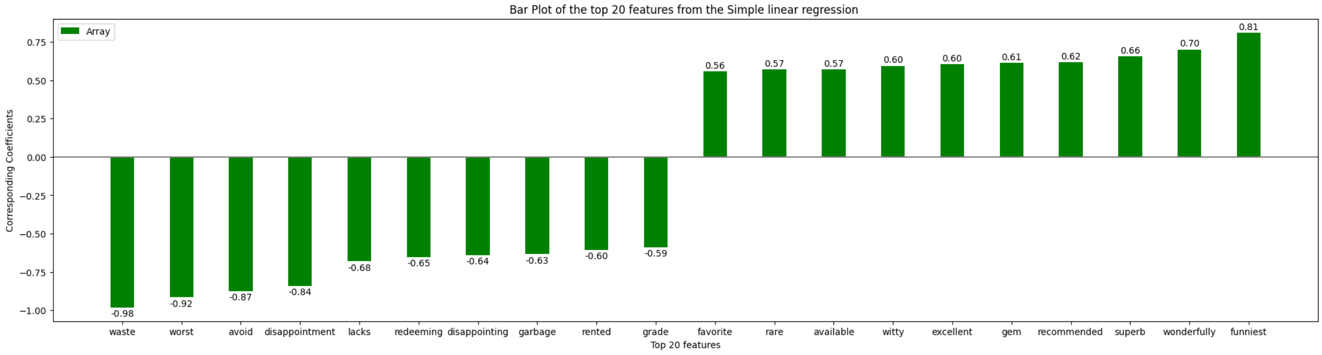


Figure 1: 20 Top features found by Simple Linear Regression

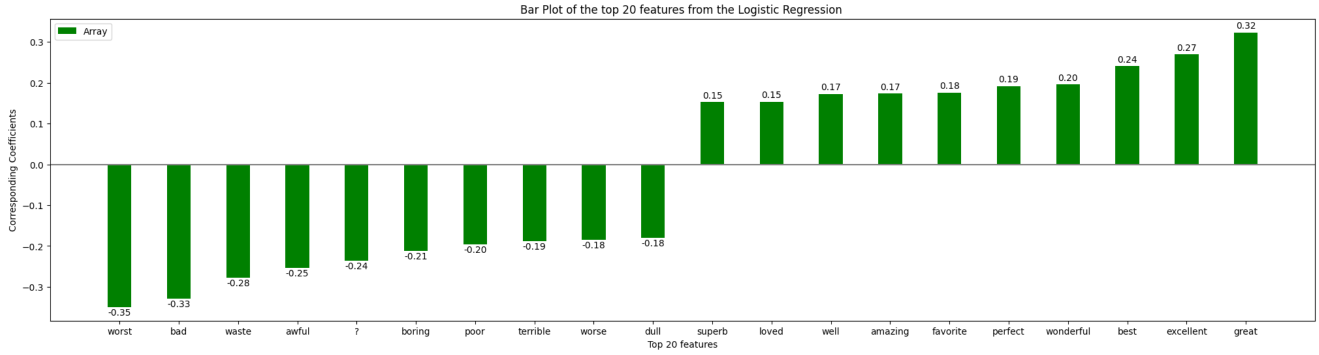
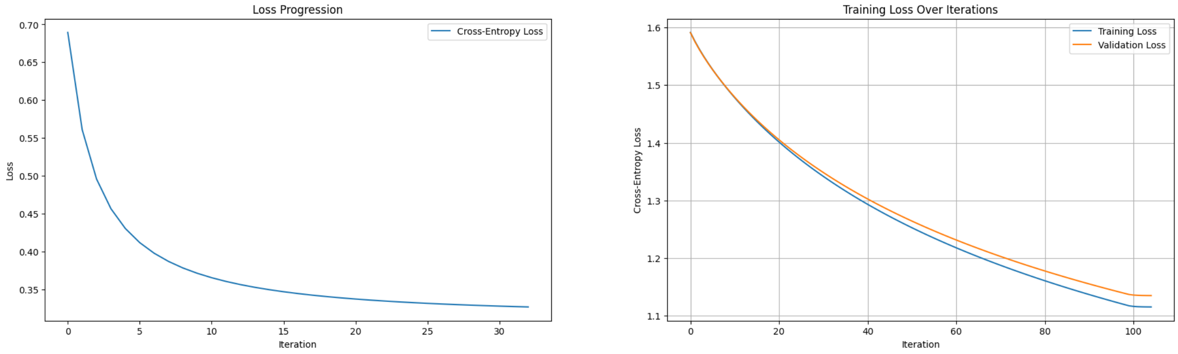


