Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification

Dr Gurinder Singh
Group Vice Chancellor
Amity Universities
India
gsingh@amity.edu

Prof. Bhawna Kumar
Vice President-RBEF
Amity University
Noida, India
bkumar@amity.edu

Dr Loveleen Gaur
Associate Professor
AIBS, Amity University
Noida, India
lgaur@amity.edu

Akriti Tyagi
B. Tech., Computer Science & Engineering
Amity University
Noida, India
akriti1997tyagi@gmail.com

***Abstract*— Document/Text Classification has become an important area in the field of Machine Learning. On account of its wide applications in business, ham/spam filtering, health, e-commerce, social media sentiment, product sentiment among customers etc., various approaches have been devised to accurately predict the category or to classify any of the new text/document under consideration. Nowadays, news articles in the newspaper present various kinds of sentiments or inclination of the news article towards a negative or positive sentiment and hence, the content of the news can actively be used to judge the impact on the reader. The paper aims to predict that whether the sentiment of the news article is positive or negative using the two popular approaches of Naïve Bayes Text Categorization i.e. Multivariate Bernoulli Naïve Bayes Classification and Multinomial Naïve Bayes Classification. Also, the research aims to identify that which approach between the given two approaches perform better for the given dataset.**

Keywords—Text Classification, Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Accuracy Comparison

I. INTRODUCTION

Text Classification with the help of machine learning can be achieved using many machine learning algorithms devised for these tasks over various years and which have proven to achieve high accuracies of prediction and are thus, highly reliable. Some the most popular algorithms used for this purpose are Naïve Bayes Classification algorithms, Support Vector Machines and Deep Learning algorithms using architectures such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) [7].

In this research, we make use of the Naïve Bayes family of algorithms to achieve the make predictions. Naïve Bayes algorithms are the statistical classification algorithms based on the Bayes' theorem helps us find the conditional probability of happening of two events based on the probabilities of happening of each individual event. [11] These algorithms work on the principle that each attribute is independent of the other. Three popular Naïve Bayes Classifiers are Gaussian Naïve Bayes Classifier, Multinomial Naïve Bayes Classifier and Bernoulli Naïve Bayes Classifier. This research focuses on the latter two algorithms and their comparison only. Both of these algorithms are used for document classification but considerably differ in their approaches to classify the documents [11]. We provide a gentle overview of these algorithms and further discuss these two approaches in the following sections.

Multinomial Naïve Bayes classifier works on the concept of term frequency which means that how many times does the word occur in a document. This model tells two facts that whether the word occur in a document or not as well as its frequency in that document. While predicting the polarity of a new news article, we multiply the probabilities of the occurrence of all the words in the article against both the polarities and the one which is higher gives the polarity of this article [12].

On the other hand, Bernoulli Naïve Bayes Classifier works on the binary concept that whether the term occurs in a document [7] or not but unlike Multinomial Naïve Bayes, it does not tell about the term frequency. While predicting the polarity of a new news article, we multiply the probabilities of the occurrence of all the words in the article and also the probabilities of non-occurrence of words which do not occur in the article against both the polarities and the one which is higher gives the polarity of this article [15].

This paper aims to use both of these algorithms for classification of textual news articles on many important events that happened in India in 2018. We divide the dataset into training set and the testing set and classify the test set by building a Bernoulli Naïve Bayes model and Multinomial Naïve Bayes model on the cleaned and pre-processed training set.

II. DATASET

The dataset comprises of 312 records which has two columns namely 'news' which has the news article as its contents and 'polarity' which tells about the inclination of the news article in terms of positive (indicated by 1) and negative (indicated by -1).

	news	polarity
0	A day after they were blocked while proceeding...	-1
1	Train 18, India's first indigenously built eng...	1
2	Pudukkottai is not alone. Out of the 95 distri...	-1
3	A petition was filed by an advocate in the Pun...	-1
4	The driver in his statement on Saturday said t...	-1

