Comparative Study of Logistic Regression and Naive Bayes on Text Classification

Hannah Reed, Nadia Hawarri, Malena Mendilaharzu

Abstract

Classification algorithms are used for many applications such as pattern recognition, text classification, filtering of spam emails. In this paper, we will provide a comparison of the performance of two different models on text classification. We will compare discriminative and generative classification by evaluating the accuracy of logistic regression and multinomial Naive Bayes on two different datasets (20 newsgroup and IMDb reviews). Furthermore, we will evaluate the differences in the accuracy of both models when varying the training size. After performing various experiments, we discovered that with optimal hyperparameters, logistic regression outperformed Naive Bayes when the training size is larger. However, better results were achieved by the multinomial Naive Bayes classifier on a smaller training size. As expected, we found that a decrease in training size caused a decrease in the accuracy of both models. Training time of logistic regression was found to be significantly faster than that of Multinomial Naive Bayes.

Introduction

In this paper, we attempt to provide a detailed comparison of the performance of Multinomial Naive Bayes (MNB) and Logistic Regression (LR) on two different datasets: IMDb reviews and 20 News Group. Naive Bayes is a generative classifier meaning that it "learns a model of joint probability p(x,y) and makes predictions using Bayes rules to calculate p(y|x)" [2]. Logistic regression is a discriminative classifier that models the posterior p(y|x) directly (without assuming independence). Research has shown that "if the training data is scarce, one can expect Naive Bayes to outperform Logistic Regression, but as the number of training examples grows, Logistic Regression will outperform Naive Bayes" [1]. Therefore, since the training size of IMDb is larger than the one for the 20 News Group, we hypothesize that Logistic Regression will perform better in the IMDb dataset, while Multinomial Naive Bayes will perform better in the 20 News Group dataset. After running our experiments, we found that our hypothesis was supported.

In addition to comparing the performance of LR and MNB, we also attempted to study the effect of the training size on the accuracy of our models. It was found that the accuracy of both models decreased as training size became smaller. This is consistent with previous research which had found that "the size of the training set and the classification rate are indeed positively correlated" [3].

Furthermore, we discovered that multinomial naive Bayes had a slower training time than Logistic Regression.

Datasets

Both datasets were cleaned and preprocessed before being used for the experiments. They were then split between "Train" and "Test" and subsets containing 20, 40, 60, and 80% of the "Train" data were created. CountVectorizer was used as the primary natural language processing tool in order to analyze the frequency of words in each document.

The 20 newsgroup dataset was imported from sklearn.datasets. We then used the default ‘train’ and ‘test’ subsets and removed the headers, footers, and quotes as required. Then, we used train_test_split from sklearn.model_selection to create the subsets of the training counts.

The IMDb reviews dataset is split evenly with 25k reviews intended for training and 25k reviews intended for testing the classifier. Each set contains 12.5k positive and 12.5k negative reviews. We first downloaded the tar file containing all 50k reviews, unpacked it and merged it into two separate files: one for testing and one for training. Each file contained all 25k reviews intended for training/testing the classifier, with the positive reviews appended to the beginning of the file and the negative reviews appended to the end of the file. We then cleaned the data by removing all symbols that were not words. We decided to remove features like "[.;:!\'?,\"()\[\]]" in order to avoid taking them into consideration at the time of classifying a document. We didn’t want the frequency of such symbols to influence the classification of our document. Finally, we created the feature vectors using countVectorizer.

We then analyzed the class distributions of each of the datasets.

As shown in Figures 1.1 and 1.2, the IMDb reviews contain an equal number of Class 0 (negative reviews) and Class 1 (positive reviews). However, the 20 News Group does not have equal representation for each of the classes which might lead to a faulty classification.

