**Sms spam Detection USING MACHINE LEARNING ALGORITHM**

**Submitted in partial fulfilment of the requirements for the award of B.Tech Degree in Computer Science and Engineering (Data Science)**

BY:-

**DEEPSANKAR BHATTACHARYA**

**PRIYANSHU MONDAL**

**SANTANU BERA**

**SUDIPTA DEY**

Last Updated :- Sep’2023

**Abstract :-** The number of people using mobile devices increasing day by day. SMS (short message service) is a text message service available in smartphones as well as basic phones. So, the traffic of SMS increased drastically. The spam messages also

increased.

The hackers try to send spam messages for their financial or business benefits like market growth, lottery ticket information, credit card information etc. So, spam classification has special attention. In this paper, we applied various machine learning for SMS spam detection. We used a dataset to train the machine learning and deep learning models like NLTK and NB. The SMS spam collection data set is used for testing the method. The dataset is split into two categories for training and testing the research.

Our experimental results have shown that our **MultiNomial Naïve Bayes** model outperforms previous models in spam detection with an accuracy of good.

**OBJECTIVE :-**

Prediction of SMS spam has been an important area of research for a long time. The goal is to apply different machine learning algorithms to SMS spam classification problem, compare their performance to gain insight and further explore the problem and design an application based on one of these algorithms that can filter SMS spams with high accuracy. The current work proposes a gamut of machine learning based predictive models for accurately predicting the sms spam movement.

**Problem Statement :-**

A number of major differences exist between spam-filtering in text messages and emails. Unlike emails, which have a variety of large datasets available, real databases for SMS spams are very limited. Additionally, due to the small length of text messages, the number of features that can be used for their classification is far smaller than the corresponding number in emails. Here, no header exists as well.

Additionally, text messages are full of abbreviations and have much less formal language that what one would expect from emails. All of these factors may result in serious degradation in performance of major email spam filtering algorithms applied to short text messages.

**External Search :-**

**Dataset :- https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset**

**Theoritical Background :-**

* **Python**
* **Machine Learning Algorithms**
* **NLTK**
* **Stopwords**
* **SciKit-learn**
* **HTML**
* **CSS**
* **Bootstrap**
* **Flask**

**Aim & Scope :-**

**1.** The Proposed mode is based on the study of sms text data and technical

indicators. Algorithm selects best free parameters combination for MNB to avoid

over-fitting and local minima problems and improve prediction accuracy.

**2.** Our dataset consists of one large text file in which each line corresponds to a text

message. Therefore, preprocessing of the data, extraction of features, and

tokenization of each message is required. After the feature extraction, an initial

analysis on the data is done using label encoder and then the models like naive

Bayes (NB) algorithm and NLTK are used on next steps are for prediction.

**3.** The two methods used to predict the spam messages that are Fundamental and

technical analyses.

**Methods :-**

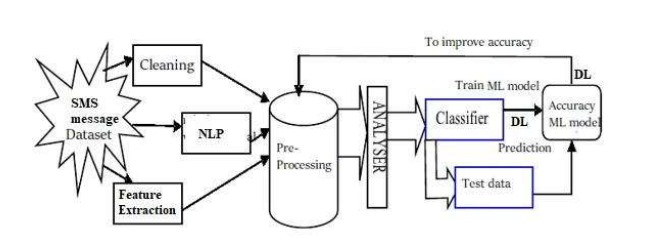
**MultiNomial Naïve Bayes :-**

Applying NB algorithm to the dataset using extracted features with different training set sizes. The performance in learning curve is evaluated by splitting the dataset into 80% training set and 20% test set. The NB algorithm shows good overall accuracy.

We notice that the length of the text message (number of characters used) is a very good feature for the classification of spams. Sorting features based on their mutual information (MI) criteria shows that this feature has the highest MI with target labels.

Additionally, going through the misclassified samples, we notice that text messages with length below a certain threshold are usually hams, yet because of the tokens corresponding to the alphabetic words or numeric strings in the message they might be classified as spams.