Fig. 1. Snapshot of Data Under Analysis

III. DATA PREPROCESSING

After loading the dataset into the workspace and before building a model, the text should be cleaned and pre-processed to achieve better accuracy [17]. Data Pre-processing means to change the data in a way that it is more effective while building a model by minimizing the least important features in the data. Basic pre-processing steps include the following [8][20]:

a). Lowercasing:

The data is converted into the lowercase so that the uppercase and lowercase words with same meaning are not treated differently [18].

b). Tokenization:

Tokenization refers to the process of converting an entire text into a series of tokens with each token being separate and independent of each other.

c). Punctuation Removal:

Punctuations hold no actual importance when it comes to the analysis of the data. So, the better practice of data analysis involves the removal of punctuation beforehand.

d). Stopwords Removal:

There are some words in the tokenized text which do not account to any significant concept or result but can have a significant effect on classifier. It is better to remove such words beforehand.

The dataset is hence ready for building a classifier [19].

IV. NAÏVE BAYES CLASSIFIERS

A. Multinomial Naïve Bayes Classifier

The multinomial model is designed to determine term frequency i.e. the number of times a term occurs in a document [1]. Considering the fact that a term may be pivotal in deciding the sentiment of the document, this property of this model makes it a decent choice for document classification [13]. Also, term frequency is also helpful in deciding that whether the term is useful in our analysis or not. Sometimes, a term may be present in a document many times which increases its term frequency [3] in this model but at the same time, it may also be a stopword which potentially adds no meaning to the document but possesses a high term frequency so, such words must be removed first to gain better accuracy from this algorithm [2][9].

Multinomial Naïve Bayes Classifier can be formulated as follows:

A news article 'n' being of polarity 'p' is calculated as [10]:

$$P(p|n) \propto P(p) \prod_{1\le k\le nd} P(t_k|p) \quad .. \text{(i)}$$

where $P(t_k|p)$: represents the conditional probability that whether the term t_k occurs in a news article of polarity p which is calculated as follows:

$$P(t_k|p) = \frac{count(t_k|p)+1}{count(t_p)+|V|} \quad .. \text{(ii)}$$

Here, count $(t_k|p)$ means the number of times the term tk occurs in the news articles which have polarity p and count (t_p) means the total number of tokens present in the news articles of polarity p.

Also, 1 and |V| are added as smoothing constants which are added to avoid the mishaps in the calculation when the term does not occur at all in the news article or the news article is empty or null. This concept is better known as Laplace Smoothing. |V| is the number of terms in the total vocabulary of news articles.

P(p): represents the prior probability of news article being of polarity p which is calculated as follows:

$$P(p) = \frac{Number\ of\ news\ articles\ of\ polarity\ p}{Total\ number\ of\ news\ articles} \quad .. \text{(iii)}$$

n_d: represents number of tokens in a news article

t_k: represents the k^{th} token in the news article

The probability P(p|n) is calculated for both i.e. the positive polarity as well as the negative polarity and the maximum is considered to be the predicted polarity for a news article.

B. Multivariate Bernoulli Naïve Bayes

In the multivariate Bernoulli Naïve Bayes Classifier algorithm, features are independent binary variables which represents that whether a term is present in the document under consideration or not [1][14]. Being slightly similar to the multinomial model in the classification process, this algorithm is also a popular approach for text classification tasks but it differs from the multinomial approach in the aspect that multinomial approach takes into account the term frequencies whereas Bernoulli approach is only interested in devising that whether a term is present or absent in the document under consideration [5].

Multivariate Bernoulli Naïve Bayes Classifier can be formulated as follows [9]:

A news article 'n' being of polarity 'p' is calculated as [16]:

$$P(p|n) \propto P(p) \prod_{1\le k\le nd} P(t_k|p)(1 - P(t_k'|p)) \quad .. \text{(iv)}$$

where $P(t_k|p)$: represents the conditional probability of the occurring term t_k in a news article of polarity p and $P(t_{k'}|p)$ represents the conditional probability of non-occurring term $t_{k'}$ in a news article [2]. Both of these conditional probabilities are given as:

$$P(t_k|p) = \frac{count(t_k|p)+1}{count(N_p)+2} \quad .. \text{(v)}$$

$$P(t_k'|p) = \frac{count(t_k'|p)+1}{count(N_p)+2} \quad .. \text{(vi)}$$

Here, count $(t_k|p)$ means the count of occurrence of a term in news articles of polarity p where for a particular news article, the value can be 0 or 1 and count (N_p) means the total number of news articles having polarity as p.

