Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, No. 5, May : 2023

SPAM DETECTION IN SMS (TEXT) DATA USING MULTINOMIAL NAÏVE BAYES CLASSIFIER

Harish Nikavade*[1], Student, B. Tech, CSE, Medi-Caps University, Indore
Dr. Hemlata Patel[2], Mr. Vivek Kumar Gupta[2] Professor, Medi-Caps University, Indore

ABSTRACT
During the last few decades, more and more individuals are using mobile devices. Both Smart phones and basic phones support SMS (short message service), a text messaging service. Consequently, there was a sharp increase in SMS traffic. The ability to identify SMS spam can be significantly impacted by the use of well-known words, phrases, abbreviations, and idioms. Traditional machine learning techniques and deep learning techniques have been compared. This paper compares various classification methods on a variety of datasets gathered from prior studies and evaluates them according to their accuracy, precision, recall, and CAP Curve.
Keywords: Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Accuracy Evaluation, text Classification.

Introduction:
A spam message is an unwanted and unsolicited message, typically sent in bulk to a large number of recipients without their consent. Spam messages can take various forms, including emails, text messages, social media messages, and even phone calls. The content of spam messages can range from advertisements for products or services to fraudulent messages aimed at obtaining personal information or financial gain. Spam messages are frequently sent by automated programs or bots, and they can be annoying, intrusive, and potentially harmful if they contain malicious content.
It is important to understand and identify spam messages because they can have several negative impacts. Some of the reasons why it is important to be aware of spam messages include Security risks, Time wasting, Resource waste, Reputation damage, Legal issues.
Text classification with machine learning can be accomplished using many machine learning algorithms that have been developed for these tasks over the years and have demonstrated to achieve high prediction accuracies and are thus highly reliable. Some of the most common algorithms for this purpose are Nave Bayes Classification algorithms, Support Vector Machines, and Deep Learning algorithms based on architectures such as convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).
In this research, we forecast using the Nave Bayes family of algorithms. The Bayes' theorem-based statistical classification algorithms assist us in determining the conditional chance of two events occurring based on the probabilities of each event occurring separately. These methods work under the assumption that every characteristic is distinct from every other characteristic. There are three popular Nave Bayes classifiers: the Gaussian, the Multinomial, and the Bernoulli.Only the latter two techniques are examined and contrasted in this research.

Although both of these algorithms are used for document classification, they take very different ways to doing so. Before examining these two strategies in greater detail in the following sections, we give a short overview of these algorithms.
Multinomial Naive Bayes (MNB) is a probabilistic machine learning algorithm that is commonly used for text classification tasks such as sentiment analysis, spam detection, and topic classification. MNB is a variant of the Naive Bayes algorithm, which is based on the Bayes theorem of probability. MNB is well-suited for text classification tasks because it assumes that the presence of a term in a document is independent of the presence of other terms in the same document. This assumption,

known as "naive" assumption, simplifies the calculations and reduces the complexity of the algorithm.
Bernoulli Naive Bayes (BNB) is a variant of the Naive Bayes algorithm that is commonly used for binary classification tasks, where each instance has a set of binary features (i.e., features that can take on only two values, such as "yes" or "no" or "1" or "0"). BNB is particularly suited to problems where the features are boolean variables representing the presence or absence of a particular feature in the data.

Literature Survey

Applying ML and DL techniques for spam detection is not a new era. Previously, various researchers applied ML techniques for classification SMS spam [1].In this system applied several text classification algorithms, including the Multivariate Bernoulli Naïve Bayes model with TF-IDF feature weighting, on a variety of text classification tasks [2]. dataset contains 5169 SMS messages labeled as either spam or ham (non-spam). I have used this dataset to develop a spam detection algorithm based on the Bag-of-Words (Bow) model and achieved an accuracy of 95.93%[3]. applied here several data pre-processing techniques which includes Lowercasing, Tokenization, Punctuation Removal, Stopwords Removal, Bag of words[3]. applied a convolutional neural network (CNN) to extract features from the SMS messages and then use a fully connected neural network for classification. The proposed approach achieves high accuracy in SMS spam detection[5]. applied classify the techniques into three categories: rule-based, machine learning-based, and hybrid approaches. They provide a detailed analysis of each technique and highlight their strengths and weaknesses[6]. also discuss various challenges in SMS spam detection, such as data imbalance, feature selection, and cross-language detection[8].