Figure 2: 20 Top features found by Logistic Regression

Then we checked the cross-entropy as a function of iterations in the following graphs for both problems. The monotonically decreasing behavior proves that the model is training to a better set of coefficients, indicating that the learning rate is proper and it converges when the training loss is under the threshold. Notice that we early stopped the training for multiclass regression when the loss for validation set increases.



(a) Cross Entropy Trend in Logistic Regression

(b) Cross Entropy Trend in Multiclass Regression

Figure 3: Cross Entropy Trend

For model evaluation, we plotted the ROC curves of logistic regression and sklearn-DT (Decision Trees) on the IMDB test data in Figure 4, which shows that logistic regression is more stable under all thresholds and DT even doesn't outperform random guess very well. Furthermore, we took a closer look at how different amount of training data affects the model stability, shown inn Figure 5 and 6. In general, the more data we trained on, higher AUROC the model obtained for Logistic Regression, although the difference is not that big; while for DT, it osccilates a bit. The same trend is observed for accuracy of multiclass regression and DT shown in the figure below. In general, the logistic regression and multiclass regression outperforms DT.

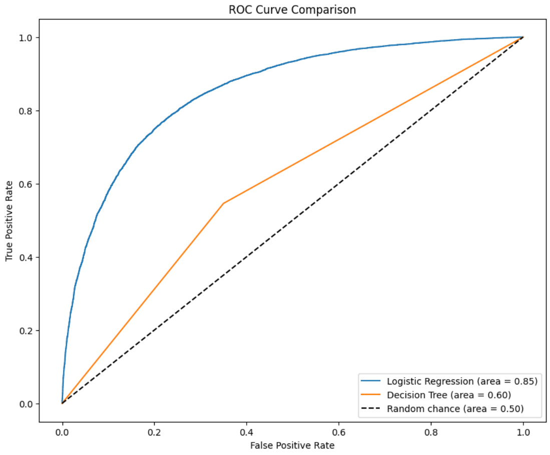
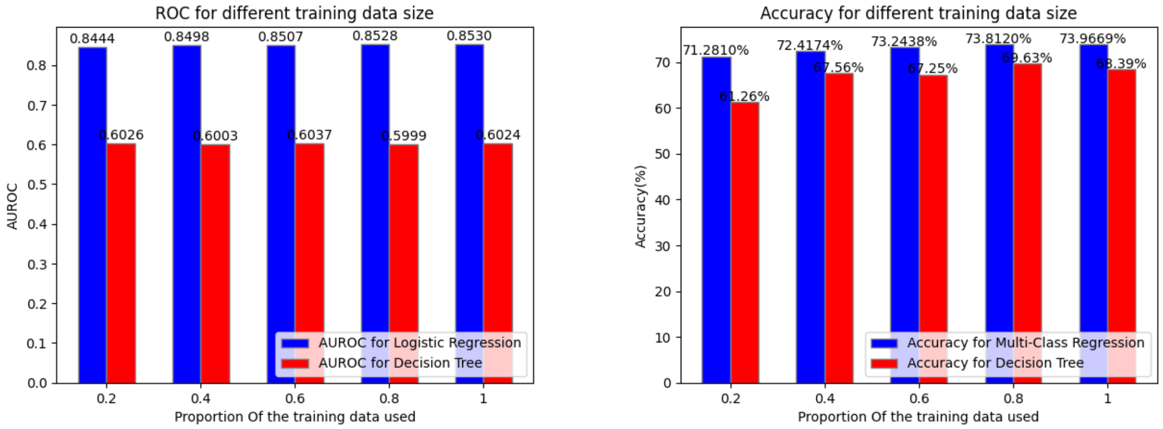