P(p): represents the prior probability of news article being of polarity p which is calculated as follows:

$$P(p) = \frac{Number\ of\ news\ articles\ of\ polarity\ p}{Total\ number\ of\ news\ articles} \quad .. \text{(vii)}$$

n_d: represents number of tokens in a news article

t_k: represents the k^{th} token in the news article

V. EXPERIMENTAL RESULTS

This section summarizes the results obtained using the Multinomial Naïve Bayes and Multi-variate Bernoulli Naïve Bayes on the dataset of news articles of 2018(India). It was observed that on the given dataset Multinomial Naïve Bayes performs better than Multivariate Bernoulli Naïve Bayes.

The brief on the dataset structure was already discussed in an earlier section. In this section, we discuss the results.

For all the necessary model building use for this paper, the number of features taken into consideration are 15000.

The Multinomial and the Bernoulli Naïve Bayes classifiers have been generated using the following code on the training set [4].

```
from sklearn.naive_bayes import MultinomialNB,BernoulliNB
mnb = MultinomialNB()
mnb.fit(news_train_vect, polarity_train)
mnb.score(news_train_vect, polarity_train)
bnb = BernoulliNB()
bnb.fit(news_train_vect, polarity_train)
bnb.score(news_train_vect, polarity_train)
```

Fig. 2. Code for building classifiers in python

Confusion matrix for Multinomial Naïve Bayes is displayed in Figure-3. a) indicates that 41 negative polarity news was predicted as negative and was actually a negative but 12 positive polarity bearing news have also been predicted as false negative. Similarly, 28 positive news articles are predicted as positive and are actually positive, but 13 negative polarity of news articles have falsely been predicted as positive.

Similarly, analyzing the confusion matrix for Bernoulli Naïve Bayes displayed in Figure-3.b) indicates that 38 negative polarity news was predicted as negative and was actually a negative but 13 positive polarity bearing news have also been predicted as false negative. Similarly, 27 positive news articles are predicted as positive and are actually positive, but 16 negative polarity of news articles have falsely been predicted as positive.

Hence, from the confusion matrices it is evident that the performance of Multinomial Naïve Bayes is better than the performance of Bernoulli Naïve Bayes on the given dataset.

