Text Classification and Sentiment Analysis: Naive Bayes vs. Logistic Regression Classifier

Abstract

This report mainly focused on exploring and comparing the performance of Naive Bayes (Multinomial & Gaussian) and Logistic Regression Classifier in text classification of 20 distinct categories of news and binary sentiment analysis of IMDB movie reviews. Additionally, we also investigated the performance of a SVM classifier: Linear Support Vector Classifier (LinearSVC), which uses a linear kernel. Interestingly, Logistic Regression and Linear SVM classifiers obtained similar accuracy of approximately 85% in both binary and tf-idf weighted data, and are consistently better than Naive Bayes models. Besides, both Multinomial and Gaussian Naive Bayes take significantly more time to train and make predictions. Plus, we examined how the strength of regularization and size of the training set will impact two linear models' performance.

1. Introduction

The core task of text classification and sentiment analysis is to extract the different characteristics of the subjective content generated by humans, and utilize it to predict the main topics of the text and the attitudes of authors, which is extremely important for spam and hate speech filtering. This work attempts to analyze and compare 2 feature extraction techniques for textual data (binary occurrence and tf-idf weighting), and the performance of 4 classification algorithms: Multinomial & Gaussian Naive Bayes Logistic Regression, and Linear SVM.

1.1 Background Information

Naive Bayes is a generative model learning the joint probability distribution of the data based on Bayes' theorem, with the naive assumption of conditional independence.

Multinomial Naive Bayes is a variation of Naive Bayes model with multinomial features, which is adapted to deal with data with features of discrete values, like word counts, etc.

Logistic Regression is a variation of Linear Regression model, and both of them are discriminative; they learn the conditional probability distribution directly instead of the joint distribution like generative models.

Linear SVM is a discriminative model defining an explicit hyperplane decision boundary between classes with maximum margin and equidistant.

Binary Occurrence vectorizes a document by binary code to indicate whether a certain word in the bag of words occurs in a document.

TF-IDF is a weighting algorithm for textual data, which uses the term frequency of a word multiplying the inverse of its document frequency. This approach effectively extracts featured words in a particular document among all other documents.

1.2 Task & Data Description

In this work, we investigated the performance on news topic classification and sentiment analysis of 4 classifiers, 2 of which are generative (Gaussian & Multinomial), and the other 2 are discriminative (Linear SVM & Logistic Regression). We used fetch_20news dataset in sklearn for evaluating models' performance on text classification and IMDB movie reviews dataset for testing their abilities to distinguish positive and negative reviews.

The task can in general be divided into three parts. First, we preprocessed the data, removing headers, footers and quotes in the 20 news dataset, eliminating stop words and meaningless symbols in the content, and converting words in each document into vectors. To vectorize the textual data used in this report, "bag of words" technique was implemented, and we compared the effect of two encoding methods: binary occurrence and tf-idf weighting. Second, we implemented models needed in the report: Naive Bayes, and k-fold cross validation from scratch, but logistic regression and linear SVM classifiers used here are the standard model in sklearn package. In addition, we 5-cross validated and analyzed the performance of these four models on multi classification on the 20 new group dataset and binary sentiment classification on the IMDB Reviews. Plus, we explored how the size of training data affects the accuracy of these models and their sensitivity to training data size changes. Ultimately, we examined the impact of regularization strength on the two linear models and developed extra

smoothing techniques for Guassian Naive Bayes to avoid catastrophic failure in edge cases.

1.3 Important Findings

We found that, compared to binary occurrence, tf-idf weighting improved the performance of three models out of four, except for Guassian Naive Bayes. And Linear SVM reached the highest accuracy (82.79%) in 20 news dataset while Logistic Regression slightly outperforms Linear SVM and got an accuracy of 87.7%, and both of the linear model were significantly faster to train and made predictions than Naive Bayes Classifiers did. Besides, in general, accuracy of models increases as the size of the training set increases, a function that concaves down. Lastly, we detected that there's a certain level of randomness inherent in logistic regression due to the random initialization of the gradient descent algorithm, and it may greatly influence how regularization strength will affect the accuracy of the model.