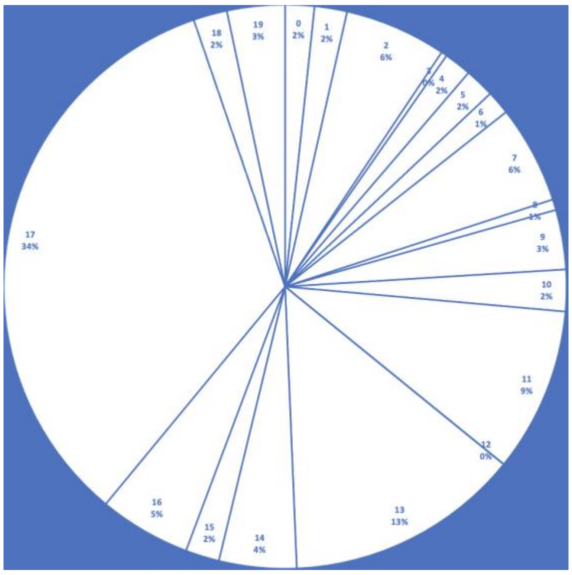

Figure 1.1: Class distribution for 20 newsgroup

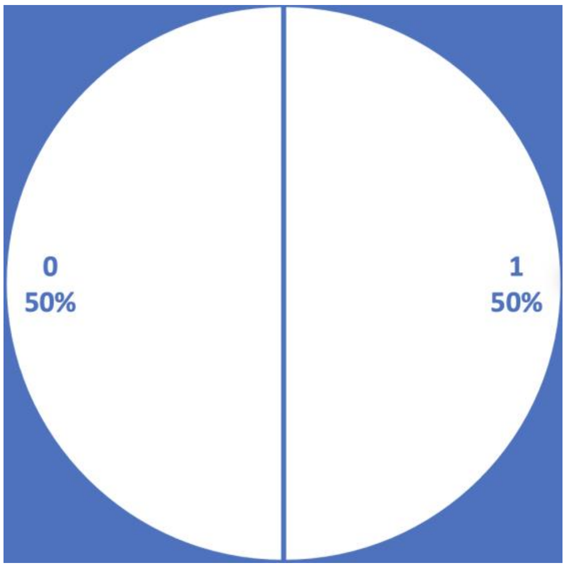

Figure 1.2: Class distribution for IMDb reviews

Results

Hyperparameter tuning:

Using the scikit library for hyperparameter tuning, we ran 5-fold cross-validation on Logistic Regression and Multinomial Naive Bayes for both datasets in order to find the optimal C regularization value and alpha smoothing value, respectively.

Optimal values for Logistic Regression:

	Rank 1	Rank 2	Rank 3
Mean Validation Score	0.667	0.662	0.662
Standard Deviation	0.012	0.014	0.013
Parameters	0.25	0.1	0.5

Figure 2.1: Tuning of C for logistic regression on 20 news group

Logistic Regression for IMDb movie reviews:

	Rank 1	Rank 2	Rank 3
Mean Validation Score	0.867	0.866	0.863
Standard Deviation	0.006	0.004	0.007
Parameters	0.1	0.01	0.25

Figure 2.2: Tuning of C for logistic regression on IMDb

As shown in Figures 2.1 and 2.2, the optimal C values for Logistic Regression are 0.25 and 0.1 for 20 News Group and IMDb reviews respectively.

Optimal values for Multinomial Naive Bayes:

	Rank 1	Rank 2	Rank 3
Mean Validation Score	0.708	0.708	0.707
Standard Deviation	0.006	0.007	0.008
Parameters	0.011	0.005	0.039

Figure 2,3: Tuning of alpha for MNB on 20 news group

	Rank 1	Rank 2	Rank 3
Mean Validation Score	0.807	0.779	0.749
Standard Deviation	0.009	0.01	0.749
Parameters	1	0.1	0.01

Figure 2,4: Tuning of alpha for MNB on IMDb

As shown in Figures 2.3 and 2.4, the optimal alpha values are 0.011 for the 20 News Group and 1 for the IMDb reviews.

Performance achieved with optimal parameters:

We then used these optimal hyperparameters to evaluate the performance of the classifiers on different training sizes (20%, 40%, 60%, 80% and 100%). As expected, we found that increasing the training size increased the performance of the classifier. This result was consistent across both models and datasets. We also observed that the change in performance was higher for the 20 News Group dataset.

Performance of the MNB vs training size:

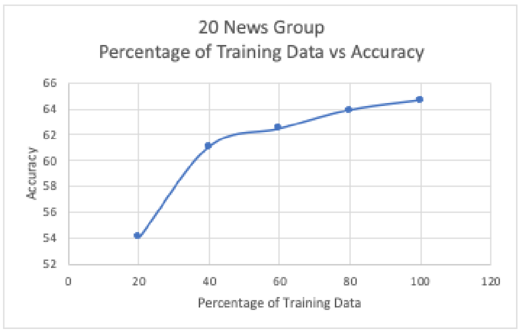

Figure 3.1: Accuracy of MNB vs training size for 20 news group

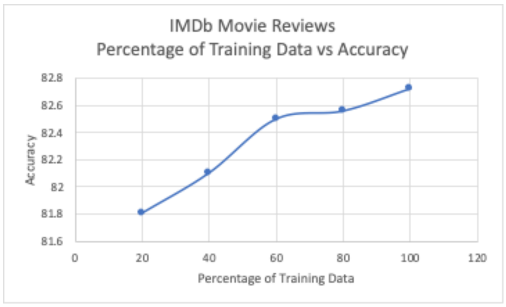

Figure 3.2: Accuracy of MNB vs training size for IMDb reviews

Performance of the LR vs training size:

