Analysis of Text Classification using Logistic Regression, Naive Bayes and KNN

Prateek Jeet Singh Sohi and Arian Shah Kamrani

Abstract

In this project, the effect of the performance of three classifiers known as logistic regression, Naive Bayes and KNN (K-nearest neighbors) on two datasets with a high number of samples is investigated. The first dataset contains about 20,000 documents from 20 different news categories. The purpose of this section is to implement a model to predict the category of each document based on the words used in it. The second dataset contains a collection of 1.6 million tweets that have positive or negative polarity. The goal in this section is to implement a model that can predict the polarity of a tweet based on the words used in it. In order to find the optimal parameters and also to validate the accuracy of the algorithms, k-fold cross-validation has been used in their implementation. The results show that on both datasets, logistic regression performs better than the other two classifiers and its accuracy on the first and second dataset test data is 66.52% and 81.33%, respectively. This amount is 61.97%, 80.5% for Naive Bayes and 63.98% and 40.36% for KNN. Also in this project, the optimal values of the parameters were determined for each classifier according to each data set and the results of changing the parameters were examined.

1 Introduction

Today, the importance of classification with large datasets in all fields such as medicine, education, entertainment and etc. is more important than in the past. Many large companies, such as Amazon and Netflix, use classification algorithms to categorize their customers. Two of the most important classification algorithms used in this project are logistic regression and Naive Bayes. Also in this project, famous datasets such as 20 newsgroups and sentiment 140 have been used, which have been used in previous works too.

Authors in [1] described an efficient interior-point method for solving large-scale L1-regularized logistic regression problems. This method has been tested on a large dataset of 20 newsgroups and the results have been obtained with high speed and accuracy. In [2] a new method based on Naive Bayes estimator was examined. Also a correlation factor is introduced to incorporate the correlation among different classes. in [3] authors implement Naive Bayes using sentiment140 training data and propose a method to improve classification. They also showed using SentiWordNet along with Naive Bayes can improve accuracy of classification of tweets, by providing positivity, negativity and objectivity score of words. In [4] reaction of the people of countries with different cultures towards COVID-19 pandemic with respect to 140 sentiment dataset and COVID-19 related dataset have been examined. For instance, results show Denmark and Sweden, which share many similarities, stood poles apart on the decision taken by their respective governments. Finally, authors in [5] used a boosted regression tree to predict stock market movement in the USA based on sentiment 140 dataset.

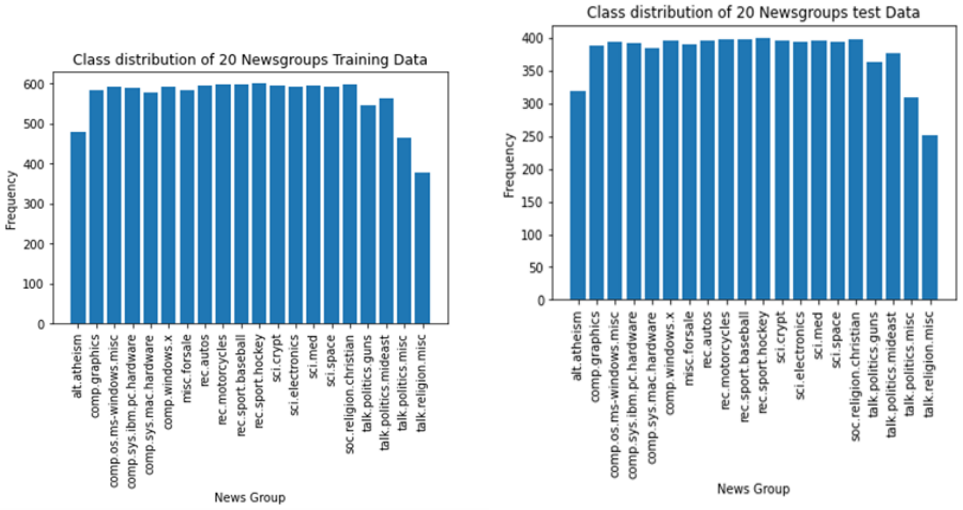
In this project, Naive Base implementation and k-fold cross validation have been done from scratch and the implemented model is tested on two datasets of 20 newsgroups and sentiment 140. The purpose of cross-validation is to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. The results of Naive Bayes are also compared with two other classifiers called logistic regression and KNN. The results show that the performance of the logistic regression classifier is better than the other two classifiers, although its running time is also longer. Finally, in this project, to improve the performance of classifiers, optimization is performed on their parameters and the best parameters that had a higher percentage of accuracy for prediction are selected.

