Mini Project 2 - COMP 551 W2022

Sung Jun Lee, Joseph Boehm, Namdar Kabolinejad
March 7, 2022

1 Abstract

Sentiment analysis and classification are major parts of Machine Learning that rely on mathematically accurate computational models. In this project, we attempt to build and test several multiclass classification models, including Gaussian Naive Bayes and Multinomial Naive Bayes, from scratch to perform sentiment analysis and group classification on two datasets and compare their performance. Moreover, we also use the built-in Scikit-learn logistic regression model for comparison. We train our models on two different datasets, and also experiment with NLP-based feature extraction techniques to augment our original datasets. Additionally, we implement, from scratch, K-Fold cross validation to test and fine tune our models on different hyperparameters. Lastly, we also aim to experiment the effects of varying training set sizes on our models' test accuracy, by testing the accuracy of our models using different dataset sizes then use data visualisation techniques to better understand the performance of our models. From our experiments, we have found that the Logistic Regression model outperforms our Multinomial Naive Bayes model when the numbers of features are low, and that the Multinomial Naive Bayes model outperforms the Logistic Regression model when the number of features are high. Additionally, we also confirmed that the models show better performance as the size of the dataset grows.

2 Introduction

In this project we had a few major tasks to complete. We implement two Naive Bayes model from scratch - Gaussian and Multinomial - experiment with and use different text feature extraction methods, train our models on two different datasets for multi-class classification, fine tune the models using K-fold cross validation, compare the accuracy of our models with the accuracy of the Scikit-learn logistic regression model on our data sets, and observe the effects of varying training set sizes on testing accuracy of models. However, the main goal of this project has been to compare the performance of different models on different datasets of varying dimensionality.

In 2018, Indian computer scientists wanted to compare different machine learning models and how well they work on different sentiment datasets. They also wanted to create an ensemble of the models such that the combination of them would outperform all the other models. On the Sentiment140 dataset, their Naive Bayes model had an accuracy of 75.19% and their logistic regression model had an accuracy of 74.15%. Their ensembled model had an accuracy of 75.81% which was higher than any of the individual models [1]. In 2020, a paper was released that discusses categorizing text documents using multiple machine learning models. For naive bayes, the research paper states that it had an accuracy of 92% on the 20 newsgroups dataset and the scientists model for logistic regression had an accuracy of 96% on the same dataset [2]. Their accuracies are very high due to the advanced NLP-based preprocessing that they did for their feature generation. Their preprocessing included stop word removal, lemmatization and part of speech tagging. They also computed the TFIDF for each token and they also used an n-gram approach to pair words in a sequence together to get more information. Therefore, our goal is to train our models such that it is able to achieve accuracies similar to the previous works' highest scores while using less feature engineering, and for the Sentiment140 dataset, less training instances. We expected that the Scikit learn logistic regression model would give a lot better accuracy compared to our naive bayes models, but our multinomial naive bayes model outperformed the logistic regression model on the 20 newsgroup dataset. The multinomial naive bayes model had test accuracies of 52.60% and 68.25% on the 20 newsgroups and sentiment140 datasets respectively, whereas the logistic regression model had test accuracies of 51.16% and 69.92% on the 20 newsgroups and sentiment140 datasets respectively. Surprisingly, both the logistic regression model and the naive bayes model gave very similar accuracies on the test sets.

3 Datasets

The 20 newsgroups text dataset consists of about 18000 instances and has 20 topics. Scikit learn allows us to easily split up the data into test and train sets. Each instance is the text of a message that was posted to its newsgroup, where the newsgroup is the target value. Meanwhile, the Sentiment140 dataset has 1600000 instances but only 2 topics, either positive or negative sentiment. Each instance in this dataset is a tweet and its corresponding y-value is its sentiment. We then began to preprocess the dataset, where we first set all the text of each instance to lowercase and we removed any and all symbols. In an attempt to creatively explore different feature extraction techniques, we then created two training datasets from each dataset. For the first one, which we will refer to as the 'Regular' set, we tokenized each instance and converted each instance into a bag of words. Whereas for the second dataset, which we will refer to as the 'TSLS' (Tokenized, Stopword, Lemmatized, Stemmed) set, we tokenized each instance and then removed stopwords from the instances. After this, we performed lemmatization and stemming on each instance and finally we turned the instances into bags of words. The stemming and lemmatization reduce each word to its root word. For the second dataset, we additionally removed any and all stopwords using the nltk 'english' stopwords set. To generate our bag of words count vectors, we used sklearn's CountVectorizer.