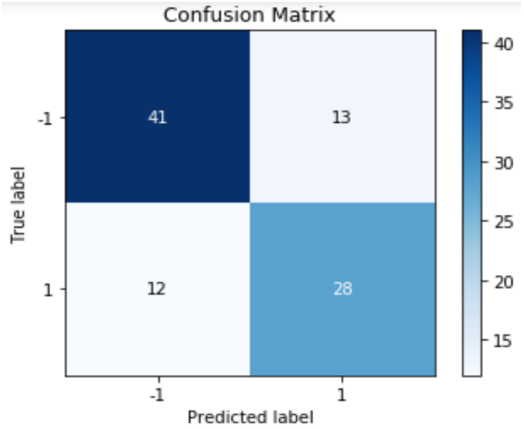

Fig. 3. a) Confusion matrix for Multinomial Naïve Bayes Classifier

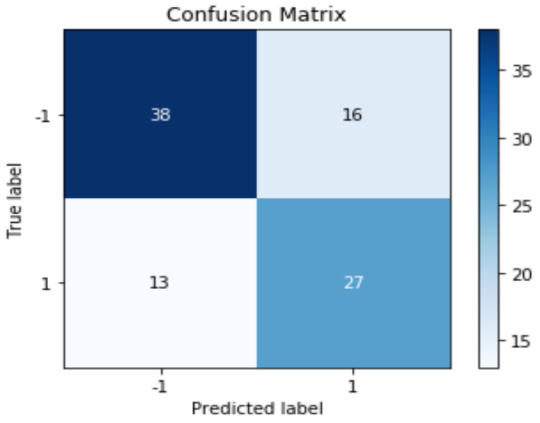

Fig. 3. b) Confusion matrix for Bernoulli Naïve Bayes Classifier

To formalize our results indicated by the confusion matrices, we use the following code to depict the difference in accuracy (Figure-5. a)) and performance of the two classification algorithms using a ROC (Receiver Operating Characteristics) curve (Figure-5. b)) which is plotted between TPR (True Positive Rate) and FPR (False Positive Rate) [6].

```
bnb_accuracy=(accuracy_score(polarity_test,
                             polarity_pred_bnb) * 100)
mnb_accuracy=(accuracy_score(polarity_test,
                             polarity_pred_mnb) * 100)
label=['Bernoulli Naive Bayes','Multinomial Naive Bayes']
index = np.arange(len(label))
acc=[accuracy_score(polarity_test, polarity_pred_bnb) *
     100,accuracy_score(polarity_test, polarity_pred_mnb) * 100]
plt.bar(index,acc)
plt.xlabel('Classifier', fontsize=15)
plt.ylabel('Accuracy', fontsize=15)
plt.xticks(index, label, fontsize=10, rotation=0)
plt.title('Comparison of accuracies')
plt.show()
fprb, tprb, thresholdsb = metrics.roc_curve(polarity_test, polarity_pred_bnb, pos_label=1)
fprm, tprm, thresholdsm = metrics.roc_curve(polarity_test, polarity_pred_mnb, pos_label=1)
plt.plot(fprm,tprm,label='Mltinomial Naive Bayes')
plt.plot(fprb,tprb,label='Bernoulli Naive Bayes')
plt.legend()
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.show()
```

Fig. 4. Code for Accuracy comparison Bar Graph and ROC Curve

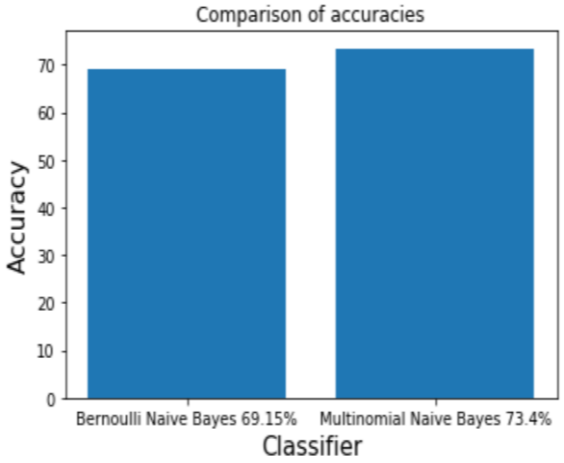

Fig. 5. a) Comparison of accuracies of two depicted by a Bar graph

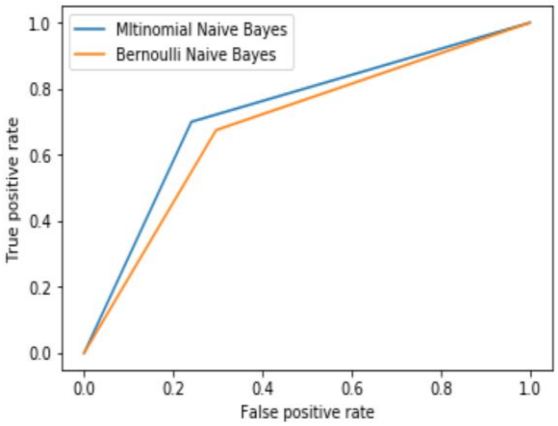

Fig. 5. b) ROC curve indicating that TPR is greater algorithms for Multinomial Naïve Bayes classifier.