1.4 Related Work

Previously, Poyu Li compared the performance of four models: Random Forest, Gaussian & Multinomial & Bernoulli Naive Bayes, and Linear SVM. In conclusion, Linear SVM using tf-idf weighted data and bigram/unigram vectorization generates the optimal result: approximately 89.22% accuracy. His work inspired us to utilize tf-idf weighting methods and bigram/unigram vectorization. Despite limited computational resources given by the machine of our group, this work still guided us to improve the performance of our model.

2. Data and Set up

In this section, we provide the context of our datasets, then describe general preprocessing methods that are common to all our methods.

2.1 Data Source

The 20 news dataset is from sklearn.datasets package; IMDB movie reviews dataset is from the repository of Stanford University AI Lab.

2.2 Pre-processing

The preprocessing phase consists of lemmaziting & vectorizing all words in each document, removing all stop words like “a, an, etc.”, and encoding them as binary or tf-idf weighted variables. Specifically, the headers, footers, and quotes in the 20 news dataset were intentionally removed for simplicity. Also, for IMDB reviews data, the original thousands of text files were formatted into two separate csv files as training and testing data, accelerating loading data.

2.3 Information

In the 20 news dataset, there are 11,314 training and 7,532 testing data points of 20 categories of news that are relatively evenly distributed. The following pie chart illustrates the details.

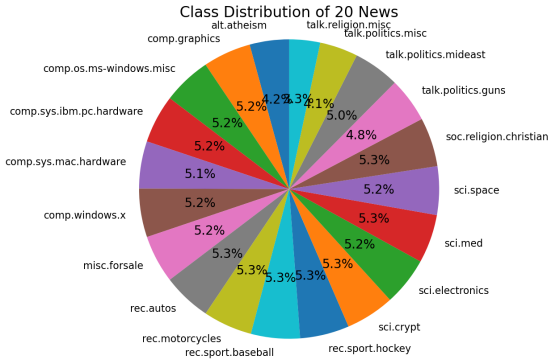


Figure 1: The distribution of 20 kinds of news

In the IMDB reviews dataset, 25,000 training and 25000 testing points were perfectly balanced and labeled as positive and negative. After vectorization, data in both sets are in more than 100,000 dimensional space. Therefore, our models may be subject to the curse of dimensionality.

3. Results

3.1 Naive Bayes vs. Logistic Regression & SVM

Via 5-fold cross validation, hyperparameters, encoding method, and cost functions of all models have been optimized. To summarize, in both datasets, tf-idf weighting gives the best results: for 20 news, Naive Bayes reached about 67-77% accuracy, but logistic regression and linear SVM got 80+% accuracy.

	MNB	GNB	LR	LSVM
Best Parameter	α=1	N/A	C=20	C=1.0
Cross Validation Score	**76.85 ±0.30%**	**65.00 ±0.52%**	**87.93 ±0.34%**	**88.36 ±0.32%**
Test Score	77.36%	67.46%	82.14%	**82.79%**

Table 1: Summary of Performance on 20 news

M/GNB: Multinomial/Gaussian Naive Bayes

LR: Logistic Regression; LSVM: Linear SVM

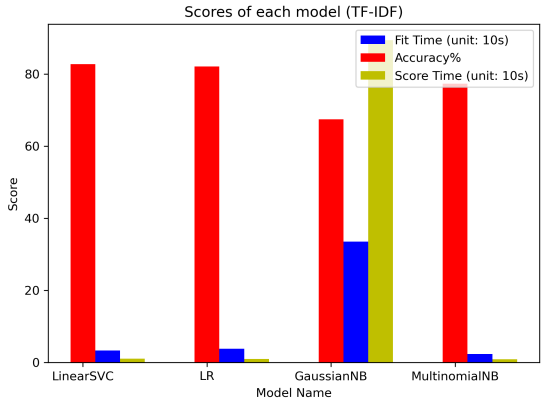


Figure 2: Performance of four models on tf-idf weighted 20 news, including fit time, accuracy, and score time.

From the figure above, we could find that Gaussian Naive Bayes took a lot of time to train and predict, while the other three spent similar time for news classification tasks. Linear SVM is the winner in this dataset.