We quickly realised we were running into memory errors as the preprocessed bag of words datasets for each datasets were massive, so we only kept words with greater than 100 total count for the newsgroups datasets and greater than 500 total count for the Sentiment140 dataset. We also took a random 5% of the dataset from the Sentiment140 dataset, leaving us with 80000 instances for the training sets, which we believed would still be plenty.

For the newsgroups dataset, the stemming and lemmatization preprocessing reduced the total number of features from 1578 to 1451. Whereas for the sentiment140 dataset, the heavy preprocessing brought the total number of features from 266 down to 188. From Figure 1, we observe that the distribution for the samples are approximately equal among all classes.

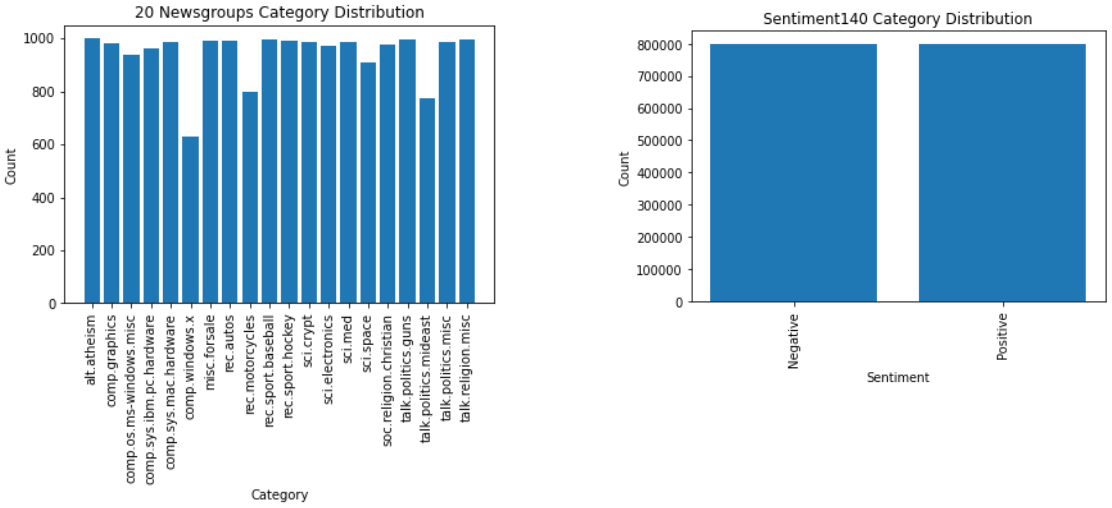

Figure 1: Distribution of categories for each dataset

4 Results

For our experimentation we tested using our custom Gaussian Naive Bayes, Multinomial Naive Bayes, and Sklearn's Logistic Regression. We conducted two separate experiments. First, we experimented with various hyperparameters for each of the 3 models on each of the four datasets using our custom 5-fold Cross Validation algorithm to determine the best performing hyperparameters, and models for each dataset. For the hyperparameters, we tried alphas of [1, 5, 10] for MultinomialNB, smoothings of [0.005, 0.05, 0.5] for GaussianNB, and max_iters of [50,75,100] and solvers of ['saga', 'newton-cg', 'lbfgs'] for Logistic Regression. The second test we conducted was to see the effects of different training set sizes on the test set accuracy of our models. We tested the effects using Multinomial Naive Bayes, and Logistic Regression on the two Regular datasets.

4.1 20 News Group

As seen in Table 1, the highest test classification accuracy achieved was 52.60% with the MultinomialNB model with alpha=1 and the TSLS dataset. In the Regular dataset, the highest testing accuracy was 48.33% with Logistic Regression and max_iter=75. The 5-fold CV was able to accurately predict the highest performing model on the test set for the TSLS dataset, but not for the Regular dataset. Additionally, we noticed that MultinomialNB on average produced the highest accuracy model with Logistic Regression in second place and GaussianNB quite a bit behind. Additionally, we were not able to collect reliable data for the 'saga' and 'newton-cg' solvers as they either often failed to converge or would be extremely slow.

In our second test with varying training set sizes, we spotted a clear trend in the test accuracies of both the MNB and LR models. As shown in Figure 2, as training set sizes increased, the models became more accurate. The hyperparameters used here for MNB and LR were from the highest performing models from the first experiment. Interestingly we see that MNB outperformed LR on smaller datasets but LR eventually performed marginally better than MNB, which agrees with our data from Table 1.

