International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 12 | Dec - 2021 ISSN: 2582-3930

# SMS Spam Detection using Machine Learning Classifiers

**Tanya Goel[1], Garima Arora[2]**

[1]*CSE Department, Maharaja Agrasen Institute of Technology*
[2]*CSE Department, Maharaja Agrasen Institute of Technology*

---

**Abstract -** The consistent development in the technology field has given rise to digitization. Short Message Service (SMS) has now become one of the most important forms of communication. SMS is different from other chatting-based messaging systems because it does not require an active internet connection for transferring the message. Due the this the use of SMS has increased to such a significant level that devices are sometimes flooded with a lot of spam SMS which can even lead to SMS attacks, which in turn may lead to theft of private and useful information. So, to identify the spam messages we have created a system that will predict whether a message is spam or ham i.e., whether it is a malicious message or not. We compared various algorithms like Naive Bayes, Random Forest, SVM, etc. to find the most efficient one for this classification and have used the TF- IDF vectorizer algorithm for creating a dictionary, which will include all the top words that a spam message possesses. The system will classify SMS as ham or spam after referring to this dictionary.

***Key Words*:** Classifier, Ham, SMS, Spam, TF-IDF vectorizer

## 1.INTRODUCTION

SMS is a technique of sending short messages from one device to another. SMS technology evolved out of the global system for mobile communications standards and almost everyone is using it for communication. Various organizations use the SMS service for communicating with their customers, even government organizations use SMS for communication. Thus, SMS is playing an important role in communication because it does not require an active internet connection for transferring the message. This wide usage of SMS attracts hackers and spammers. Spam is any kind of unrequired, unrequested digital communication that gets sent out in bulk format. Spam is usually sent out through emails, they can also be distributed through phone calls, text messages, or social media platforms. SMS Spam is unsolicited bulk messaging with some business interest. SMS spam is used for advertising commercials and for spreading links that carry out phishing. Most of the spam messages are typically longer than the ham messages and these spam messages show a clear pattern. Most spam messages ask the users to call a particular number, reply to the SMS, or visit a certain URL. This pattern can be concluded by the results obtained using a simple SQL query on the spam entity. The low pricing and the high bandwidth available to the SMS network have attracted a large amount of SMS spam in recent times.

SMS spam detection is a vital task in which spam SMS messages are identified. As the number of SMS messages that are communicated every day is increasing, it is becoming even more challenging for a user to remember and correlate the newer SMS messages received with reference to the SMS received previously.

In this paper, the aim is to train, test and compare different traditional machine learning classifiers on the dataset. The classifiers are evaluated on the basis of their accuracy and precision. Thus, using the knowledge of machine learning we have developed an SMS spam classifier.

## 2.LITERATURE SURVEY

According to "SMS Spam Filtering Using Supervised Machine Learning Algorithms" by Pavas Navaney, Gaurav Dubey, and Ajay Rana, the SVM algorithm gives the highest accuracy in terms of classifying ham and spam messages, followed by the naïve Bayes method, and then Maximum Entropy method. According to their research, SVM is best with an accuracy of 97.4% while Naive Bayes has an accuracy of only 95%.

So, we tried to improve the accuracy of Naive Bayes, and have successfully increased it to 97.09%.

## 3.METHODOLOGY

Data Collection: We have collected a dataset from Kaggle which is SMS Spam Collection Dataset (Collection of SMS messages tagged as spam or legitimate).

Data Cleaning: In this phase, we cleaned the data which will be used for experimentation. We removed the columns which had null values and also removed the duplicate values. The columns were renamed for a better understanding of the data.

Exploratory Data Analysis (EDA): In this phase, we have calculated the percentage of spam and ham messages in the dataset and represented them using a pie chart. The number of alphabets, words, and sentences used in a message counted and the histogram was plotted for better analysis.

