Classification of Textual Data - Comparison and Analysis of Naive Bayes and Logistic Regression

Yuhe Fan, Xinyu Wang, Yicheng Huang

March 8, 2022

1 Abstract

In this project, we implemented multinomial Naive Bayes, Gaussian naive Bayes and performed cross-validation to compare and determine the best parameters for Naive Bayes and logistic regression. We conducted classification with these two models and found the performance (accuracy) of naive Bayes is worse than that of logistic regression and the speed of naive Bayes is slower than that of logic regression. Moreover, we compared the performance (accuracy) of each model performed on different sizes of training set. In 20 news data set, the performance increases as the training set size increases while in sentiment 140 data set, the performance is stable across different training set size. For this phenomenon, we discussed and gave reasonable interpretation which would be explained later in this report.

2 Introduction

Two datasets were prepossessed firstly to reduce the feature number and running time. We implemented multinomial naive Bayes and with Laplace smoothing and imported logistic regression. Then we tuned the hyper-parameters using 5-fold cross-validation and trained the two models with different size of 20 news and sentiment 140 (full training set, 20%; 40%; 60% and 80%) and compare performance. For the 20 news dataset, the accuracy for naive Bayes and logistic regression performed on full training set is 30% and 30% respectively, while for sentiment 140 dataset, For the 20 news dataset, the accuracy for naive Bayes and logistic regression are 10 and 20 respectively. For different size of training data, we observed different change in two datasets and discussed.

3 Datasets

Check the visualization results in the appendix.

3.1 20 news

20 news originally has 11314 training data and 7532 testing data that belongs to 20 categories. We observed a nearly uniform class distribution hence no need to worry under-represented or over-represented topics. Each world would be counted as a feature. Without prepossessing there would be 101322 features for classifying. To reduce the number of features, we removed stop words, derivative words, plural form words, same words in different tense, numbers, person names and highly correlated words (detected by calculating the covariance less or equal than 0.01). After the prepossessing the data, there are 2413 features left. We also take another sample with randomly selected features with dimensionality and observe the feature distribution, two observations stands out, first, with our prepossessed feature, the distribution are more neatly bell-shape like, second, convert from occurrence to tf-idf would help adjust the skewed distribution.

3.2 Sentiment 140

Sentiment 140 originally has 160000 observations belongs to 2 categories (positive and negative), by observation, 2 categories are evenly partitioned. The only modification is we convert 4 to 1 for positive class for further convenience. Given that 160000 is too large. Therefore, we selected 50000 positive observations and 50000 negative observations and sliced the sentiment and text. When previewing the data set sentiment 140, we found that there some observations having text length larger than 140. This might be caused by links or @sombody, so we removed non-English words.Then to reduce the number of features, we did the same process as 20 news (remove stop words etc.). The only difference is the threshold for highly correlated word (0.01 for 20 news and 0.001 for sentiment 140). Finally, we got 567 features for sentiment 140. By observing the the feature distribution, it is found the features are more skewed, this may due the small number of features we decided.

4 Results

All parameters are tuned with respect to two datasets and two models through cross-validation firstly. For naive Bayes, the smoothing factor is tuned to be 0.001 for 20 news and sentiment 140.

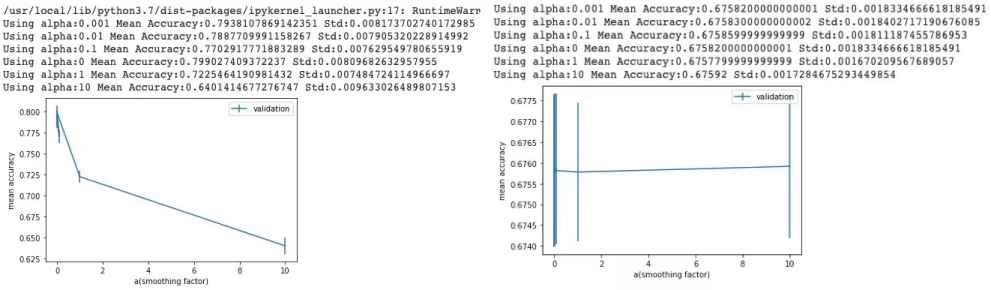

Figure 1: Cross validation of different smoothing factor for naive Bayes (left: 20 news, right: sentiment 140)

For logistic regression the penalty and max iteration number are tuned to be l2 and 1000 respectively.