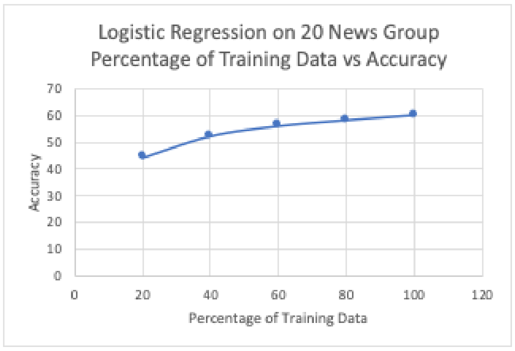

Figure 3.3:: Accuracy of LR vs training size for 20 news group

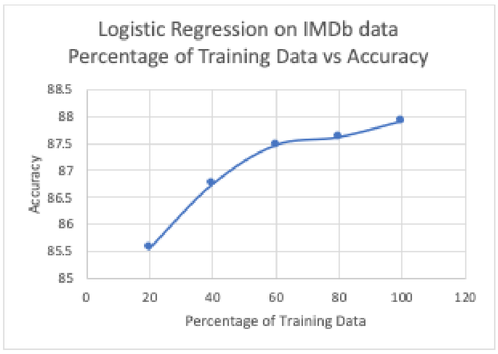

Figure 3.4: Accuracy of LR vs training size for IMDb

Performance of LR vs MNB:

After running 5-fold cross-validation and finding the optimal parameters for both MNB and LR, we used these parameters to compare the performance between the two models. We found that MNB performed better in the 20 News Group dataset by around 4 percent. While LR performed better in the IMDb dataset by around 5 percent. The differences in the results between the two datasets are due to one key difference between MNB and LR. MNB estimates a joint probability p(x,y) = p(y) * p(x|y) from the training data (assuming conditional independence between features). LR, on the other hand, estimates p(y|x) directly from the training data (and without assuming conditional independence). Thus, MNB tends to perform better than LR when you have a smaller training size, while LR performs better than MNB as the training size increases. The reasoning behind this is that LR tends to overfit when there is not enough data to estimate p(y|x). However, when the training set is large, MNB might “double count” features that are correlated with each other, because it assumes that each event is independent. This could negatively affect the accuracy of the Multinomial Naive Bayes . In the 20 NewsGroup dataset, we have 11314 data points in the training set, while in the IMDb movie reviews we have 25,000 data points in the training set. Therefore, we can conclude that LR performs better than MNB in the IMDb dataset due to an increase in training data size, while MNB performs better in the 20 NewsGroup dataset due to a smaller training data size.

Figure 4.1: Comparison of MNB and LR on 100% of the training data

	20 News Group	IMDb Movie Reviews
Multinomial Naïve Bayes	64.62%	82.72%
Logistic Regression	60.26%	87.92%

Figure 5 shows the classification reports and confusion matrices for both datasets to help visualize the performance of each model.

MNB on the 20 News Group

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.48	0.54	0.51	319
1	0.55	0.70	0.62	389
2	0.50	0.01	0.02	394
3	0.52	0.71	0.60	392
4	0.58	0.70	0.63	385
5	0.76	0.71	0.74	395
6	0.85	0.70	0.77	390
7	0.69	0.72	0.70	396
8	0.68	0.74	0.70	398
9	0.89	0.79	0.84	397
10	0.60	0.84	0.70	399
11	0.74	0.73	0.73	396
12	0.66	0.56	0.61	393
13	0.80	0.76	0.78	396
14	0.76	0.70	0.73	394
15	0.60	0.80	0.69	398
16	0.59	0.62	0.61	364
17	0.77	0.71	0.74	376
18	0.43	0.41	0.42	310
19	0.32	0.26	0.29	251
ACCURACY			0.65	7532
MACRO AVG	0.64	0.64	0.62	7532
WEIGHTED AVG	0.65	0.65	0.63	7532

Figure 5.1: Classification report for MNB on the 20 newsgroups, all training data and alpha = 0.011