Data Preprocessing: In this phase, we followed the steps listed below:

1. Converted messages to lowercase
2. Tokenization
3. Removed special characters
4. Removed stop words and punctuation
5. Stemming

After these steps word cloud was made for both ham and spam to see the top words used in them respectively.

Model Building: We converted the textual data to numerical data using the TF- IDF vectorizer algorithm, trained the model, and used various algorithms to find out the most efficient one for SMS spam classification.

Evaluation and Improvement: We improved the accuracy of the system by taking only the top 3000 words.

Prediction: In the last phase, we gave various text messages as input to check whether the message is spam or ham.

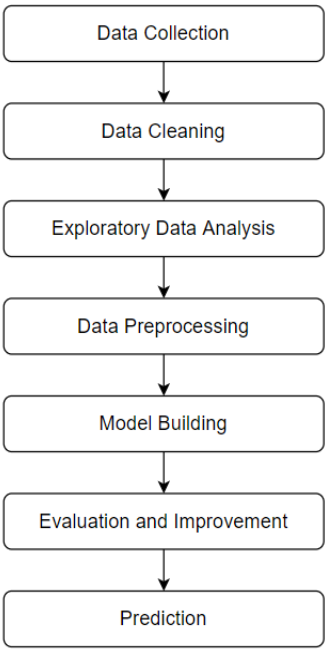


**Figure-1:** Workflow

## 4.ARCHITECTURE

For any new SMS the system will follow the steps mentioned below:

Text transformation: The incoming message is passed into the transform_text function, which does the following job:

1. Converting text to lowercase
2. Tokenization
3. Removing special characters
4. Removing stop words and punctuations
5. Stemming

Vectorization: As Naive Bayes requires a number as an input so we converted our text into a number or text to vector using the TF-IDF vectorizer algorithm.

Applying algorithm: In the last step, the Naive Bayes algorithm is applied to the text to predict whether the message is spam or not.

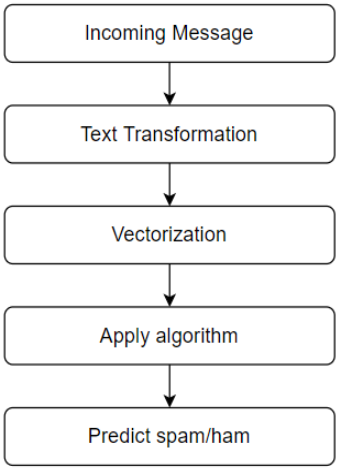


**Figure-2:** Architecture

## 5. EXPERIMENTATION

As a part of experimentation, we tested our system by providing various inputs in the “Enter the message box” of our system, to predict whether the message given as input is spam or not.

Two of the inputs given to the developed system are:

Input 1: Congratulations you have won 1000 INR. Call on this number to get your prize.

Input 2: Hi! Saw your presentation today, and was really impressed with the graphics that you used.

The outputs of the above-mentioned inputs are discussed in section 6.4 of the paper.

## 6.RESULTS AND DISCUSSION

6.1 Visualization of Dataset

In the data, 4516 are ham samples and 653 are spam samples which is 87.37% ham and 12.63% spam messages which are shown in figure 3 in the form of a pie chart.

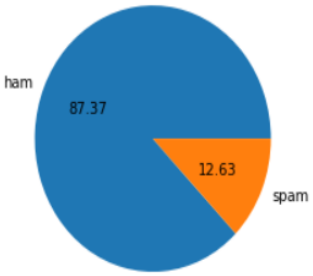

**Figure-3:** Percentage of spam and ham messages in the dataset

From figure 3, it is clear that our data is imbalanced so precision should be given priority over accuracy.