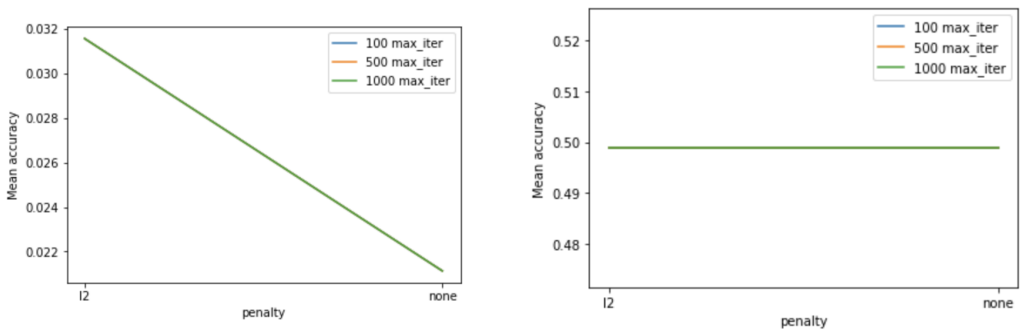

Figure 2: Cross validation of different solver and max iteration number for logistic regression (left: 20 news, right: sentiment 140)

Then we conducted experiments to compare the performance of naive Bayes and logistic regression on each dataset. The performance of naive Bayes and logistic regression on each of the two datasets is shown as following:
In each dataset, logistic regression is slightly better than naive Bayes.

The accuracy of the two models as a function of the size of training set is shown as following:

	multinomial naive Bayes	logistic regression
20 news	0.55112	0.56585
sentiment 140	0.67582	0.69248

Figure 3: Accuracy of multinomial naive Bayes and logistic regression applied on two datasets

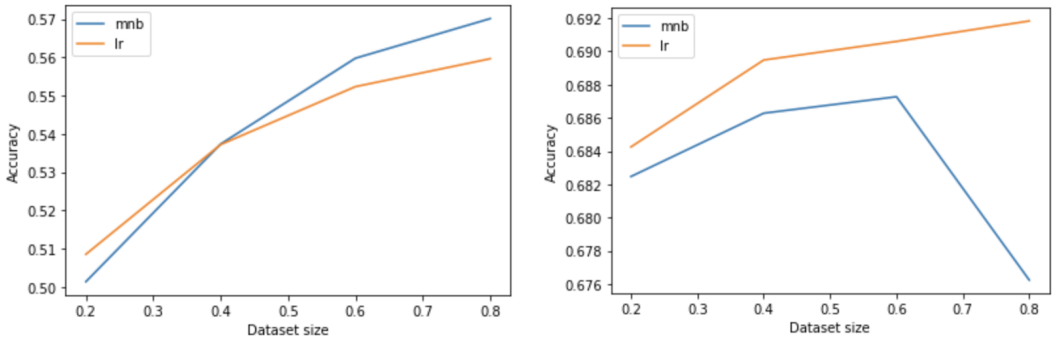

Figure 4: left: news 20, right: sentiment 140

We can see that for 20 news the accuracy increases as the size of training set increases, but for sentiment 140 the accuracy does not change obviously as size of training set changes.

To determine whether our selected features improved the performance of models, we randomly chose same number of features and compared the performance with our selected features.
The result shows that our selected features performed much better than the random feature.

selected features performance

	multinomial naive Bayes	logistic regression	Gaussian naive Bayes
20 news	0.55112	0.56585	0.48407
sentiment 140	0.67582	0.69248	0.67258

random features performance

	multinomial naive Bayes	logistic regression	Gaussian naive Bayes
20 news	0.06386	0.21548	0.18229
sentiment 140	0.49780	0.69248	0.50033

Figure 5: our selected features vs. random features

Furthermore, we compared the performance of multinomial naive Bayes and Gaussian naive Bayes and found their performance in each dataset are similar.