2 Dataset

This section provides a detailed account about the characteristics of datasets used. As mentioned in the introduction section, these datasets are in text format, and in order to be able to implement machine learning algorithms on them, they need to become numerical features. As a result, the CountVectorizer function is used in this project for this purpose.

2.1 Dataset-1

20 newsgroups is a dataset containing more than 20,000 documents collected from 20 different categories. As always to work with datasets they should be cleaned, for example in this project numbers, uppercases and punctuations have been removed. The train data in this dataset is more than 11000 instances and the test data is more than 7000 instances. The frequency of each of the 20 news classes in the test and train data is shown in the figure below. class distribution in figure-1 indicates that this dataset is not biased because almost all classes have equal sample instances.

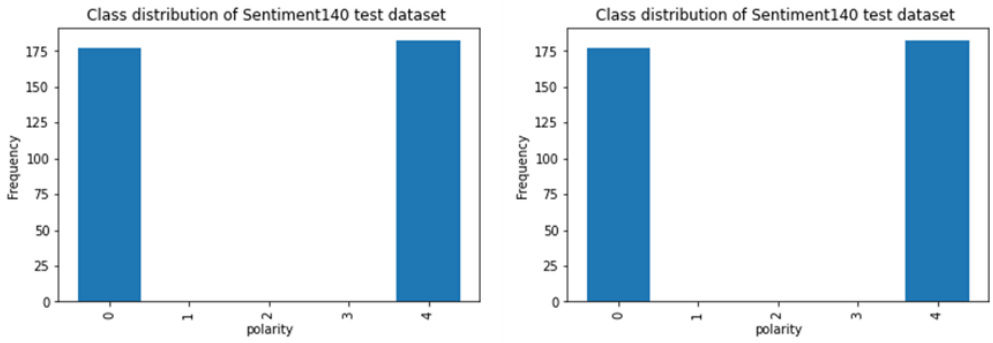


((a)) ((b))

Figure 1: (a) Distribution of training set for first dataset (b) Distribution of test set for first dataset

2.2 Dataset-2

sentiment 140 is a dataset containing more than 1.6 million tweets whose polarity is denoted by 0 and 4 for negative and positive, respectively. For the cleaning step as in the previous section, all numbers, uppercases and punctuations have been removed. Also, in this dataset some unnecessary columns such as "id", "flag", "date", "user" were removed. The frequency of each of polarities in the test and train data is shown in the figure below. As shown in the figure, there are a few points that need to be addressed. First, the number of instances for test and train data is so different that there are about 1.6 million train data, while instances for test data are about 500. Secondly, the dataset is unbiased since the number of tweets with negative and positive polarity in the test and train data is almost equal.



((a)) ((b))

Figure 2: (a) Distribution of training set for second dataset (b) Distribution of test set for first dataset

3 Results

This section contains a comprehensive analysis of the results of different classifiers with the help of 5-fold cross-validation on two datasets.

3.1 Data Set-1

((a)) ((b)) ((c))

Figure 3: Accuracy vs percentage of dataset used for training for first dataset with cross validation (a) KNN (b) Logistic Regression (c) Naive Bayes

In this section, the results of three different classifiers on first dataset are examined. As shown in Table 1, when the entire training dataset is used, the accuracy of test dataset with the logistic regression algorithm is higher than the other two algorithms. In this case, the accuracy of logistic regression is 66.52% and the accuracy of Naive Bayes and KNN are 61.97% and 63.98%, respectively. The accuracy of the train data also shows that while all the train data has been used, the accuracy of the logistic regression, Naive Bayes and KNN algorithms is 98.08, 93.63 and 83.47, respectively. Table 1 and Figure 3 show that by gradually changing the size of the training dataset from 100% to 20%, although the accuracy of training dataset remains constant in almost all cases, the accuracy of test dataset gradually decreases and results in over-fitting. For example, the accuracy of logistic regression, Naive Bayes and KNN algorithms on test data in the case where 20% of the train data is used compared to the case where all train data is used, has decreased 13.19%, 39.37% and