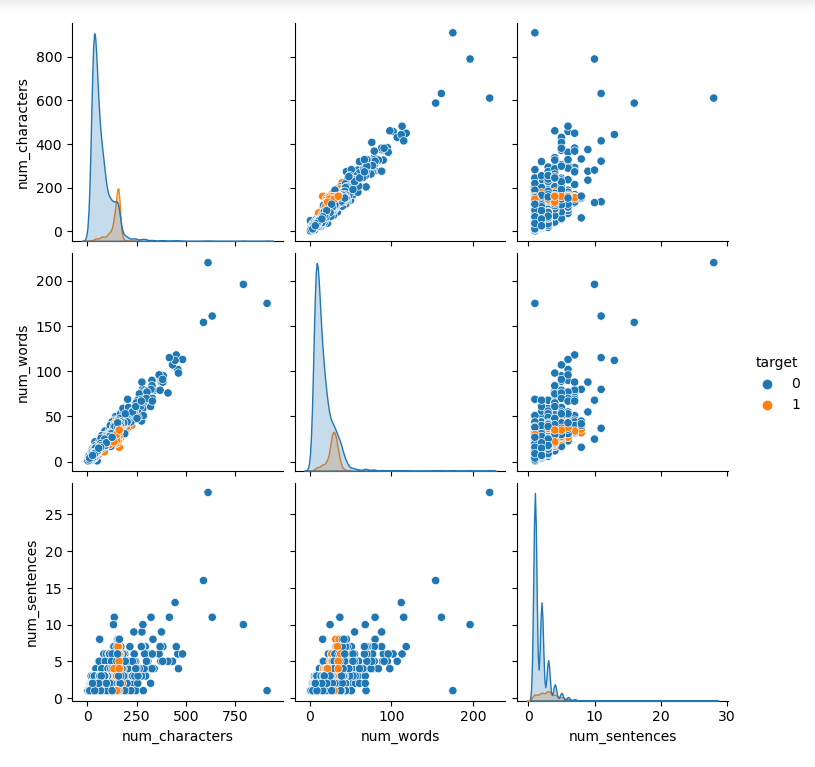
By looking at the learning curve, we see that once the NB is trained on features extracted, the training set error and test set error are close to each other. Therefore, we do not have a problem of high variance, and gathering more data may not result in much improvement in the performance of the learning algorithm. As the result, we should try reducing bias to improve this classifier. This means adding more meaningful features to the list of tokens can decrease the error rate, and is the option that is explored next.



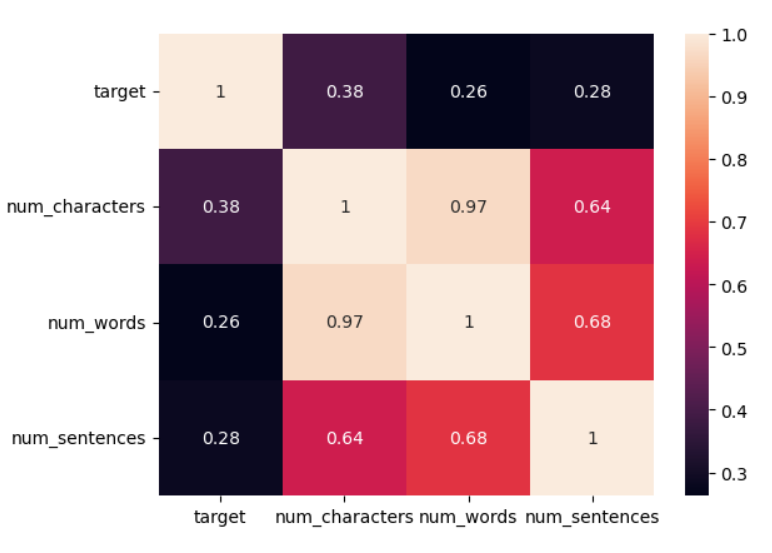
**Data Pre-Processing :-**

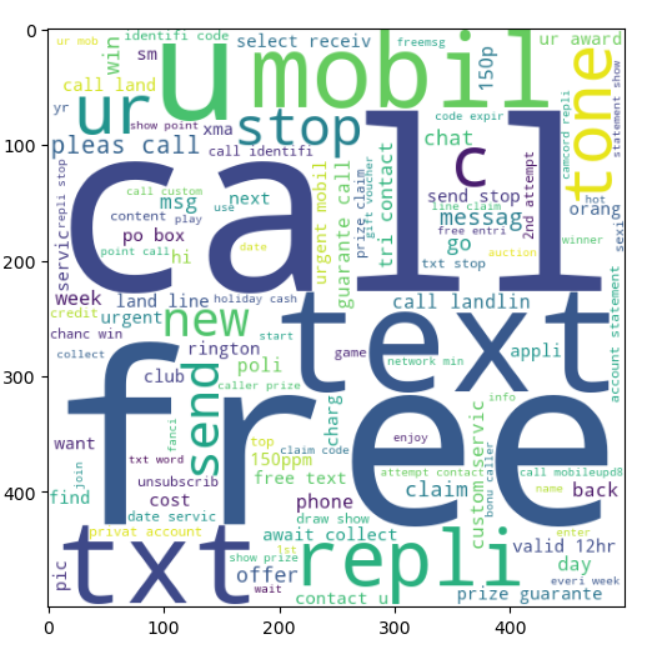
Hear the raw data in the spam dataset is cleaned and the metadata is appending to it by removing the things which are converted to the integer. So, the data is easy to train. In this pre-processing, we first load the metadata into this and then this metadata will be attached to the data and replace the converted data with metadata. Then this data will be moved further and remove the unwanted data in the list and it will divide the data into the train and the test data.