Finally, we compared the running time of naive Bayes and logistic regression referenced to colab running time. For 20 news, the running time for naive Bayes and logistic regression is 33 sec and 20 sec respectively. For sentiment 140, the running time for naive Bayes and logistic regression is 31 sec and 19 sec respectively. Therefore, for these two datasets, logistic regression is faster than naive Bayes.

	multinomial naive Bayes	Gaussian naive Bayes
20 news	0.55112	0.48407
sentiment 140	0.67582	0.67258

Figure 6: Accuracy of multinomial naive Bayes and Gaussian naive Bayes applied on two datasets

5 Discussion

5.1 Laplace smoothing

Laplace smoothing is necessary because even if a word is not present in the training set, its probability should not be 0. It could be observed when doing cross-validation, the 20 news datasets gives division by zero error when alpha is 0, however, it still have high accuracy. These may due to the dense nature of the data in which there's only few error. For sentiment 140, alpha = 0 has high accuracy and no error is reported. Thus sentiment 140 data may coincidentally have no zero probability calculated naturally. In all, we still decide to use 0.001 for both datasets. This is reasonable because 0.001 avoided zero counts and had relatively small impact on the data (our data is in a small scale, so large smoothing factor would infect the data too much).

5.2 multinomial naive Bayes vs. logistic regression

Our results shows that for both datasets, logistic regression and multinomial naive bayes demonstrate similar performance accuracy, with logistic regression only wins more 2% accuracy. Regarding the speed, however, logistic regression is faster than multinomial naive bayes with 1/3 less computation time approximately. Given that naive bayes differ from the logistic regression in which is use an assumption and it works better with small datasets, it is estimated, our assumption about the distribution is quite appropriate, thus, with a medium-sized datasets, two models would achieve similar results.

5.3 multinomial naive Bayes vs. Gaussian naive Bayes

When prepossessing the data, the text is converted to bag of words representation, which contains discrete values. Therefore we chose multinomial naive Bayes as our primary model. To avoid the effect of different text length, we used tf-idf count to avoid the influence of text length. And this leaded that the data range converged to continuous data behaviour, this could be visualized in our feature distrubution, which is almost bell-like. So using Gaussian naive Bayes is also reasonable. Therefore, Gaussian naive Bayes and multinomial naive Bayes have similar performance on 20 news and sentiment 140.

5.4 accuracy vs. data size

This is because after prepossessing in training set, 20 news has 11314 observations and 2413 features, while sentiment 140 has 100000 observations and 567 features. For 20 news dataset, the accuracy increases as data size increases, which is reasonable. However for sentiment 140, the accuracy does not vary much as data size changes. This is because, compared to 20 news, sentiment 140 has a much larger training set and fewer features, which means 20 % of training set might be already enough for training the model.
It was expected that with more data chosen, logistic regression would achieve better accuracy, however, both of the model have relatively same accuracy. It may due to the volume of datasets, are still considered medium for both model.

5.5 selected feature vs. random feature

The performance of our selected feature is much better than the performance of random selected features. This prove that our choice of features is reasonable and improved the model performance.

6 Statement of Contributions

All works are distributed equally with group participation and discussion.

