COMP 551 Assignment 2

Shelley Xia, Mingze Li, Yiwei Cao

March 7, 2022

Abstract

In this project, we investigated the performance of the logistic regression model and Naive Bayes model across two benchmark datasets. In particular, we implemented three different kinds of Naive Bayes classifiers: the Gaussian Naive Bayes, the multinomial Naive Bayes, and the Bernoulli Naive Bayes. We compared the performances of the four models on the two distinct datasets: the 20 news group dataset from scikit-learn [1] and the Sentiment140 dataset [2]. We found that the multinomial Naive Bayes classifier yielded the best accuracy out of the three Naive Bayes classifiers, and it had a comparable accuracy to that of the logistic regression model.

We started by converting the raw text data to feature vectors. Our exploratory phase included data preprocessing, such as feature selection and testing out various vectorizers provided by the scikit-learn package. We then implemented different Naive Bayes classifiers and performed model selection by cross-validation. Next, we conducted hyperparameter tuning by cross-validation. Overall, we learned that multinomial Naive Bayes was the most pertinent Naive Bayes classifier for these two datasets. Furthermore, a small smoothing parameter, such as alpha = 1, rendered better results than larger ones. For the logistic regression model, we discovered that the default values set by scikit-learn, max interactions = 100 and l2 regularization, gave the best model predictions.

1 Introduction

Machine learning techniques have been widely employed in text classification and sentiment analysis, classification algorithms being some of them [3]. Naive Bayes and the logistic regression models are two very common classifiers and are often compared with each other [4]. In this project, we plan to explore the performances of the logistic regression classifier and three types of Naive Bayes classifiers: the Gaussian Naive Bayes, the multinomial Naive Bayes, and the Bernoulli Naive Bayes.

Specifically, we performed analysis on the following two text datasets: the 20 news group dataset from scikit-learn [1] and the Sentiment140 dataset [2]. The 20 news group dataset contains 18846 instances of text features on 20 topics. The Sentiment140 dataset contains tweet information scraped from Twitter on 3 sentiment groups: positive, neutral, and negative. Note that we were only concerned about a binary classification of positive and negative because the training instances do not include neutral tweets. These datasets were experimented with by previous researchers to investigate different machine learning algorithms [5][6].

To train our models, we first conducted data preprocessing such as feature selection and trying out different vectorizers provided by the scikit-learn package. In particular, we tried CountVectorizer and TfidfVectorizer. Whereas CountVectorizer only produces a sparse representation of simple counts, TfidfVectorizer emphasizes on the relative frequency of words and thus renders more useful information. In order to fit the Bernoulli Naive Bayes model, we also performed one-hot encoding on the results produced by TfidfVectorizer. For feature selection, we tested out different values for parameters of these vectorizers such as 'max_features', 'max_df', and 'max_tf' to limit the number of features while keeping the most important ones.

We then trained all four models using appropriate data. To find the best models to use in our final testing stage, we performed cross-validation for model selection and hyperparameter tunning. Overall, we found that the multinomial Naive Bayes with a small smoothing parameter is the best Naive Bayes classifier in terms of model accuracies. Thus, the multinomial Naive Bayes with alpha=1 was chosen for final tests for both datasets. With regards to the logistic regression model, we examined the effects of different max iterations for convergence and different penalty functions. The result was intuitive: the best hyperparameters were the default ones. In addition, we also investigated the influence of training set on the model performances. We found model accuracies increased

along with the size of training data.

Finally, we compared the performances of the multinomial Naive Bayes and the logistic regression model, both with their best sets of hyperparameters, on the two datasets. Their results were very close. The multinomial Naive Bayes model scores 60.26% on the 20 news group dataset, which is slightly higher than 59.69% of the logistic regression model. Conversely, the logistic regression model scores 81.06% on the Sentiment140 dataset, compared to 79.67% scored by the multinomial Naive Bayes.

2 Datasets

The 20 newsgroups dataset is comprised of over 18000 instances of newgroup posts, divided into 20 categories based on topic. Train and test sets were obtained from the scikit-learn library, with headers, footers, and notes removed. Plotting a pie chart shows that posts are roughly equally divided into the 20 categories in the training set (see graph in code). The sentiment140 dataset was read into pandas dataframes from CSV files. After shuffling the training set, the polarity column was converted into numpy arrays while the text column was converted to lists so that they can be inputed into the vectorizers. Bar plots show that, like the 20 newsgroups dataset, instances in the sentiment140 are roughly equally divided into the classes.