**Pairplot :-**

****

**Heatmap :-**

****

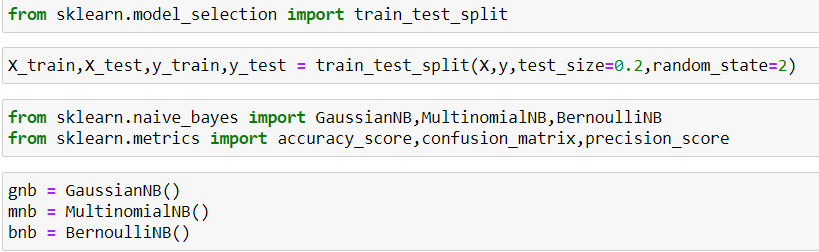
**WordCloud :- **

WordCloud for Spam

****

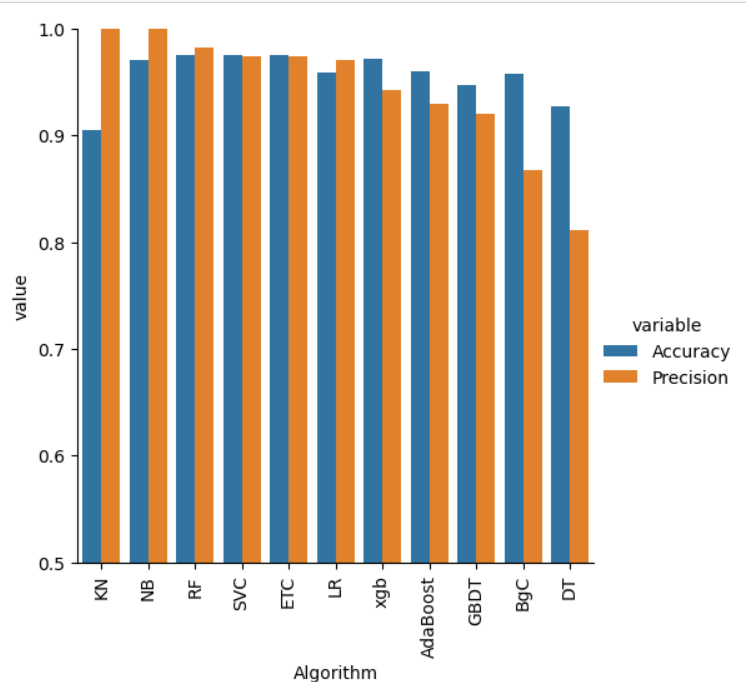
WordCloud for Ham

**Train - Test Split :-**

****

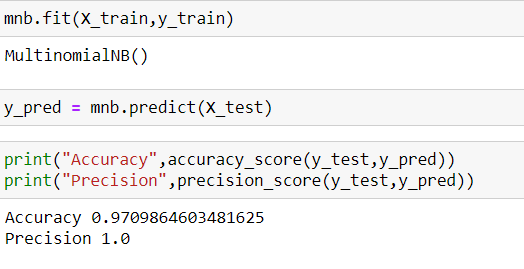
**Performance Comparison :-**

****

****

**Spam Prediction Module :-**

****

****

**Conclusion :-**

The SMS spam message problem is plaguing almost every country and keeps increasing without a sign of slowing down as the number of mobile users increase in addition to cheap rates of SMS services. Therefore, this paper presents the spam filtering technique using various machine learning algorithms. Based on the experiment, TF-IDF with Nave bayes classification algorithm outperforms good compare to other algorithm in terms of accuracy percentage. However, it is not enough to evaluate the performance based on the accuracy alone since the dataset is imbalanced. After some examinations, NB algorithm still manages to provide good precision and accuracy with 1.00 of precision while 0.9709 for accuracy. Different algorithms will provide different performances and results based on the features used. For future works, adding more features such as message lengths might help the classifiers to train data better and give better performance.

Thank You