Figure 4: ROC curves of logistic regression and DT



(a) AUROC of logistic regression and DT when part of the training data is used

(b) Accuracy of multiclass regression and DT when part of the training data is used

Figure 5: Model Evaluation for logistic regression and multiclass regression

Finally, we showed a heatmap showing the top 5 most positive features for each of the classes, where the yellow bars stands for the positive features along the diagonal. We can see that there is clear correlation between the important words and corresponding categories. For example, 'weapon'/'law'/'government'/'gun'/'guns' are found for class talk.politic.guns.

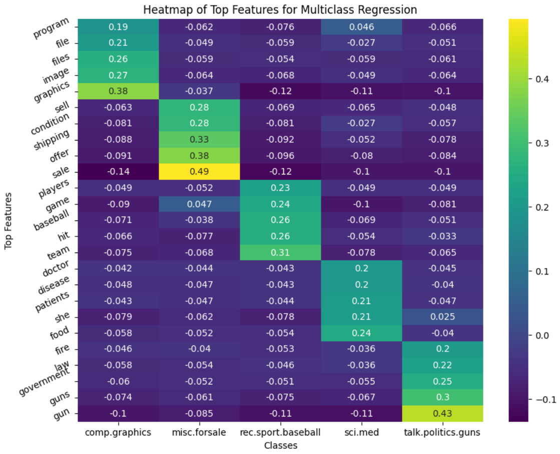
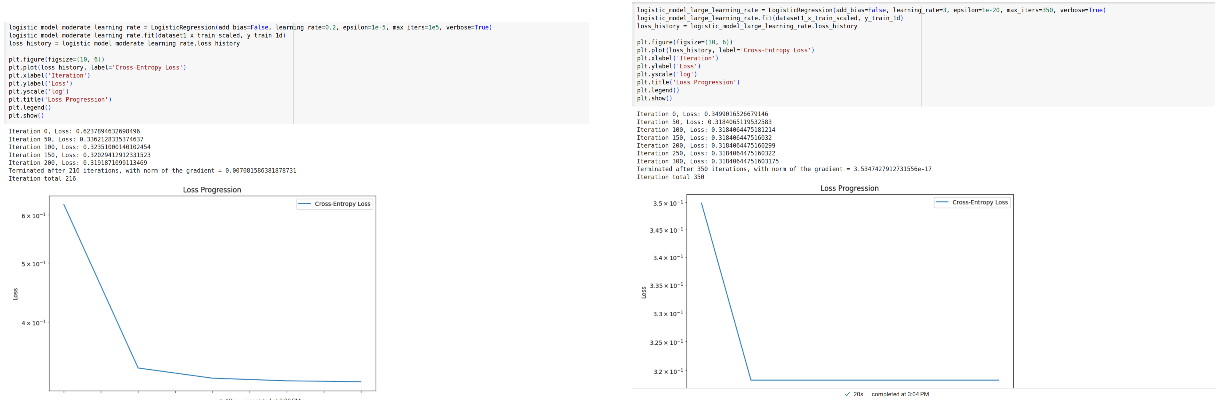