Proposed Methodology

Proposed Model**:-**First of all, we collected an SMS spam dataset from the kaggle repository. Later, we applied different text preprocessing techniques to clean the dataset. Then we applied, we applied various machine learning algorithms and LSTM. The proposed model was depicted in Figure-1.

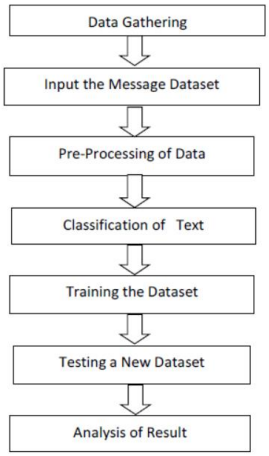

Fig. 1. Proposed Framework for SMS Spam Detection

Data Set

The dataset contains 5169 entries with two columns: 'text,' which contains the news article and 'target,' whichindicates the inclination of the news article in terms of positive (indicated by 0) and negative (indicated by 1).

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, No. 5, May : 2023

	target	text
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

Data Preprocessing

To improve accuracy, the text should be cleaned and preprocessed after loading the dataset into the workspace and before constructing a model. Data Preprocessing means changing the data so that it is more effective when building a model by removing the least essential features.

Pre-Processing stages are as follows :

a). Lowercasing: The data is converted to lowercase so that uppercase and lowercasewords with the same meaning are not handled differently.

b). Tokenization**:** A text is tokenized when it is broken down into a series of unique tokens that are unrelated to one another.

c). Punctuation Removal**:** Punctuation is irrelevant when it comes to data processing. As a result, improved data analysis practice entails removing punctuation beforehand.

d). Stopwords Removal**:** Some tokenized text words do not account for any important concept or result, but they can have a significant impact on the classifier. It is preferable to remove such phrases in advance. As a result, the dataset is now available for classification.

e).Bag of words**:**since transformed text is in words, we're transforming it to vectors with Count Vectorizer. We vectorize the changed text because it is in text form and we want it in numerical form.

f).TF-IDF: TF-IDF (Term Frequency-Inverse Document Frequency) Vectorizer is a widely used text feature extraction technique in Natural Language Processing (NLP) that aims to convert text data into numerical vectors that can be used as input to machine learning algorithms.

Classifier used

Multinomial Naïve Bayes Classifier: A term's frequency, or how frequently it occurs in a text, can be determined using the multinomial model. The feature of this model makes it an excellent choice for document classification, given that a term can be crucial in figuring out the sentiment of a document. In deciding whether a word will be helpful in our analysis, term frequency is also helpful. In some cases, a term appears more than once in a document, increasing its term frequency in this model. However, it may also be a stopword that may not contribute any meaning to the document but has a high term frequency; as a result, such words must be eliminated first to increase the accuracy of this algorithm.

The formula used in Multinomial Naïve Bayes algorithm is as follows:

$P(y \mid x_1, x_2, ...,x_n) = P(y) * P(x_1 \mid y) * P(x_2 \mid y) * ... * P(x_n \mid y)$

Where:

$P(y)$ is the prior probability of class y.

$P(x_i \mid y)$ is the probability of observing feature x_i in a document of class y, which is estimated using the training data.

$x_1, x_2, ...,x_n$ are the features (or words) in the document.

Multivariate Bernoulli Naïve Bayes: A term's frequency, or how frequently it occurs in a text, can be determined using the multinomial model. The feature of this model makes it an excellent choice for

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, No. 5, May : 2023

document classification, given that a term can be crucial in figuring out the sentiment of a document. In deciding whether a word will be helpful in our analysis, term frequency is also helpful. In some cases, a term appears more than once in a document, increasing its term frequency in this model. However, it may also be a stopword that may not contribute any meaning to the document but has a high term frequency; as a result, such words must be eliminated first to increase the accuracy of this algorithm.
The formula for the Multivariate Bernoulli Naïve Bayes algorithm is as follows:

$P(y \mid x_1, x_2, ...,x_n) = P(y) * \prod(P(x_i \mid y)^{x_i} * (1 - P(x_i \mid y))^{(1 - x_i)})$
Where:
P(y) is the prior probability of class y.
P(x_i | y) is the probability of observing feature x_i in a document of class y, which is estimated using the training data.
x_1, x_2, ...,x_n are the binary features (or words) in the document, where x_i = 1 if the feature is present in the document, and x_i = 0 otherwise.
$\prod$ represents the product over all features i.