The count of the number of characters, words, and sentences in spam messages are described through the figure shown below:

| | num_characters | num_words | num_sentences |
|---|---|---|---|
| count | 653.000000 | 653.000000 | 653.000000 |
| mean | 137.891271 | 27.667688 | 2.969372 |
| std | 30.137753 | 7.008418 | 1.488910 |
| min | 13.000000 | 2.000000 | 1.000000 |
| 25% | 132.000000 | 25.000000 | 2.000000 |
| 50% | 149.000000 | 29.000000 | 3.000000 |
| 75% | 157.000000 | 32.000000 | 4.000000 |
| max | 224.000000 | 46.000000 | 9.000000 |

**Figure-4:** Spam message analysis

Similarly, the count of the number of characters, words, and sentences in ham messages are described through the figure shown below:

| | num_characters | num_words | num_sentences |
|---|---|---|---|
| count | 4516.000000 | 4516.000000 | 4516.000000 |
| mean | 70.459256 | 17.123339 | 1.815545 |
| std | 56.358207 | 13.491315 | 1.364098 |
| min | 2.000000 | 1.000000 | 1.000000 |
| 25% | 34.000000 | 8.000000 | 1.000000 |
| 50% | 52.000000 | 13.000000 | 1.000000 |
| 75% | 90.000000 | 22.000000 | 2.000000 |
| max | 910.000000 | 220.000000 | 38.000000 |

**Figure-5:** Ham message analysis

From figure 4 and figure 5, we can interpret that spam messages are typically longer than ham messages.

6.2 Visualization using Word Cloud

For better understanding and analysis, we found out the top words used in ham and spam messages by creating their word clouds which can be seen in figure 6 and figure 7 for spam and ham respectively.

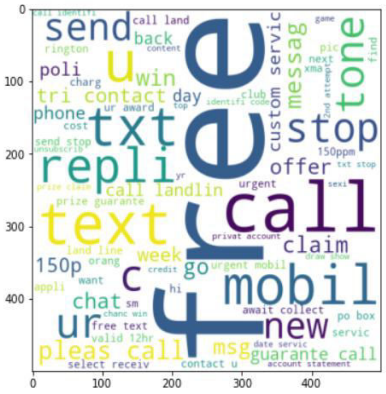


**Figure-6:** Spam Word Cloud


**Figure-7:** Ham Word Cloud

To see the top 30 words clearly in the messages we plotted a bar graph which can be seen in figure 8 and figure 9 for spam and ham respectively.

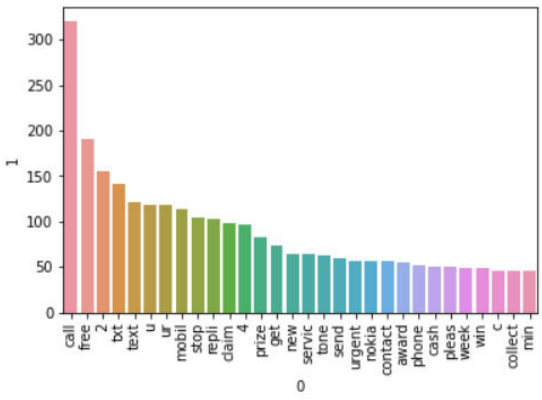


**Figure-8:** Spam Bar Graph

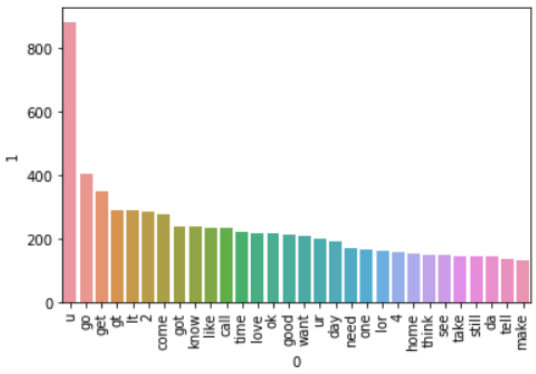


**Figure-9:** Ham Bar Graph

6.3 Classification Results of Classifiers

To classify the messages as ham and spam, in this paper, we used the classification algorithms such as K-nearest neighbor classifier, Naive Bayes (Multinomial) classifier, Random Forest classifier, Extra Trees classifier, Support Vector classifier, AdaBoost classifier, Logistic Regression classifier, XGB classifier, Gradient Boosting classifier, Bagging classifier and Decision Tree classifier with all the basic essential hyperparameters. The complete dataset was divided into two parts, one for training and one for testing. All the classifiers were trained using the training data, which was 80 % of the sample data, and was validated using the remaining 20% sample data.