For the IMDB movie review dataset, Logistic Regression gives the best result; the distribution of fit & score time is also similar to that of 20 news.

	MNB	GNB	LR	LSVM
Best Parameter	α=1	N/A	C=5	C=1.0
Cross Validation Score	**78.21 ±0.57%**	**55.03 ±0.44%**	**89.16 ±0.51%**	**88.94 ±0.67%**
Test Score	77.08%	56.98%	**87.70%**	86.66%

Table 2: Summary of Performance on IMDB

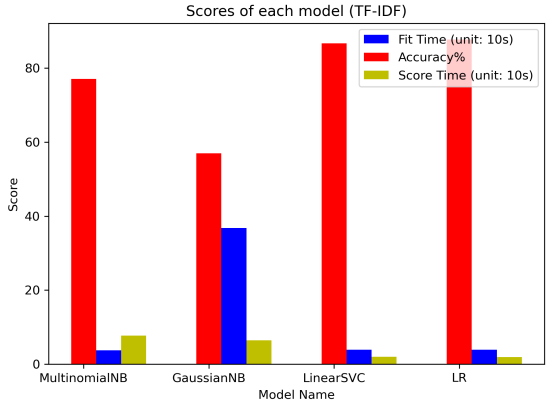


Figure 3: Performance of four models on tf-idf weighted IMDB reviews, including fit & score time, and accuracy

3.2 Effect of Training Set Size on Accuracy

Mostly, increasing training set size results in monotonously increasing accuracy as a concave-down function, but in a few cases, some model (e.g. Multinomial Bayes) is not very sensitive to the number of training data.

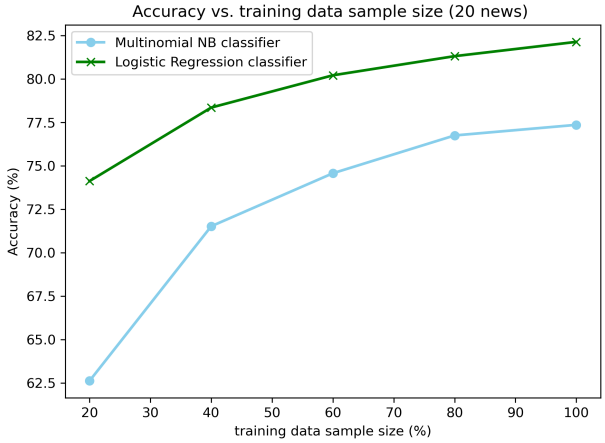


Figure 4: Effect of training data size on accuracy in the tf-idf weighted 20 news dataset.

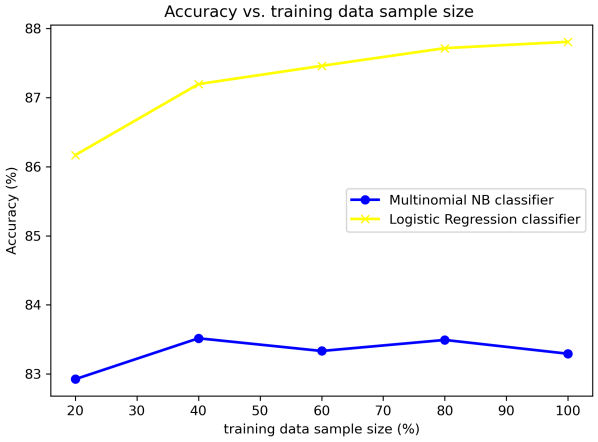


Figure 5: Effect of training data size on accuracy in the tf-idf weighted IMDB review dataset.

Since more training data offered more insights into the pattern of such data in the real world, increasing training data seems to always improve the performance of models. However, We can also see that when the complexity of classification(i.e. number of classes available for classification) is low (e.g. binary), it's easy to maintain the distribution of data, so Naive Bayes can perform similarly in small dataset as well as large dataset if training data are carefully chosen (according to Stanford, these reviews are highly polar and representative). However, when the number of classes to classify is high, 20 classes in this case, a relatively large size of training data is pivotal for reflecting the true distribution, i.e. $p(y)$, of the data, therefore allowing Naive Bayes to perform better as more training data are available.