Experimental results
This part provides a summary of the findings from the Multinomial Naive Bayes (MNB) and Multivariate Bernoulli Naive Bayes (BNB) analyses performed on the Kaggle dataset. It was found that Multinomial NaveBayes outperforms Multivariate Bernoulli Nave Bayes on the provided dataset.On the training set, the Multinomial and Bernoulli Nave Bayes classifiers were produced using the following code.

```
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score


mnb = MultinomialNB()
bnb = BernoulliNB()

    mnb.fit(X2_train,Y2_train)
    Y2_pred2 = mnb.predict(X2_test)
    print(accuracy_score(Y2_test,Y2_pred2))
    print(confusion_matrix(Y2_test,Y2_pred2))
    print(precision_score(Y2_test,Y2_pred2))

0.9593810444874274
[[896   0]
 [ 42  96]]
1.0

    bnb.fit(X2_train,Y2_train)
    Y2_pred3 = bnb.predict(X2_test)
    print(accuracy_score(Y2_test,Y2_pred3))
    print(confusion_matrix(Y2_test,Y2_pred3))
    print(precision_score(Y2_test,Y2_pred3))

0.9700193423597679
[[893   3]
 [ 28 110]]
0.9734513274336283
```

fig. 1. Result of mnb and bnb

In the fig-3we can clearly see that MNB is more accurate as compare to BNB.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, No. 5, May : 2023

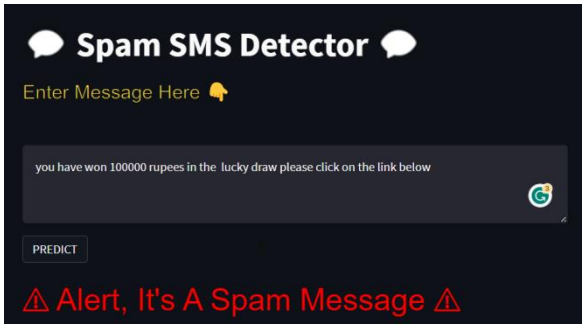

fig. 4. GUI of MSG spam detector

In the fig-4 we can see the GUL of our current project, in the square box we can past the MSG and check where the MSG is spam or Not.

Conclusion

This study found that Multinomial Nave Bayes performs slightly better than Bernoulli Nave Bayes on datasets with fewer records (5169 records in this instance), but it only achieves an accuracy of about 95 percent, which is inefficient. This is consistent with the fact that it is difficult to obtain high accuracies with less data, and that more data leads to higher accuracies with both algorithms discussed. The authors also conclude that, while Multinomial Nave Bayes provides greater accuracy, the difference in accuracies is not statistically significant because Bernoulli Nave Bayes also provides an accuracy of nearly 97 percent, implying that the performance of the two methods does not differ significantly.

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. NilamNur Amir Sjarif, N F MohdAzmi, SuriayatiChuprat, “SMS Spam Message Detection using Term Frequenct-Inverse Document Frequency and Random Forest Algorithm,” in The Fifth Information Systems International Conference 2019, Procedia Computer Science 161 (2019). 509515,ScienceDirect.
2. “A comparative study on text classification algorithms" by C. Zhang and L. Zhang (2015).
3. "SMS Spam Collection: A Public Set of SMS Labeled Messages" by J. Almeida et al. (2011)
4. "Survey on Data Preprocessing for Machine Learning" by M. Islam et al. (2020)
5. "A novel approach for SMS spam detection using deep learning" by S. Kaur and S. S. Kaur (2021)
6. "A systematic review of SMS spam detection techniques" by S. Al-Samarraie and A. H. Al-Saedi (2021)
7. "A comparative study of SMS spam filtering techniques using machine learning algorithms" by A. L. Mohammed and M. A. Hamad (2021)
8. "A review on SMS spam detection techniques: Recent trends and future directions" by M. A. Rahman and A. Rahman (2022)