DataSet-1	Naive Bayes (Multinomial)		Logistic Regression		KNN K=30	
Training Size	Training accuracy	Test accuracy	Training accuracy	Test accuracy	Training accuracy	Test accuracy
20%	93.31	37.57	99.17	57.75	77.16	52.52
40%	93.83	53.49	98.67	62.62	81.27	59.22
60%	94.76	59.59	98.45	64.03	82.04	61.60
80%	93.77	60.19	98.14	64.69	82.62	62.85
100%	93.62	61.97	98.08	66.52	83.47	63.98
DataSet-2	Naive Bayes (Multinomial)		Logistic Regression		KNN K=100	
Training Size	Training accuracy	Test accuracy	Training accuracy	Test accuracy	Training accuracy	Test accuracy
20%	84.69	80.50	82.75	80.50	70.90	69.35
40%	83.92	78.83	81.97	81.05	56.64	53.76
60%	83.47	81.89	82.35	81.89	60.08	47.63
80%	83.145	80.77	82.66	81.05	63.80	50.60
100%	82.93	80.50	82.63	81.33	66.23	40.36

Table 1: Training and test accuracy of three classifiers on different portions of training datasets.

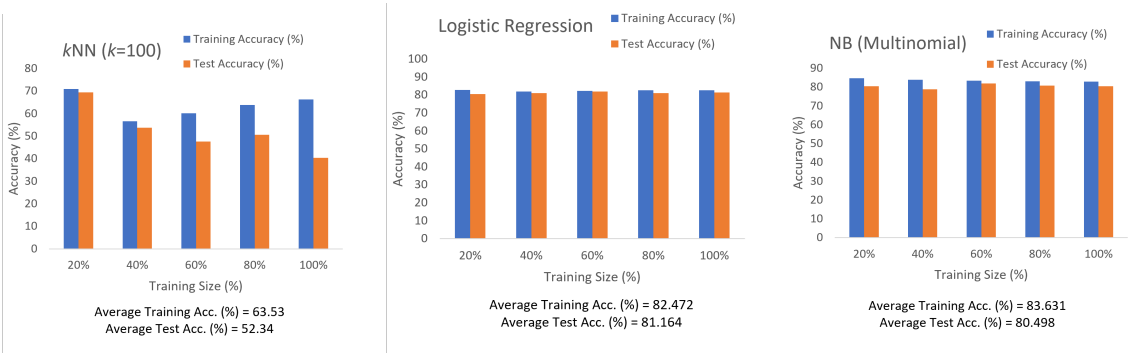
Number of iterations	10	20	30	40	50	60	70	80	90	100
L1(training)	83.70	83.55	83.48	83.41	83.43	83.46	83.38	83.39	83.40	83.43
L1 (testing)	55.43	55.09	54.83	54.71	54.66	54.64	54.58	54.56	54.59	54.59
L2 (training)	98.07	98.09	98.09	98.09	98.08	98.08	98.08	98.08	98.08	98.08
L2(testing)	65.56	65.54	65.53	65.53	65.54	65.52	65.56	65.56	65.57	66.52

Table 2: Impact of Iterations of gradient decent and regularization on training and test set for dataset-1

17.91% respectively. All the results mentioned above were calculated using 5-fold cross-validation, as well as the optimal hyper parameters mentioned in Table 1. As we know, Naive Bayes does not have a hyper parameter that can be tuned, so cross validation was used for model selection and Multinomial was selected. For logistic regression, parameters l1 and l2 were examined and l2 was selected as the optimal parameter. Table 2 shows the effect of changing iterations of gradient descent method on the accuracy of logistic regression for the first dataset. The results show that the accuracy increases with increasing iteration. For KNN, K = 30 has been selected.