7 Appendix

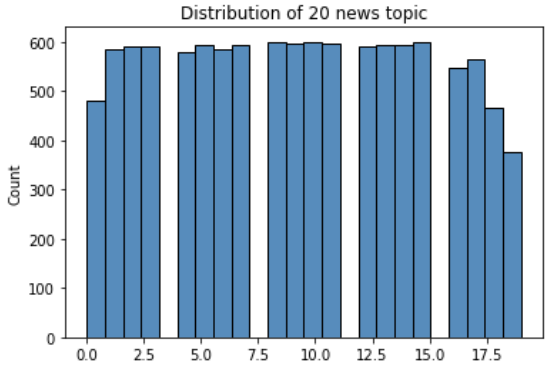

Figure 7: Distribution of 20 Newsgroups Topics

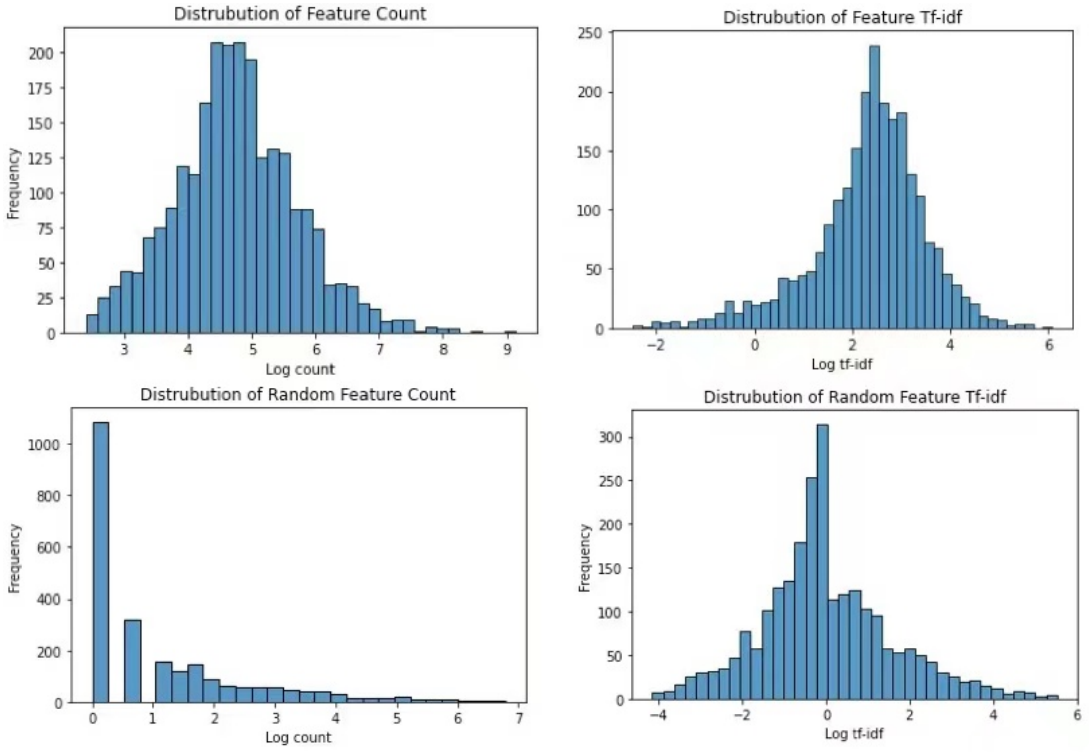

Figure 8: Distribution of 20news Features

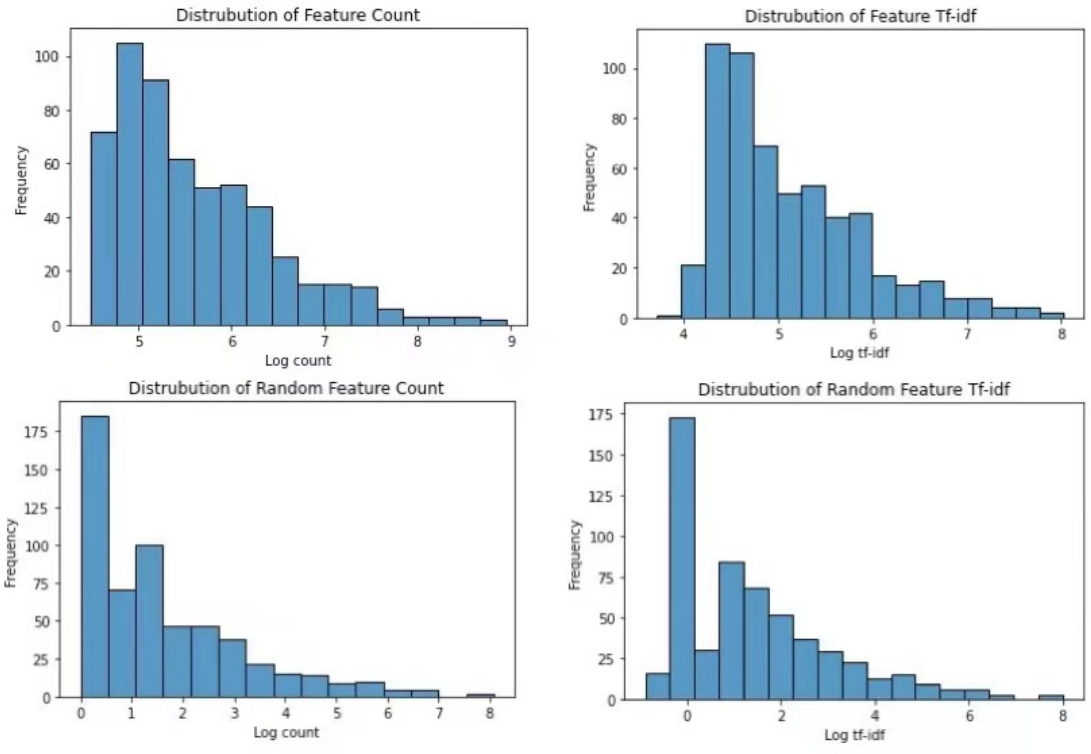

Figure 9: Distribution of 140sentiment Features