Figure 6: 5 Top features per class found by Multiclass Regression for 5 classes

In addition to the required tests, we also tried different learning rates and tested out the influence on model trained. When using a slight larger learning rate is used (0.2), the model converges faster after 216 iterations while it converges after 1613 iterations in the previous experiments when learning rate is 0.01. However, if the learning rate is too large(3), we can see that the loss after iteration 250 is even larger than the loss after iteration 200, indicating that the coefficients are bouncing around the optimal but cannot get there.



```
logistic_model_moderate_learning_rate = LogisticRegression(add_bias=False, learning_rate=0.2, epsilon=1e-5, max_iters=1e5, verbose=True)
logistic_model_moderate_learning_rate.fit(dataset1_x_train_scaled, y_train_1d)
loss_history = logistic_model_moderate_learning_rate.loss_history

plt.figure(figsize=(10, 6))
plt.plot(loss_history, label='Cross-Entropy Loss')
plt.xlabel('Iteration')
plt.ylabel('Loss')
plt.yscale('log')
plt.title('Loss Progression')
plt.legend()
plt.show()
```

```
Iteration 0, Loss: 0.6237894632698496
Iteration 50, Loss: 0.3362128335374637
Iteration 100, Loss: 0.32351000140102454
Iteration 150, Loss: 0.32029412912331523
Iteration 200, Loss: 0.3191871099113469
Terminated after 216 iterations, with norm of the gradient = 0.007081586381878731
Iteration total 216
```

```
logistic_model_large_learning_rate = LogisticRegression(add_bias=False, learning_rate=3, epsilon=1e-20, max_iters=350, verbose=True)
logistic_model_large_learning_rate.fit(dataset1_x_train_scaled, y_train_1d)
loss_history = logistic_model_large_learning_rate.loss_history

plt.figure(figsize=(10, 6))
plt.plot(loss_history, label='Cross-Entropy Loss')
plt.xlabel('Iteration')
plt.ylabel('Loss')
plt.yscale('log')
plt.title('Loss Progression')
plt.legend()
plt.show()
```

```
Iteration 0, Loss: 0.3499016526679146
Iteration 50, Loss: 0.3184065119532583
Iteration 100, Loss: 0.3184064475181214
Iteration 150, Loss: 0.318406447516032
Iteration 200, Loss: 0.3184064475160299
Iteration 250, Loss: 0.3184064475160322
Iteration 300, Loss: 0.31840644751603175
Terminated after 350 iterations, with norm of the gradient = 3.5347427912731556e-17
Iteration total 350
```

(a) CE and converge time when moderate learning rate is used

(b) CE and converge time when large learning rate is used

Figure 7: Influence of learning rate selection

We also used Ridge Regression for binary classification on IMDB dataset, with the ROC curve comparison with Logistic Regression as below. The accuracy for ridge regression is 0.7734 while the accuracy for logistic regression is 0.7747. We can see that both accuracy and ROC(AUROC) are very similar for the two methods.

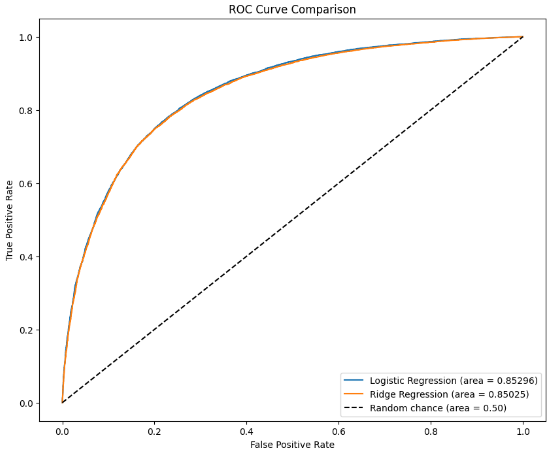


Figure 8: ROC curves of logistic regression and Ridge Regression on IMDB dataset

Lastly, we extend the 5-class classification problem to an 8-class classification problem by extending the training set, with a looser condition that if the true label appears in the top 3 classes predicted, then it's a correct prediction. The accuracy for multiclass regression model in this scenario is even better than previous, reaching 78.8% on the testing data. Then we plotted the heatmap for the 8 classes as follows. It's still great difference betweeen classes, however there are also some words like ”team”, which is the key word for class ”rec.sport.baseball” but still have a high coefficient for class ”misc.forsale”, which appears frequently in both topics.