Result when all words used:

| | Algorithm | Accuracy | Precision |
|---|---|---|---|
| 1 | KN | 0.900387 | 1.000000 |
| 2 | NB | 0.959381 | 1.000000 |
| 5 | RF | 0.971954 | 1.000000 |
| 8 | ETC | 0.972921 | 0.982456 |
| 0 | SVC | 0.972921 | 0.974138 |
| 6 | AdaBoost | 0.961315 | 0.945455 |
| 4 | LR | 0.951644 | 0.940000 |
| 10 | xgb | 0.970019 | 0.934959 |
| 9 | GBDT | 0.952611 | 0.923810 |
| 7 | BgC | 0.958414 | 0.862595 |
| 3 | DT | 0.935203 | 0.838095 |

**Figure-10:** Precision and accuracy when all words are used

All the results, when we used all the words, are recorded in the above figure number 10. According to the said table, the K-Nearest Neighbor classifier, Naive Bayes (Multinomial) classifier and Random Forest classifier obtained a precision of 100% individually, but had varying accuracies, i.e., 90.04%, 95.94% and 97.19% respectively. The classifiers Extra Trees classifier and Support Vector classifier achieved the same accuracy of 97.30%, but had different precisions, i.e., 98.25% and 97.41% respectively. Similarly, the AdaBoost classifier, Logistic Regression Classifier, XGB classifier, Gradient Boosting classifier, Bagging classifier and Decision Tree classifier obtained an accuracy of 96.13%, 95.16%, 97%, 95.26%, 95.84% and 93.52% respectively, and had a precision of 94.55%, 94%, 93.50%, 92.38%, 86.26% and 83.81% respectively.

As our sample data set was unbalanced and we wanted best precision and better accuracy for all our classifiers. Hence after various experimentation, we selected the top 3000 words and performed our analysis on the selected words.

Result when top 3000 words were used:

| | Algorithm | Accuracy | Precision | Accuracy_num_max_3000 | Precision_num_max_3000 |
|---|---|---|---|---|---|
| 0 | KN | 0.900387 | 1.000000 | 0.905222 | 1.000000 |
| 1 | NB | 0.959381 | 1.000000 | 0.970986 | 1.000000 |
| 2 | RF | 0.971954 | 1.000000 | 0.975822 | 0.982906 |
| 3 | ETC | 0.972921 | 0.982456 | 0.974855 | 0.974576 |
| 4 | SVC | 0.972921 | 0.974138 | 0.975822 | 0.974790 |
| 5 | AdaBoost | 0.961315 | 0.945455 | 0.960348 | 0.929204 |
| 6 | LR | 0.951644 | 0.940000 | 0.958414 | 0.970297 |
| 7 | xgb | 0.970019 | 0.934959 | 0.967118 | 0.933333 |
| 8 | GBDT | 0.952611 | 0.923810 | 0.946809 | 0.919192 |
| 9 | BgC | 0.958414 | 0.862595 | 0.958414 | 0.868217 |
| 10 | DT | 0.935203 | 0.838095 | 0.929400 | 0.828283 |

**Figure-11:** Precision and accuracy when top 3000 words are used

All the results, when we used the top 3000 words only, are recorded in the above figure number 11. According to the said table, the K-Nearest Neighbor classifier and Naive Bayes (Multinomial) classifier obtained a precision of 100% individually, but had varying accuracies, i.e., 90.52% and 97.10% respectively. The classifiers Random Forest classifier and Support Vector classifier achieved the same accuracy of 97.59%, but had different precisions, i.e., 98.29% and 97.47% respectively. Similarly, the Extra Trees classifier, AdaBoost classifier, Logistic Regression Classifier, XGB classifier, Gradient Boosting classifier, Bagging classifier and Decision Tree classifier obtained an accuracy of 97.48%, 96.03%, 95.84%, 96.71%, 94.68%, 95.84% and 92.94% respectively, and had a precision of 97.46%, 92.92%, 97.03%, 93.34%, 91.92%, 86.82% and 82.83% respectively.