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	172	0	0	2	0	2	0	5	3	2	9	3	2	3	7	47	7	9	15	31
1	4	271	1	15	26	21	2	1	5	1	5	15	7	5	5	3	0	0	2	0
2	6	76	3	132	50	53	3	2	5	0	15	9	7	8	10	3	0	0	8	4
3	0	9	0	280	53	3	8	3	0	1	7	4	22	1	0	0	0	0	1	0
4	1	14	1	29	269	1	8	7	6	1	14	5	18	0	6	1	1	0	1	2
5	0	57	1	10	11	282	3	1	5	0	5	6	3	7	2	0	1	0	0	1
6	0	2	0	29	25	0	273	14	10	3	10	1	6	3	8	1	2	1	2	0
7	1	2	0	2	1	0	8	284	38	1	24	2	9	1	5	4	3	2	7	2
8	6	2	0	1	2	1	4	31	293	2	13	1	9	5	2	4	9	4	5	4
9	8	3	0	0	1	1	3	3	9	314	22	1	1	4	3	9	5	0	9	1
10	7	2	0	0	0	0	1	4	5	20	335	2	0	1	2	5	2	3	8	2
11	2	5	0	3	6	1	0	1	6	3	16	288	8	4	8	5	15	8	15	2
12	1	18	0	29	21	0	3	22	10	1	11	31	222	8	10	1	1	2	1	1
13	6	5	0	3	1	0	2	9	8	0	14	1	6	300	5	13	5	5	9	4
14	10	13	0	1	0	2	0	10	7	0	18	3	7	7	277	6	2	5	22	4
15	26	5	0	1	1	0	0	1	0	1	14	1	0	2	0	318	1	0	2	25
16	11	0	0	0	0	1	1	5	5	0	11	6	2	5	5	13	227	14	30	28
17	24	1	0	1	0	2	0	2	4	2	6	6	1	0	0	22	7	267	21	10
18	22	2	0	0	0	0	1	6	8	0	7	2	1	7	6	8	80	16	127	17
19	50	4	0	1	0	0	0	2	7	1	7	3	4	5	4	63	16	11	8	65

Figure 5.2: Confusion Matrix for MNB on the 20 newsgroups, using all training data and alpha = 0.011

MNB on the IMDb Dataset

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.79	0.89	0.84	12500
1	0.87	0.77	0.82	12500
ACCURACY			0.83	25000
WEIGHTED AVG	0.83	0.83	0.83	25000

Figure 5.3: Classification report for MNB on the IMDb reviews

	0	1
0	11095	1405
1	2915	9585

Figure 5.4: Confusion Matrix for MNB on the IMDb reviews using all training data and alpha = 1.0

Performance of Linear Regression on IMDb Data

We used the Linear Regression model from sklearn to predict ratings with the IMDb dataset. We found that linear regression performed significantly worse for this task than Logistic Regression and MNB. We trained the linear regression model on 20% of IMDb training set, 40% ,60%, 80%, and 100%. The best score was obtained on the test data using 20% of the training data, with a score of 0.12. Performance got worse as we used more of the training data, with scores dipping into the negatives. This could be due to non-linearity of the data, and could be improved by using a nonlinear base function.

Discussion and Conclusion

In conclusion, we found that the MNB performs better than LR for small training sizes, while LR performs better for large training sizes. Further, we found that the overall accuracy of both models on the IMDb dataset was higher than on the 20 News Group. This could be caused by the difference on the learning rate, the training size or the class distribution for each dataset. As previously mentioned, the 20 News Group had more inconsistencies and many classes were underrepresented on the training set. In particular, as we show in Figure 1.2, the data points in Class 3 can be approximated to represent a zero percent of the whole training set. It is evident that this affects the classification of such points. As shown in the confusion matrix in Figure 5.2 , Class 2 and 3 have a tendency of being confused, and hence misclassified. A possible solution to this could be adding more data points of Class 3 to the training set. We could also consider using the TfidfVectorizer instead of the CountVectorizer as our natural language processing tool. With CountVectorizer, we only count the number of times a word appears in a document. This results in biasing in favour of the most frequent words. Therefore, we end up ignoring rare words that could have helped in classifying the data more efficiently. On the other hand, TfidfVectorizer considers the overall document weightage and allows for the recognition of infrequent and frequent words. One can then choose to penalize frequent words as they don’t provide much to differentiate between classes and give more importance to infrequent words.

Contributions

Equal amounts of work were provided by the three members throughout the whole assignment. Malena and Nadia focussed on the cleaning and preprocessing of the data and on the code for the multinomial naive bayes classifier. Hannah focussed on the code for k-fold cross validation as well as running experiments for linear regression. The three of us discussed the results and worked on the report.

References

[1] Halloran, J. (2009). Classification: Naive Bayes vs Logistic Regression.

[2] Andrew Y. Ng, Michael I. Jordan (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes

[3] Sordo, M., & Zeng, Q. (2005). On sample size and classification accuracy: A performance comparison. In *Biological and medical data analysis* (pp. 193–201). Springer.