3.2 Dataset-2

In this section, the performance of three classifiers on dataset2 is examined. Table 1 shows that logistic regression and Naive Bayes perform better on dataset2 compared to dataset1 and have higher accuracy on test data. The first reason for this can be the number of instances in dataset2, which is more than 100 times that of dataset1, the second reason can be related to the smaller number of classes in dataset2 than dataset1. As shown in Table 1, when the entire training dataset is used, the accuracy of test dataset with the logistic regression algorithm is higher than the other two algorithms. In this case, the accuracy of logistic regression is 81.33% and the accuracy of Naive Bayes and KNN are 80.50% and 40.36%, respectively. In this dataset, unlike dataset1, the accuracy of train and test data is close to each other, which indicates that the implemented model



((a)) ((b)) ((c))

Figure 4: Accuracy vs percentage of dataset used for training for second dataset with cross validation (a) KNN (b) Logistic Regression (c) Naive Bayes

Number of iterations	10	20	30	40	50	60	70	80	90	100
L1(testing)	80.16	80.16	82.51	81.69	81.79	82.15	81.00	80.33	81.33	81.03
L1(training)	80.47	81.32	82.34	82.11	82.16	82.41	81.60	81.70	82.81	83.36
L2 (testing)	81.16	81.16	81.00	81.89	81.89	82.17	81.05	81.33	81.33	82.63
L2(training)	80.47	81.32	81.77	82.21	82.36	82.49	82.59	82.70	82.71	81.33

Table 3: Impact of Iterations of gradient decent and regularization on training and test set for dataset-2

performance is acceptable. Also, unlike dataset1, according to the results of Table 1 and Figure4, the effect of training dataset resizing on test data is very small, which is due to the high number of instances in this dataset. As in the previous section, it should be noted that all the above results are obtained with 5-fold cross-validation. Table 3 shows the effect of changing iterations of gradient descent method on the accuracy of logistic regression for the second dataset.
As can be seen, the KNN results on the second dataset are not ideal. One reason could be the large size of this dataset because KNN does not perform well on large datasets. One way to improve the results is to find the optimal parameter K, which due to the fact that this algorithm is not one of the minimum requirements of the project and is mentioned as an extra method, it was not possible for our team to find the optimal parameter K according to the project time limit. .

3.3 Logistic regression vs Naive Bayes

The results on the two datasets used in this project showed that the logistic regression performance was better than Naive Bayes. Of course, from the beginning it was expected that Naive Bayes would perform worse than logistic regression in large datasets, given the assumptions made in its modeling. Logistic regression accuracy was 7.34% and 1.03% higher than Naive Bayes in dataset1 and dataset2, respectively.

4 Conclusion

The aim of this project was to determine the effect of different classifiers (logistic regression, Naive Bayes and KNN) on two datasets of 20 newsgroups and sentiment 140. First, the data was cleaned and the distribution of different classes on train and test data was examined. Then the Naive Bayes

algorithm and k-fold cross validation were implemented from scratch. Then the optimal parameters for different classifiers were determined and the results showed that the logistic regression algorithm performed better than Naive Bayes and KNN on 2 mentioned datasets. For future work, for example, people's personality type can also be considered as a feature for prediction of polarity of tweets. For example, there are people who are pessimist and their tweets have negative polarity. On the other hand, there are people who are optimist and the polarity of their tweets are mostly positive. Also, determination of optimal hyper parameter K for KNN can be considered.

5 Statement of Contribution

All team members contributed equally to mini project 2. We had regular meetings via zoom and developed each task of mini project 2 together.

References

[1] K. Koh, S. Kim, and S. Boyd, "An Interior-Point Method for Large-Scale Logistic Regression," Jmlr '07, vol. 8, pp. 1519–1555, 2007.

[2] J. Chen, Z. Dai, J. Duan, H. Matzinger, and I. Popescu, "Naive bayes with correlation factor for text classification problem," Proc. - 18th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2019, pp. 1051–1056, 2019.

[3] A. Goel, J. Gautam, and S. Kumar, "Real time sentiment analysis of tweets using Naive Bayes," Proc. 2016 2nd Int. Conf. Next Gener. Comput. Technol. NGCT 2016, no. October, pp. 257–261, 2017.

[4] A. S. Imran, S. M. Daudpota, Z. Kastrati, and R. Batra, "Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on covid-19 related tweets," IEEE Access, vol. 8, pp. 181074–181090, 2020.

[5] P. Chakraborty, U. S. Pria, M. R. A. H. Rony and M. A. Majumdar, "Predicting stock movement using sentiment analysis of Twitter feed," 2017 6th International Conference on Informatics, Electronics and Vision 2017 7th International Symposium in Computational Medical and Health Technology (ICIEV-ISCMHT), 2017, pp. 1-6, doi: 10.1109/ICIEV.2017.8338584.