		MultinomialNB			GaussianNB			Logistic Regression (lbfgs)		
		alpha=1	alpha=5	alpha=10	smoothing =0.005	smoothing =0.05	smoothing =0.5	max_iter= 50	max_iter= 75	max_iter= 100
Regular Dataset	5-Fold CV	52.51%	52.05%	49.62%	32.91%	31.92%	13.07%	52.73%	54.72%	**55.28%**
	Test Set	47.66%	47.21%	45.36%	28.63%	28.63%	12.23%	46.92%	**48.33%**	47.13%
TSLS Dataset	5-Fold CV	**57.81%**	57.23%	56.69%	35.46%	34.78%	13.09%	57.35%	56.94%	56.73%
	Test Set	**52.60%**	52.38%	52.10%	31.01%	30.72%	12.15%	51.16%	50.96%	50.74%

Table 1: Validation and Test Set Accuracies of the Three Models on the Two 20 News Group Datasets, highest in **bold.**

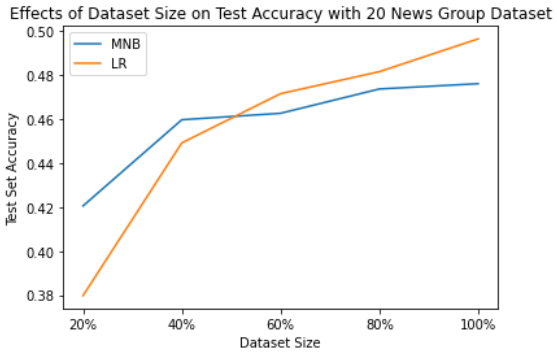

Figure 2: Test Set Accuracy of MNB and LR Models on 'Regular' 20 News Group Dataset

4.2 Sentiment140

		MultinomialNB			GaussianNB			Logistic Regression (lbfgs)		
		alpha=1	alpha=5	alpha=10	smoothing =0.005	smoothing =0.05	smoothing =0.5	max_iter= 50	max_iter= 75	max_iter= 100
Regular Dataset	5-Fold CV	71.03%	71.01%	70.99%	68.52%	68.10%	50.67%	71.82%	71.82%	**71.83%**
	Test Set	**67.97%**	67.97%	67.69%	66.57%	64.62%	51.53%	67.41%	67.41%	67.41%
TSLS Dataset	5-Fold CV	69.50%	69.50%	69.49%	68.53%	68.69%	53.78%	69.95%	69.95%	**69.95%**
	Test Set	67.97%	67.97%	68.25%	68.80%	66.57%	51.81%	69.92%	69.92%	**69.92%**

Table 2: Validation and Test Set Accuracies of the Three Models on the Two Sentiment140 Datasets, highest in **bold.**

As seen in Table 2, the highest test classification accuracy achieved was 69.92% with the Logistic Regression model with max_iter=50,75,100 and the TSLS dataset. In the Regular dataset, the highest testing accuracy was 67.97% with MultinomialNB and alpha=1. Again, the 5-fold CV was able to accurately predict the highest performing model on the test set for the TSLS dataset, but not for the Regular dataset. Additionally, we noticed that the Logistic Regression model on average produced the highest accuracy model with MultinomialNB slightly behind in second place and GaussianNB in third place. Additionally, 'lbfgs' again proved to be the best solver, as 'saga' often would not converge, and 'newton-cg' would often take too long to finish leading to incomprehensible results.

In our second test with varying training set sizes, we spotted a weaker trend than Figure 2 in the test accuracies of both the MNB and LR models. As shown in Figure 3, as training set sizes increased, the models loosely became more accurate. The hyperparameters used here for MNB and LR were from the highest performing models from the first experiment. Interestingly we see that LR outperformed MNB on smaller datasets but MNB eventually performed slightly better than LR, which does not agree with our data from Table 2. However, seeing as the difference in accuracies is <1%, this could simply be due to the random sampling of data.

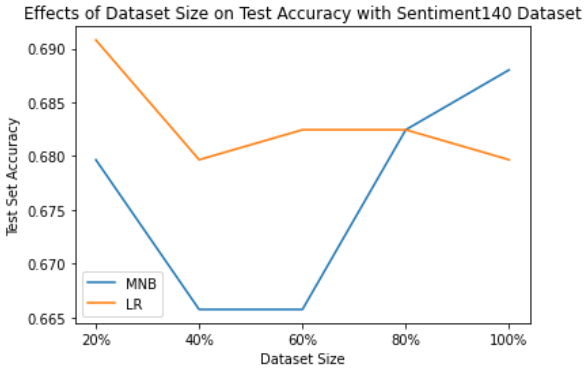

Figure 3: Test Set Accuracy of MNB and LR Models on 'Regular' Sentiment140 Dataset