3.3 Additional Experiments

Besides the experiments above, we also conducted additional experiments to further our analysis and understanding with these models.

3.3.1 Binary Occurrence vs. TF-IDF

Compared to binary occurrence, which only indicates whether a word occurs in the document or not, tf-idf weights the characteristic words in the document more than other words, which could extract certain featured words from a document and suppress noises. If we use *t* for word, *d* for document, and *D* for the document set, the definition of tf-idf score is:

$$tf\ idf\ (t, d, D) = tf\ (t, d)\ .\ idf\ (t, D)$$

where,

$$tf\ (t, d) = log\ (1 + freq\ (t, d))$$

$$idf\ (t, D) = log\ (\ \frac{N}{count\ (d \in D : t \in d)}\ )$$

The result of tf-idf weighted data has been presented in the section 3.1, so here we only show the result of binary encoded data.

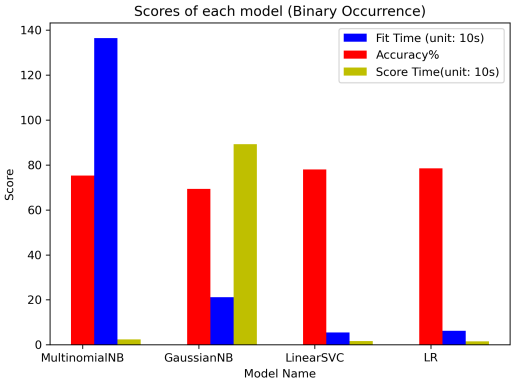


Figure 6: Performance of four models on binary encoded 20 news, including fit time, accuracy, and score time.

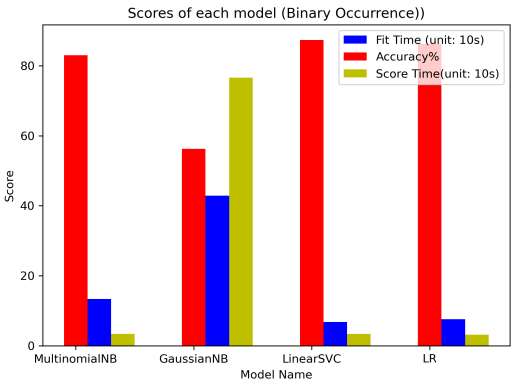


Figure 7: Performance of four models on binary encoded IMDB data, including fit time, accuracy, and score time.

To summarize, compared to that of tf-idf weighted data, the accuracy of binary encoded data is consistently lower in all models by 2-5%, but Multinomial Naive Bayes much more time to fit binary encoded 20 news data.

3.3.2 The Effects of Regularization Strength

Regularization is a very useful technique to avoid overfitting, and the strength of regularization can greatly affect models' accuracy.

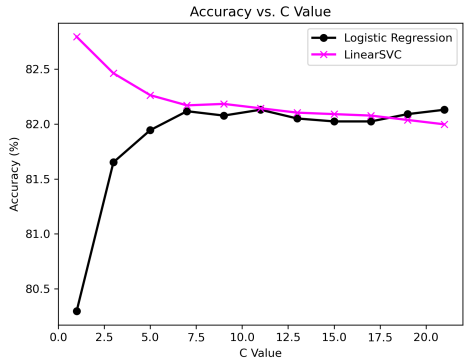
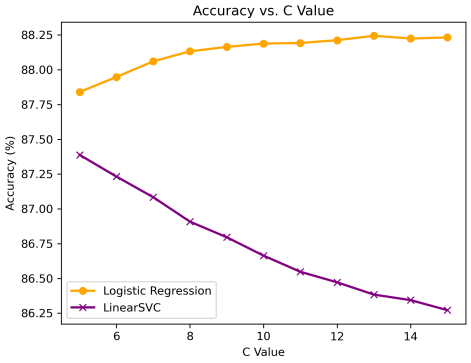


Figure 8: Inverse of regularization strength vs. accuracy on tf-idf weighted 20 news dataset



