COMP551 Project 2

Classification of Textual Data

Xiaoman Sun, Chuqi Wang, Yubai Zhang

2nd March 2022

Abstract

There are two classical language models, Logistic Regression and Naive Bayes, which have been widely used and are successful at classifying text. Various methods of extracting text features (CountVectorizer and TF-IDF Transformer) would produce different results on models. In this study, logistic regression and Naive Bayes models (Multinomial Naive Bayes and Gaussian Naive Bayes) are applied to two text datasets: 20 news group dataset and Sentiment140 dataset. The purpose of this study is to compare the performance of the Naive Bayes model and softmax regression model for two distinct textual datasets and softmax regression model on various training data sizes with their best hyper-parameters (Logistic Regression) or best model (Naive Bayes). We found that TF-IDF Transoformer method will make a better accuracy for both two models on two datasets and the number of features is proportional to the accuracy of two models. In conclusion, logistics regression has slightly better accuracy performance than Naive Bayes on both two datasets.

Contents

1. Introduction

The classification of textual data plays an essential role on standardizing search which facilitates the user's experience by simplifying navigation. There is a growing interest in news classification since news sites lack effective topic-based search functions, which brings inconvenience to web readers. Both Naive Bayes and Logistic Regression are considered to be two of the most effective text data classifiers owing to their simplicity and relatively good performance. [2] used Naive Bayes, achieving 0.98 of F1 score on 20 news group dataset and [4] applied Logistic Regression to get over 70% of accuracy with a TF method sentiment analysis dataset. The literature relating to these topics illustrates the potential of Naive Bayes and Logistic Regression Models for text classification.

In this study, we generally implemented Naive Bayes model and K-fold cross-validation model with 5-fold cross-validation on 20 news group dataset[3] and the Sentiment140 dataset[1]. Multiclass classification tasks were implemented on 20 news group and sentiment140 datasets individually, and Naive Bayes and softmax regression calculations were performed on each dataset. In the meantime, we investigated the effect of different training sizes on model performance.

2. Datasets

20 News Group Dataset

Dataset we have chosen for this project was "20 news group dataset" from scikit-learn datasets. It was basically made up of 11314 pieces of news text with its corresponding topic. The dataset was split into two subsets: training set and testing set. We started with the text data and converted text to feature vectors by extracting TF-IDF and count-vect vectors with unigram tokens. After extracting TF-IDF and count-vect vectors, the remaining data was sparse. From Figure 1, we observed that the class distribution of this dataset was relatively even among 20 topics, the minimum one was talk. religion.misc which differs 223 from the maximum value.

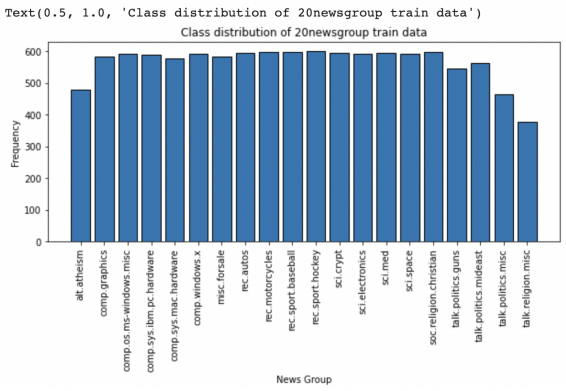

Figure 1. 20 News Group Dataset Distribution

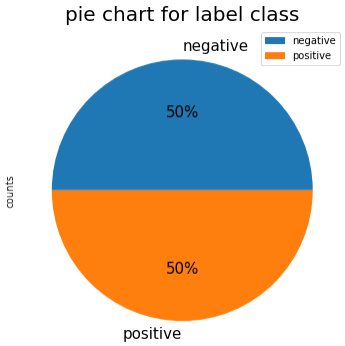

Figure 2. Sentiment140 Dataset Pie Chart

Sentiment140 Dataset

Dataset II chosen for this project was "Sentiment140 dataset" downloaded from Sentiment140 website. It was used for sentiment analysis training using customer reviews of brand, product, or topic on Twitter. It was composed of 1600359 instances, was split into a training set for 1600000 and testing set for 359. The sentiment was classified into two groups: 0(negative), 4(positive). We did this binary classification of the sentiment analysis to predict the given review is either positive or negative. From Figure 2, we observed that the dataset was distributed uniformly, with an approximately ratio of 1:1.