We used CountVectorizer, TfidfVectorizer, and Binarizer (corresponding to multinomial, Gaussian, and Bernoulli NB models, respectively) to extract features from the raw text data and stored these numerical features using the bags of words representation, such that for each document, the number of occurrences, frequency, or the presence or absence of each word is stored in a matrix with indices corresponding to the index of the document and the index assigned to the word. For the CountVectorizer, terms with a document frequency higher than 0.5 were ignored, and only 5000 words were considered when building the vocabulary. Likewise, for the TfidfVectorizer, terms with frequency higher than 0.8 were ignored and the dimension of the features was reduced to 1000 using TruncatedSVD. All vectorizers were given a list of English stop words obtained from github [7]. Note that for the sentiment140 dataset, only 100,000 out of 1,600,000 instances out of the training set were fed into the TfidfVectorizer and the Binarizer due to limitation in RAM. The CountVectorizer was given all 1,600,000 instances.

3 Results

3.1 Hyperparameter Tuning

We first experimented with the Logistic Regression model's hyperparameter tuning while testing the effect of feature selection. Since we were getting convergence warnings while we ran the model, we chose to experiment with the maximum iterations (max_iter) hyperparameter in the hope that more iterations would lead to a better converged model. To our surprise, we found that not only does altering the value of max_iter (100,300,500,700,900) have little impact on our accuracy rate, using dataset 1 without feature-selection gave rise to slightly higher accuracy (figure 1). The same result is also observed from running the same experiment with dataset 2. Following our rather bizarre observation with regard to tuning the maximum iteration hyperparameter, we proceeded to test if having a penalty would lead to better performance. Indeed, the accuracy for the logistic regression model without penalty on dataset 1 is 64.076%, whereas that of the logistic regression model with l2 penalty is 65.447%. However, the result does not follow for our dataset 2. Its model accuracy without penalty is 73.608%, yet the model accuracy with l2 penalty is 73.600%.

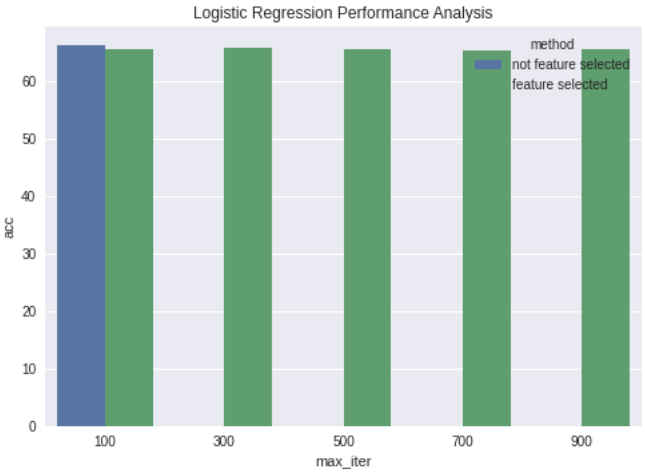

Figure 1

Since we wrote three different Naives Bayes models (Gaussian, multinomial, Bernoulli), we decided to experiment with the Bernoulli Naive Bayes model as well to give our curiosity the full range of due satisfaction. We hyperparameter-tuned the Bernoulli Likelihood's one-hot threshold, the threshold that binarizes words with frequencies past which to 1 and the rest, 0. From the set of values we tested (threshold = 0.00, 0.05,0.1,0.15,0.2,0.25), we found that the performance drops to the lowest when threshold is 0.15, and the performance peaks when the threshold value is 0.25

(figure 2).

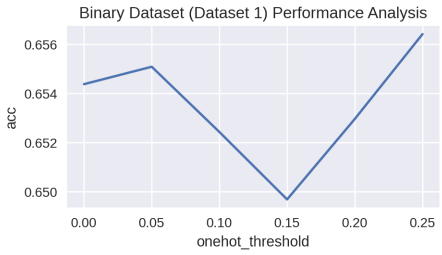

Figure 2

We then tested the effects of having different alpha values(alpha = 0,1,2,3,4,5,10,20,50,100), which is used in Laplace Smoothing on our multinomial Naive Bayes model. The model performance peaks at the alpha = 1, and it gradually decreases as the alpha value increases. Interestingly, the accuracy is at the lowest when alpha is 0, which means there is no Laplace Smoothing taking place (figure 3).