5 Discussion & Conclusion

The results of our experiments were mostly as expected. For both datasets, we achieved much higher than random accuracies across all of our thereby proving the effectiveness of our models and feature engineering. For the 20 News Group Dataset (NGD), we noticed that Logistic Regression outperformed the Naive Bayes models in the Regular dataset, while it was the other way around for the TSLS dataset. Thus we can draw our first conclusion that MultinomialNB performs well on datasets with a small number of features and a small number of rows, whereas Logistic Regression performs well on datasets with a large number of features and a small number of rows. Secondly we see, for both the 20 News Group and Sentiment140 (S140) datasets, that the GaussianNB model performs the worst every time. This is not surprising as GaussianNB is typically to be used with feature vectors consisting of continuous values and MultinomialNB is typically to be used with feature vectors composed of counts. In our investigations, we deal with discrete counts, and thus Multinomial is expected to perform better. Additionally for the NGD, we see that smaller values of alpha, smoothing, and large values of max_iter and 'lbfgs' were the most effective hyperparameters for MultinomialNB, GaussianNB, and Logistic Regression respectively. This is most likely because large values of alpha and smoothing can overinfluence the model and thus cause it to deviate from the optimal solution. The fact that larger max_iter led to higher accuracies could indicate that the model could be further improved with greater values of max_iter. Perhaps in future investigations, it may be helpful to experiment with higher values of max_iter. And surprisingly, the 'newton-cg' and 'saga' solvers were not effective. They often did not converge at all or would take a very long time to compute, thus rendering them useless for our purposes. For NGD, we also noticed the effectiveness of our creative feature engineering approach based on NLP techniques. We consistently found that the TSLS dataset yielded higher accuracies than the Regular dataset. Since the TSLS dataset removes a lot of "useless" words and eliminates word variations, it is able to model more important words, thus resulting in higher accuracies. We should also note that the difference in validation accuracies and testing accuracies are not vastly different, thereby confirming that our models were not overfitting. Lastly for the NGD, we confirmed that having more training samples was beneficial for both the MultinomialNB and Logistic Regression models. We saw significant increases (6-12%) in

accuracy from using 20% to 100% of the NGD. As for the S140 dataset, we noticed that MultinomialNB outperformed the Logistic Regression model in the Regular dataset, while it was the other way around for the TSLS dataset, although only by a small margin. One key difference here is that the differences in accuracies are very small. This led us to believe that model MultinomialNB and Logistic Regression are almost equally effective models for large datasets. Again, similar to NGD, we see that smaller alpha and smoothing are the best performing hyperparameters for our models. However, this time, we see that the differences in accuracies per differing hyperparameters are not huge. This is most likely due to the large sample size of the S140 dataset. This time, we observed that the 'lbfgs' was able to converge rather quickly leading to the same accuracies for each max_iter. And again, 'newton-cg' and 'saga' were not consistent. Perhaps in future investigations, other solvers may be worth testing. The TSLS feature engineering approach was also effective here, thereby strengthening the validity of our approach. Lastly for S140, we found a rather interesting result when it came to testing different dataset sizes. Unlike NGD, where significant improvement was seen, the S140 models didn't show a huge accuracy jump for varying dataset sizes, although a weak upwards trend was observed. This may be because the S140 dataset is much larger than the NGD to begin with, and thus even a small fraction is still more than enough to train the models. To combat this issue in future experiments, it may be more effective to test different concrete numbers of rows rather than a percentage of the dataset. Furthermore, for S140, we did not see a significant increase (2%) in accuracies across different sizes. Seeing as though the trade-off between accuracy and dataset size is not expensive, it may be worth using a smaller dataset for the S140 dataset if computational time and resources are an issue.

6 Statement of Contributions

- Sung Jun Lee
    - Wrote Gaussian & Multinomial Naive Bayes, Cross-Validation, Logistic Regression, Testing, Pre-preprocessing code
    - Contributed to Preprocessing code
    - Contributed to Abstract, Introduction, Datasets
    - Wrote Results and Conclusions

- Joseph Boehm
    - Wrote data preprocessing code
    - Wrote Datasets section
    - Contributed to Introduction
    - Wrote References

- Namdar Kabolinejad
    - Contributed to Naive Bayes code
    - Contributed to Preprocessing and Data Prep code
    - Wrote Abstract
    - Contributed to Introduction

References

[1] Ankit, & Saleena, N. (2018). An Ensemble Classification system for Twitter sentiment analysis. *Procedia Computer Science*, *132*, 937–946. https://doi.org/10.1016/j.procs.2018.05.109

[2] Kumar, S., Gulati, A., Jain, R., Nagrath, P., & Sharma, N. (2020). Categorising text documents using naïve Bayes, SVM and logistic regression. *Data Management, Analytics and Innovation*, 225–235. https://doi.org/10.1007/978-981-15-5619-7_14