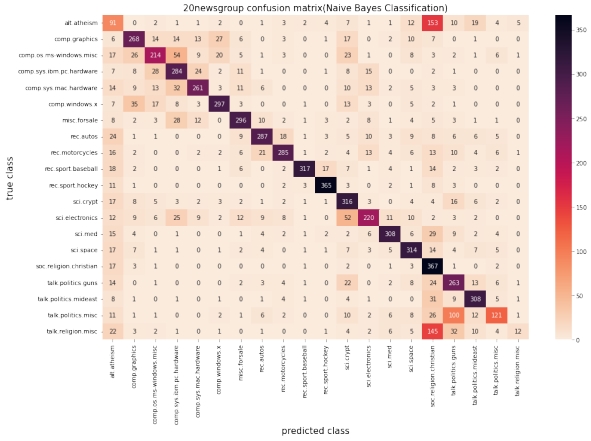

Figure 3. 20newsgroup Confusion Matrix for Naive Bayes Classification

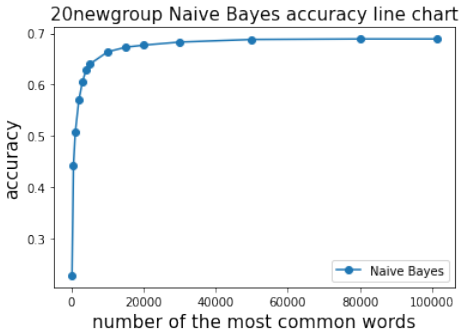

Figure 4. 20newsgroup Naive Bayes Accuracy Line Chart

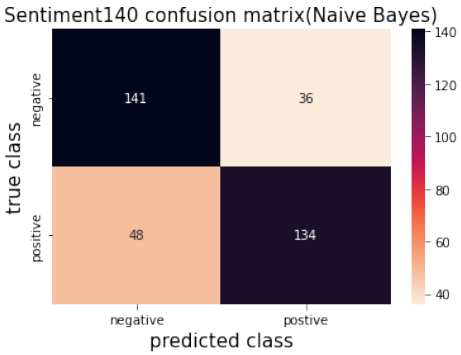

Figure 5. Sentiment140 Confusion Matrix for Naive Bayes

3. Results

3.1 Naive Bayes Result

20 News Group Dataset

For the 20 news group dataset, we first selected all of the features from the training dataset to train our own multinomial Naive Bayes model. Choosing hyper parameter alpha=0.1 and using TfidTransformer from sklearn package gave us the highest accuracy as 68.96% on the test dataset. But if we use CountVectorizer with the same alpha, the accuracy is only 59.7%. The confusion matrix is shown in figure 3. From the confusion matrix we can conclude that the multinomial Naive Bayes model has a good performance on most of the class except to alt. atheism, comp.os.ms-windows.misc, sci. electronics, talk. politics.misc and talk.religion.misc and those newsgroups that have very similar topics can be easily predicted to be the wrong group. The precision for group talk.religion.misc is only 4.78%, there are 145 news of talk.religion.misc were predicted to be soc.religion.misc group. We also investigate the impact of the max number of the most common words and accuracy performance on this dataset by passing the parameter max_features in CountVectorizer function. As Figure 4 shows, the number of the most common words from 0 to 15000, the accuracy will be rising rapidly from 0 to about 0.66 and as the number of features becomes much more and more huge, the accuracy will stay flat as the value of 0.69.

Sentiment140 Dataset

For the Sentiment140 dataset, we randomly selected 60000 data from the training dataset, with TfidTransformer method and alpha=1.5, the best accuracy we got is about 79.67%. If the hyperparameter remains the same, by using CountVectorizer function, we found that the accuracy was 78.27% which is a litter bit lower than TF-IDF Transformer method. The confusion matrix was shown in Figure 5.

Similarly to the 20 news group dataset, we changed the max_features between 0 and 66929 from CountVectorizer function and the line chart was shown in Figure 6. We found the accuracy has an extreme increase from 0 to 79% as the growth of the number of the most common words. And as the number of words become larger, the accuracy slightly decreased to about 77%.