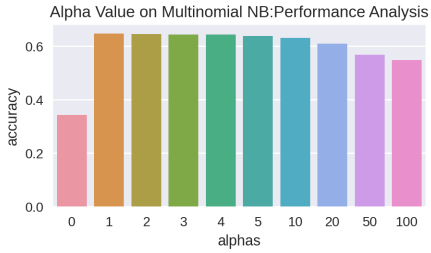

Figure 3

3.2 Naive Bayes Models comparison with cross validation

To gain a deeper understanding of Naive Bayes model performance, we wrote from scratch three different Naive Bayes models: Gaussian, multinomial, and Bernoulli. Using k-fold cross validation (k=5), we tested their performances on both of the datasets, each pre-processed to fit the type of input data required by different Naive Bayes likelihood functions. For the first dataset, the cross validation accuracy is 49.699% for Gaussian NB, 64.695% for multinomial NB, and 65.172% for Bernoulli NB. For Dataset 2, the pattern is slightly different. The cross validation accuracy is 50.192% for Gaussian NB, 72.617% for multinomial NB, and, to our surprise, 23.743% for Bernoulli NB (Table 1).

3.3 Test Set Accuracy Computation

After obtaining the best hyperparameter configurations and the best Naive Bayes model among the three we wrote, we tested the accuracy of the multinomial Naive Bayes model (the overall most consistent, best performing model) as functions of dataset size with both of our datasets. We first fixed our test set, and maintained a single variable by only altering the percentage of the training set we use for training (and discard the rest of unused training set). We tested with a training size of 20%, 40%, 60%, 80%, and 100% of our original training set. We found out that the first dataset's performance gradually increases as the percentage of training set used increases, and the performance peaks at 61% when the training set size is the largest (100% of training set used for training) (figure 4). The second dataset performs rather consistently despite the size of the training set used, maintaining a performance at 80%.

	Dataset 1	Dataset 2
Gaussian	49.699%	50.192%
multinomial	64.695%	**72.617%**
Bernoulli	**65.172%**	23.743%

Table 1

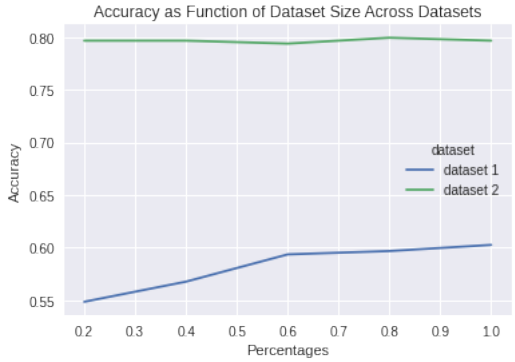

Figure 4

We also tested the model accuracy as a function of dataset size between the Logistic Regression model and the multinomial Naive Bayes model using dataset 1. We found out that both of the models' accuracy increases as the size of training set increases, and the multinomial Naive Bayes model maintains a consistently higher accuracy rate than the Logistic Regression model (figure 5) Next, we performed the same experiment on dataset 2. We discovered that the accuracy of the two models peaks at different training set sizes. For the multinomial Naive Bayes model, its accuracy is at its highest when 60% of the training set is used for training and gradually declines as the size of the training set increases. The accuracy of the Logistic Regression Model, however, monotonically increases as the size of the training set increases

(figure 6).