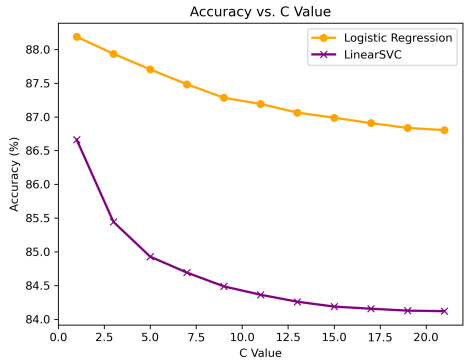


Figure 9: two different plot of inverse of regularization vs. accuracy on tf-idf weighted IMDB review data

In the figures above, C value stands for the inverse of regularization strength, meaning larger C value gives weaker regularization. Stronger regularization (smaller C value) could be good or bad, which depends on the model. In particular, logistic regression has some randomness in nature since gradient descent algorithms will randomly initialize the start point, which could lead to different local minimum points. In high dimensional space like textual data, randomness can lead to very different results despite the scope of regularization is the same. Therefore, when tuning regularization strength in a logistic regression model, we should try multiple start points to ensure we find a stable and excellent minimum point for our model.

3.3.3 Variance Smoothing Technique

In the Gaussian Naive Bayes model, the algorithm needs to compute the log-likelihood, which involves the average and standard deviation of the data. However, in some cases, standard deviation of a word could be zero if this word occurs in every document, which will cause catastrophic failure of the Gaussian Naive Bayes model. Accuray can be hugely damaged if this happens. Therefore, we developed a smoothing technique for variance in the Gaussian Naive Bayes model we implemented. We forced the variance for each word to be at least 1e-9, which is too small to affect general cases, but this will greatly help the performance in edge cases mentioned above.

4. Discussion & Conclusion

4.1 Takeaways

First, it's extremely essential to make careful choices of data preprocessing procedures when working with textual data. Vectorizing words using binary occurrence and tf-idf makes a difference, and tf-idf outperforms over the binary encoding method. Also, removing stop words, lemmatization & stemming can also reduce the noises in the data, but it's still possible to make the data overly clean and reduce the performance of models, which implies stop words and conjugation may play an implicit role in structuring the patterns of texts. Naively eliminating features that we subjectively consider as futile might be penalized by poor models' performance.

Besides, logistic regression and linear SVM, two discriminative models, are better in dealing with text classification and sentiment analysis than generative models, like Naive Bayes, do. No matter in terms of fit/score time or accuracy, two discriminative models showed huge advantages over Naive Bayes.

In addition, regularization can sometimes rescue the model from overfitting, but an overly regularized model will be underfitting. In the logistic regression model, the random starting point of gradient descent method may change the minimum point the model finally converges to, so the optimal regularization strength may also change according. Therefore, when tuning the regularization strength of models using gradient descent or other randomized approaches, it would be better to have a large range of regularization strength at first and run the algorithm multiple times, to obtain a less turbulent and well-performed minimum point.

Lastly, increasing the size of training dataset generally helps the model to get a better performance, both for discriminative and generative classifiers. When there're many classes to classify, a large training data set is essential to maintain the distribution, which greatly impacts the performance of Naive Bayes. But blindly increasing the number of training data will not benefit Naive Bayes models much if the classification complexity is low. For Naive Bayes, a small but carefully chosen dataset can generate results as good as ones using a much larger dataset. When the data is difficult to collect, Naive Bayes is not a bad choice.

4.2 Future Work

In this project, we tried the performance of Linear SVM on text classification & sentiment analysis, but it doesn't always outperform logistic regression. One possible reason is the kernel function of Linear SVM linear, which means the decision boundary generated is a set of straight lines, unlike logistic regression. Therefore, in the future work, we may try whether changing different kernel functions for SVM will improve its performance in natural language processing.

Statement of Contribution

Pingsheng Li: Data preprocessing & experiment design

Jacob Wang: Implementation and optimization of Naive Bayes models

Xiyuan Feng: Collect & analyze experimental data, compilation of the report

Reference

Hsu B-M. Comparison of Supervised Classification Models on Textual Data. *Mathematics*. 2020; 8(5):851. https://doi.org/10.3390/math8050851

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).*