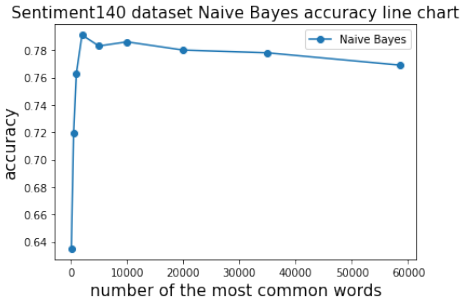

Figure 6. Sentiment140 Dataset Naive Bayes Accuracy Line Chart

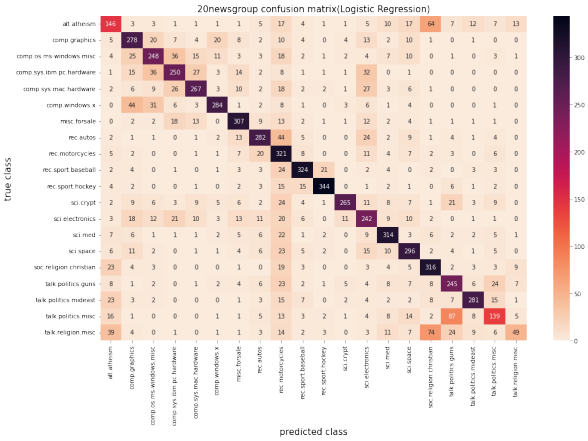

Figure 7. 20newsgroup Confusion Matrix for Logistics Regression

3.2 Softmax Regression Result

For the 20newsgroup dataset, by using the best hyperparameter of logistic regression, the best accuracy we got is 69.1%. If we look at the confusion matrix of logistic regression for this dataset and compare it with that of Naive Bayes model, we conclude that the logistics regression model has a slightly better performance for the predicted class. In addition, the group alt.atheism, talk.
politics.misc and talk. religion.misc are also difficult for softmax regression to predict. We could also found out that the relationship between the number of the most common words and the accuracy performance for logistics regression is similar to Naive Bayes on the 20newsgroup dataset.

The accuracy performance of logistic regression on sentiment140 dataset is much better than 20 news group dataset, the highest accuracy we got is 79.39% by using 60000 training dataset. The confusion matrix is shown in Figure 9. Again we altered the number of the most common words for logistics regression to see the change of accuracy. As we expected, the logistics regression accuracy line chart from Figure 10 looks similar to the that of Naive Bayes model on the same dataset.