It is evident from the accuracy bar graph and the ROC curve that the performance of Multinomial Naïve Bayes on the given dataset is more than the performance of Bernoulli Naïve Bayes because it has an accuracy of 73.404 % which is greater than that of Bernoulli Naïve Bayes and also, it is well evident from the ROC curve that the true positive rate is greater for Multinomial Naïve Bayes which indicates the performance-wise superiority of Multinomial Naïve Bayes.

VI. Conclusion

It has been concluded from this research that Multinomial Naïve Bayes performs slightly better than Bernoulli Naïve Bayes on dataset with lesser number of records (312 records in this case) but Multinomial Naïve Bayes reaches an accuracy of about 73 percent only which is not very efficient. This complies with the fact that it is very difficult to achieve high accuracies with less amount of data and more data will lead to greater accuracies with both the algorithms discussed. Here, the authors also conclude that although Multinomial Naïve Bayes provides greater accuracy but the difference in accuracies is not very significant as Bernoulli Naïve Bayes also provides an accuracy of almost 69 percent which implies that the performance of these algorithms does not differ much on the given dataset.

VII. Future work

The future prospects of this work lie in achieving a striking difference between these two algorithms by increasing the size of the dataset to achieve high degrees of accuracies with both the models. The increase in size of the dataset will provide an increased number of features and hence, the feature extraction and modelling process will achieve correctness and accuracy in terms of predicting the sentiment of news article on the reader.

References

[1] Andrew McCallum and Kamal Nigam, “A Comparison of Event Models for Naive Bayes Text Classification” AAA-I 98 workshop on Learning for Text Categorization

[2] The Bernoulli Model, https://nlp.stanford.edu/IR book/html/htmledition/the-bernoulli-model-1.html

[3] Bartosz Goralewicz, “The TF*IDF algorithm explained” published March 6, 2018, https://www.elephate.com/blog/what-is-tf-idf/.

[4] Brendan Martin and Nikos Koufos, “Predicting Reddit news sentiment with Naïve Bayes and other text classifiers”, https://www.learndatasci.com/tutorials/predicting-reddit-news-sentiment-naive-bayes-text-classifiers/

[5] *Python Data Science Handbook* by Jake VanderPlas, “In Depth: Naïve Bayes Classification”, https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html

[6] Sarang Narkhede, Understanding AUC-ROC , Curve , https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

[7] Text Classification, https://monkeylearn.com/text-classification/

[8] Text Data Preprocessing: A walkthrough in Python, https://www.kdnuggets.com/2018/03/text-data-preprocessing-walkthrough-python.html

[9] Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes,” Multinomial Naive Bayes for Text Categorization Revisited”

[10] Syed Sadat Nazrul, Multinomial Naïve Bayes Classifier for Text Classification (Python), https://towardsdatascience.com/multinomial-naive-bayes-classifier-for-text-analysis-python-8dd6825ece67

[11] Naïve Bayes Classifier, https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[12] Document Classification using Multinomial Naïve Bayes, https://www.3pillarglobal.com/insights/document-classification-using-multinomial-naive-bayes-classifier

[13] Applying Multinomial Naïve Bayes to NLP problems, https://www.geeksforgeeks.org/applying-multinomial-naive-bayes-to-nlp-problems/

[14] Bernoulli Naïve Bayes Classifier, https://chrisalbon.com/machine_learning/naive_bayes/bernoulli_naive_bayes_classifier/

[15] Bernoulli Naïve Bayes Classifier, https://mattshomepage.com/articles/2016/Jun/07/bernoulli_nb/

[16] CS340 Machine learning Naïve Bayes classifiers, https://www.cs.ubc.ca/~murphyk/Teaching/CS340-Fall07/NB.pdf

[17] How to clean text for Machine Learning with Python?, https://machinelearningmastery.com/clean-text-machine-learning-python/

[18] A General Approach to preprocessing text data, https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html

[19] Text Preprocessing in Python: Steps, Tools and Examples, https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908

[20] Shivangi Sareen, Data Preprocessing in Python, https://towardsdatascience.com/data-preprocessing-in-python-6f04e6c2cb70