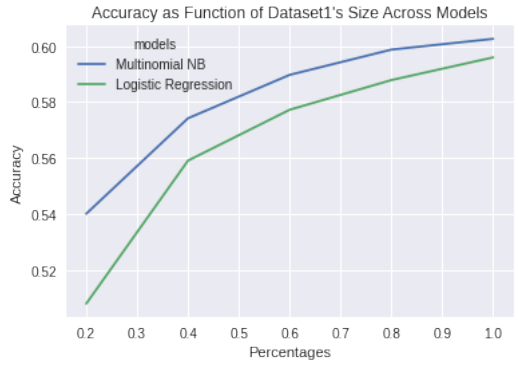

Figure 5

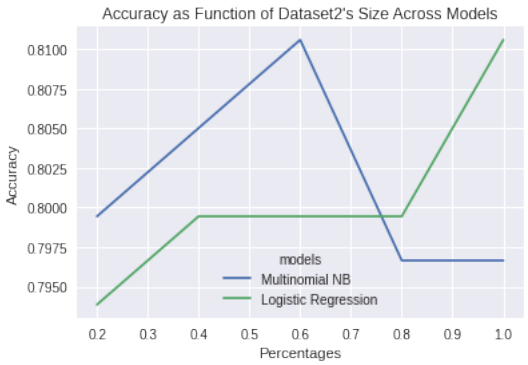

Figure 6

3.4 Final Results

We conclude our experiments by running the best performing Naive Bayes Model (multinomial Naive Bayes model) and the Logistic Regression model with the best hyperparameters chosen from our cross-validation experiments. The Logistic Regression accuracy for dataset 1, the 20 newsgroup dataset, is 59.692%, while the multinomial Naive Bayes model accuracy for the same dataset is 60.263%. The Logistic Regression accuracy for dataset 2, the sentiment140 dataset, is 81.058%, while the multinomial Naive Bayes model accuracy for the same dataset is 79.666% (table 2). For dataset 1, the better performing model is the multinomial Naive Bayes model, while the better performing model for dataset 2 is the Logistic Regression model. Since the Logistic Regression model's accuracy surpasses that of the Naive Bayes on dataset 2 a bit more than the Logistic Regression accuracy is surpassed by that of the Naive Bayes on dataset 1, we conclude that the overall winner is the Logistic Regression model.

	Dataset 1	Dataset 2
(Multinomial) Naive Bayes	**60.263%**	79.666%
Logistic regression	59.692%	**81.058%**

Table 2

4 Discussion and Conclusion

In this project, we implemented the Naive Bayes (NB) model with three different likelihood functions from scratch. We compared the performance of our model with scikit-learn's logistic regression (LR) model on two datasets: the 20 newsgroup dataset (D1) and the sentiment140 dataset (D2). In the hyperparameter tuning phase, we observed that the number of maximum iterations has little effect on the performance of the LR model, while adding L2 penalty slightly improved prediction accuracy for D1 but not D2. For the NB model, the result of using different likelihood distributions were compared against each other. The Bernoulli NB model yields the highest accuracy for D1. The fact that it outperformed the multinomial model is surprising, since having word counts as features would be more informative than only indicating the presence or absence of a word. For D2, the multinomial model gives significantly more accurate predictions than the other two versions of the NB model. However, it should be noted that this might be due to using only a portion of the dataset for the Gaussian and Bernoulli NB models. With sufficient RAM size, future work may train the Gaussian and Bernoulli NB models on all training set instances of the sentiment140 dataset and compare the results to the multinomial NB model. Lastly, we examined the performance of the models as a function of training set size. In general, prediction accuracy increases with increasing training set size, expect for multinomial NB's performance on D2.

Statement of Contributions

Yiwei worked on importing datasets, extracting features, running experiments, and writing the dataset and discussion parts of the writeup. Shelley focused on the implementation of different Naive Bayes models, the logistic regression model and cross-validation algorithms, and wrote the abstract and intro of the writeup. Mingze worked on feature selection, running experiments, and writing the results part of the manuscript.

References

[1] Sklearn.datasets.fetch$_2$0*newsgroups*.

[2] For academics - sentiment140 - a twitter sentiment analysis tool.

[3] Wang Dawei, Rayner Alfred, Joe Henry Obit, and Chin Kim On. A literature review on text classification and sentiment analysis approaches. *Lecture Notes in Electrical Engineering*, page 305–323, 2021.

[4] L. Mary Gladence, M. Karthi, and Maria Anu. A statistical comparison of logistic regression and different bayes classification methods for machine learning. *ARPN Journal of Engineering and Applied Sciences*, 10(14):5947–5953, Aug 2015.

[5] Abdulwahab O. Adi and Erbug Celebi. Classification of 20 news group with naïve bayes classifier. *2014 22nd Signal Processing and Communications Applications Conference (SIU)*, 2014.

[6] Pindi Jyothsna and M S Venugopala Rao. Text classification for news group using meachine learning. *Complexity International Journal*, 24(01), Mar 2020.

[7] rg089. A list of english stopwords.