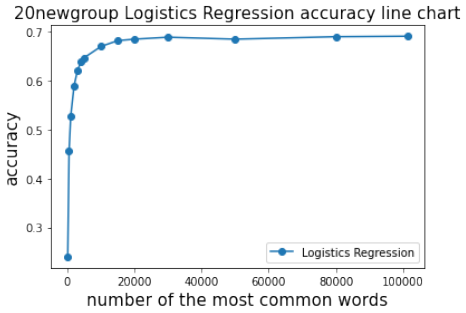

Figure 8. 20newsgroup Dataset Logistics Regression Accuracy Line Chart

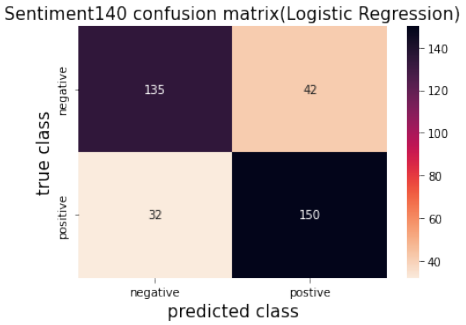

Figure 9. Sentiment140 Confusion Matrix for Logistics Regression

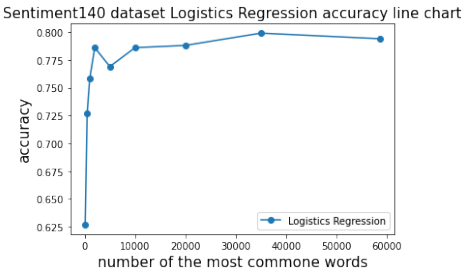

Figure 10. Sentiment140 Dataset Logistics Regression Accuracy Line Chart

3.3 Hyper-parameter Tuning

According to the instruction, we implemented the cross validation by doing some modifications to the codes of cross-validation given on the course website. we have applied the 5-fold cross-validation to both the models on both datasets. For the naive Bayes model, we decided to consider the value of alpha as our hyper-parameter for training the model. Following the class example, we also applied the cross-validation continuously on alpha from 0.1 to 1.5 to choose the best value of alpha. It has been shown that choosing **alpha=0.1** provides us the lowest error of validation based on the MSE loss function. However, when running the same cross validation repeatedly on the sentiment 140 dataset, the result is significantly different. **Alpha=1.5** gives us the lowest MSE error. On the other hand, for the softmax regression model, we decide to consider the value of C in the list of the parameter as hyper-parameter. According to the description shown on the website, we intend to have a positive float number for C. Additionally, it represents the inverse of regularization strength and smaller values specify stronger regularization. We finally perform the cross-validation continuously on the C value from 0.1 to 1.0 which is the default value. In general, the result provides us that when the C value is located close to 1.0, the MSE error will become the lowest. For 20 newsgroup dataset the best value for **C is 0.9** and for sentiment 140 dataset the best value for **C is 1.0**. As Figure 11 shows above, after selecting the best hyper-parameter, the testing accuracy also follows the same trend. On the 20 newsgroup dataset, multinomial Naive Bayes with alpha=0.1 provide us the highest testing accuracy and softmax regression with C=0.9 provide us the highest testing accuracy. On the sentiment 140 dataset, multinomial Naive Bayes with alpha= 1.5 provides us the highest testing accuracy and softmax regression with C=1.0 provides us the highest testing accuracy. Furthermore, with the Softmax Regression model, besides the C value, we also investigate another hyper-parameter which is a boolean parameter **fit intercept**. We perform the cross-validation on both datasets by setting the LogisticRegression function with default and fit intercept=False. By observing the results of this experiment, we obtain the conclusion that the model with fit intercept equals True which is default always has the lowest MSE loss. Therefore, we then run the experiment on both cases. The testing accuracy results also demonstrate the same trend. The testing accuracy of fit intercept equals to true has the highest value relative to the case when a fit intercept is set to False.

	20 newsgroup	Sentiment 140
Multinomial Naive Bayes(α=0.1)	0.6895910780669146	0.7381615598885793
Multinomial Naive Bayes(α=1.5)	0.6769782262347318	0.7966573816155988
Softmax Regression(C=0.9)	0.6909187466808284	0.7936428969359332
Softmax Regression(C=1.0)	0.6901221455124801	0.7938718662952646

Figure 11. Testing Accuracy of different Model on Different Datasets

3.4 Comparison

The following table is the general testing accuracy after choosing the best hyper-parameter when training the model on both datasets with Multinomial Naive Bayes and Softmax Regression:

	20newsgroup Dataset	Sentiment140 Dataset
Naive Bayes	68.96%	79.67%
Logistics Regression	69.09%	79.39%

Table 1. Two models' best accuracy on different datasets

From Table 1 we can conclude that for 20 news group dataset the logistics regression model has slightly higher accuracy than Naive Bayes classification. And for the sentiment 140 dataset, we found that the two models have very similar accuracy around 80%. To investigate more accurate performance on these two classification models, we split the sentiment140 training data to fit both two models. By choosing the best hyperparameter for both two models, we randomly selected 500, 1000, 5000 up to 30000 rows from the training dataset. Figure 12 illustrates that the accuracy of the two models is

Figure 12. Comparsion of two models on Sentiment140 Dataset

very similar by selecting from 0 to about 60000 data for training, however, as the number of training data becomes larger and larger, the logistics regression model has slightly better accuracy than Naive Bayes model. The accuracy of logistics regression is about 2% higher than the Naive Bayes on the sentiment140 dataset.

4. Discussion and Conclusion

In summary, we learned how to implement the Naive Bayes and K-fold cross-validation from scratch. Particularly, we gained multiple relative experiences on the model selection by utilizing the method of k-fold cross-validation. Furthermore, we also learned how to handle and preprocess the text dataset in real life which we have never done before. In addition to this, this project also provides us with experience of running experiments on large size of datasets. However, due to the limited amount of time, we were unable to try out all the hyper-parameter that are critical to the logistic regression. Therefore, doing research on these parameters and also trying to implement the logistic regression from scratch would be helpful for future applications. Lastly, based on our results on running these experiments, we noticed that in most cases, the 5-fold cross-validation provides us the correct direction on choosing the most appropriate hyper-parameter when training the model. This significant detection helps us reduce time on doing the model selection in the future.

5. State of Contributions

Xiaoman Sun: Acquire, preprocess and analyze the data, contribution on writing report
Yubai Zhang: Implemention of the K-Fold Cross Validation, contribution on writing report
Chuqi Wang: Implementation of the Naive Bayes Model,contribution on writing report

References

[1] Alec Go, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision". In: *CS224N project report, Stanford* 1.12 (2009), p. 2009.

[2] Karl-Michael Schneider. "A new feature selection score for multinomial naive Bayes text classification based on KL-divergence". In: *Proceedings of the ACL interactive poster and demonstration sessions*. 2004, pp. 186–189.

[3] Sentiment140. Ed. by empty.

[4] CM Suneera and Jay Prakash. "Performance analysis of machine learning and deep learning models for text classification". In: *2020 IEEE 17th India Council International Conference (INDICON)*. IEEE. 2020, pp. 1–6.