Thus, the Naive Bayes (Multinomial) classifier is the best classifier, for data sets such as ours, for classifying SMS as ham and spam successfully. It gave an accuracy of 97.10% and had precision of 100%.

6.4 Prediction outputs:

For testing our system, we gave various text messages as input and predicted whether they are being detected as spam or not spam correctly or not. Two of them are shown in figure 12 and figure 13 below:

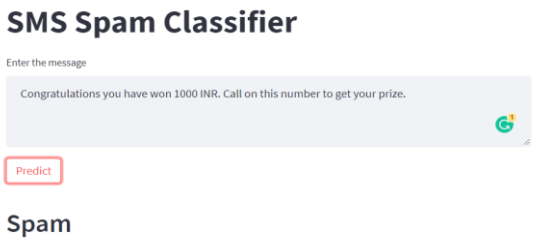


**Figure-12:** Message detected as spam


**Figure-13:** Message detected as not spam

## 7.CONCLUSION

Various machine learning algorithms were used to classify the text message as "spam" or "not spam" and comparisons were made for finding the best accuracy algorithm. It was found that the best result was given by the Multinomial Naive Bayes classifier with an accuracy of over 97% and a precision of 100%. This research paper provides an overview of using different techniques to predict whether a message is spam or ham.

## 8.FUTURE SCOPE

To improve the results furthermore, we can explore and apply deep learning models, LSTM and Bi-LSTM.

## ACKNOWLEDGEMENT

This work would not have been possible without the support of the Computer Science and Engineering Department, Maharaja Agrasen Institute of Technology. We are especially indebted to Prof. (Ms.) Namita Gupta, Head of Computer Science and Engineering Department, Maharaja Agrasen Institute of Technology, who has always encouraged us to pursue our academic and career goals.

## REFERENCES

1. Jeff Brown, William J Shipman, and Ron Vetter, "SMS: The Short Message Service," in Computer, vol. 40, no. 12, pp. 106–110, Dec. 2007.

2. Brian Whitworth and Elizabeth Whitworth, "Spam and the social-technical gap," Computer, vol. 37, pp. 38–45, 2004.

3. Hedieh Sajedi, Golazin Zarghami Parast, and Fatemeh Akbari, "SMS Spam Filtering Using Machine Learning Techniques: A Survey". Machine Learning Research. Vol. 1, No. 1, 2016, pp. 1–14. DOI: 10.11648/j.mlr.20160101.11

4. Yoon J, Kim H and Huh J. "Hybrid spam filtering for mobile communication", Journal of Computers and Security 2010; 29(4):446–459. DOI:10.1016/j.cose.2005.12.003.

5. Joe Inwhee and Shim Hyetaek "An SMS spam filtering system using support vector machine", In Proceedings of Future Generation Information Technology, Dec. 2010; 577–584, DOI: 10.1007/978-3-642-17569-5_56.

6. Hands-On Guide to Detecting SMS Spam Using Natural Language Processing from https://analyticsindiamag.com/hands-on-guide-to-detecting-sms-spam-using-natural-language-processing/

7. Pavas Navaney, Gaurav Dubey and Ajay Rana, "SMS Spam Filtering Using Supervised Machine Learning Algorithms", 2018 8th International Conference on Cloud Computing, Data Science & Engineering, Noida, 2018, pp. 43-48. Doi: 10.1109/CONFLUENCE.2018.8442564