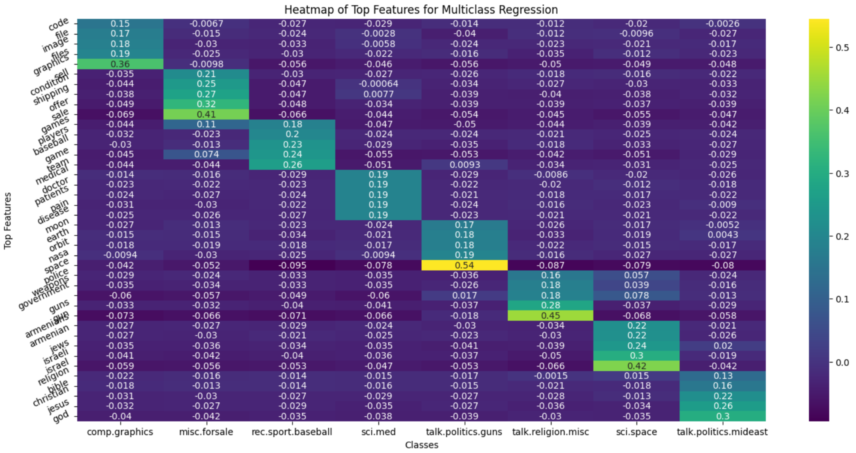


Figure 9: 5 Top features per class found by Multiclass Regression for 8 classes

4 Conclusion and discussion

This study highlights how logistic and multiclass regression models outperform traditional decision trees in text classification tasks, using IMDB Reviews and 20 Newsgroups datasets. Our experiments show that careful feature selection significantly improves model accuracy and stability, with top features from the IMDB dataset logically reflecting sentiment, which validates our approach. The success of these models suggests a promising path for future text classification research.

The key takeaways from this assignment reveal the importance of feature selection in enhancing model performance, demonstrating that both logistic and multiclass regression models can effectively use refined feature sets for improved classification outcomes. The adaptability of these models, evidenced by their consistent performance across different datasets and classification tasks, suggests a promising avenue for future exploration in text classification and sentiment analysis.

Future research could explore different methods for selecting features to better understand text, or include more types of sentiment for detailed analysis. Experimenting with newer deep learning techniques could also improve the accuracy and insights from large text datasets.

5 Statement of Contributions

This section outlines the contributions of each team member to the project.

Terrance Yan

- Task 1 and corresponding textual explanation
- The dataset part of the report

Yunjia Zheng

- Task 3 and reporting behaviors in Result section
- Communication between team members

Annie Kang

- Task 2
- Abstract, Intro and conclusion

References

[1] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, "A survey on text classification algorithms: From text to predictions," *Information*, vol. 13, no. 2, p. 83, 2022. DOI: `10.3390/info13020083`. [Online]. Available: `https://doi.org/10.3390/info13020083`.

[2] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 142–150. [Online]. Available: `http://www.aclweb.org/anthology/P11-1015`.

[3] M. Naeem, F. Rustam, A. Mehmood, Mui-Zzud-Din, I. Ashraf, and G. Choi, “Classification of movie reviews using term frequency-inverse document frequency and optimized machine learning algorithms,” *PeerJ Comput Sci*, vol. 8, e914, Mar. 2022. DOI: `10.7717/peerj-cs.914`. [Online]. Available: `https://doi.org/10.7717/peerj-cs.914`.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] K. B. P. N. D., “Efficient text classification of 20 newsgroup dataset using classification algorithm,” *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, no. 6, pp. 1236–, 2017. DOI: `10.17762/ijritcc.v5i6.934`. [Online]. Available: `https://doi.org/10.17762/ijritcc.v5i6.934`.

[6] J. Kaur and P. K. Buttar, “A systematic review on stopword removal algorithms,” *International Journal on Future Revolution in Computer Science & Communication Engineering*, vol. 4, no. 4, pp. 207–210, 2018.

[7] H. Li, *Machine Learning Methods*, trans. by L. Lu and H. Zeng. Springer Singapore, 2023. DOI: `10.1007/978-981-99-3917-6`. [Online]. Available: `https://doi.org/10.1007/978-981-99-3917-6`.