SMS Spam Detection Using Machine Learning Approach

Anikait Kapoor, Debavushan Saikia, Ishaan Dhawan
Department of Computer Science and Engineering,
Apex Institute of Technology,
Chandigarh University, Mohali, Punjab, India

DOI:10.37648/ijrst.v14i01.002

[1]*Received: 18 November 2023; Accepted: 29 January 2024; Published: 07 February 2024*

ABSTRACT

With the rise in mobile awareness in recent years, the short message service (SMS) industry has generated billions of dollars in revenue. However, this has led to an increase in unwanted commercial advertising or spam sent to regular phones, with parts of Asia having up to 30% of content messages as spam in 2012. One of the challenges in SMS spam filtering, it requires a comprehensive database and the limited usefulness and dialect used in SMS. In this extension, analysts used a real SMS spam database from the UCI Machine Learning store and connected different machine learning methods after preprocessing and extracting markup. The results were compared and the main spam filtering algorithms for the message body were distinguished. The final reconstruction using 10-fold cross-validation appeared to have the primary classifier more than halve the overall error rate compared to the best proof in a paper.

Keywords: *supervised learning; classification algorithms; feature engineering; natural language processing (NLP); text classification.*

INTRODUCTION

In recent years, the mobile phone market has experienced significant growth. In the second quarter of 2013, a total of 432.1 million mobile phones were shipped, an increase of 6.0% year-on-year [1]. Just as the use of mobile devices and phones has become ubiquitous, the short message service (SMS) has become a multi-billion dollar advertising industry [2]. SMS is a text messaging platform that allows mobile phone users to exchange short text messages, typically under 160 7-bit characters. It is one of the most widely used data applications, with about 3.5 billion active users, representing about 80% of all mobile phone subscriptions at the end of 2010. As its popularity increases, the number of unwanted advertisements sent to mobile phones via SMS also increases accordingly. Although SMS spam is not yet as widespread as email spam, it accounted for about 90% of emails in 2010 and currently contributes less than 1% of SMS exchanges in North America as of December 2012. However, due to the increasing popularity of SMS change among young people. With the population and costs of texting falling over the years, SMS spam is on the rise, with some parts of Asia reporting as much as 30% of text messages as spam in 2012. In the Middle East, some carriers even send marketing messages themselves, while in some countries SMS spam is more troublesome than email spam as it can also increase the cost to the recipient.

Spam sifting in content messages varies from e-mail sifting in a few ways. Firstly, whereas there are numerous huge datasets accessible for mail spam, the accessibility of genuine databases for SMS spam is exceptionally restricted. Furthermore, due to the shorter length of content messages, the number of highlights that can be utilized for the classifier is much littler than the comparing number in e-mail. There's too no subject line display in SMS messages. Moreover, SMS content messages tend to be full of shortened forms and have less formal dialect than e-mail, which can lead to noteworthy execution debasement of standard spam sifting calculations when connected to brief content messages.

[1] How to cite the article: Kapoor A., Saikia D., Dhawan I.; February 2024; SMS Spam Detection Using Machine Learning Approach; *International Journal of Research in Science and Technology*, Vol 14, Issue 1, 10-17, DOI: http://doi.org/10.37648/ijrst.v14i01.002

The goal of this extension is to use different machine learning calculations to handle the SMS spam classification problem and compare their implementation for better distance detection, much better, better, better, better understanding of the problem. . In addition, the extension says planning an application can filter spam correctly based on one of these calculations. To do this, a database of 5574 UCI Machine logic information file content messages compiled in 2012 is used. This dataset consists of a subset of 425 spam SMS messages extracted directly from a Object content website, a subset of 3,375 randomly selected non-spam messages from US SMS Corpus (NSC).), list of 450 non-spam SMS messages by Caroline Tag's Proposal PhD and available audience SMS Spam Corpus v.0.1 Very large (1,002 non-spam and 322 spam) Per row of file content record huge data starts with a type message, followed by a string of content messages. After pre-processing and extracting the inclusions, different machine learning strategies such as Naive Bayes, SVM and others are connected to the tests and their performance is compared. Finally, the implementation of the main classifier in this range is compared with the performance of the classifiers used in a paper also using this dataset. The information survey was performed using MATLAB and the machine learning calculations were performed in Python using the scikit-learning library.

PROJECT SCOPE

1. Recommended mode based on research on text messages and technical data Indicators. Algorithm to choose the best combination of free parameters for LSTM to avoid local minima and overfit problems and improve the accuracy of predictions.
2. Our dataset is in the form of a single text file where each line represents a text message. To prepare the data for analysis, we need to perform pre-processing steps such as feature extraction and tokenization of each message. Once the features are extracted, we first use a label encoder for data analysis, followed by applying machine learning algorithms such as Naive Bayes and LSTM for prediction.
3. Two methods used to predict spam messages are basic and technical analysis.

HIGHLIGHT EXTRACTION & INTRODUCTORY INVESTIGATION

As previously stated, our dataset is in the form of a large text file where each line represents a text message. As a result, we need to pre-process the data, extract relevant features, and tokenize each message. Following feature extraction, we conduct an initial data analysis using the Naive Bayesian (NB) algorithm with a polynomial event model and Smooth Laplace. Based on the results obtained, we will determine the next steps in the process.

To examine the beginning information, each message within the dataset is separated into tokens comprising of alphabetic characters. Extraordinary characters such as spaces, commas, and periods are expelled, and string literals are put away as tokens as long as they contain no non-alphabetic characters. Misspellings and truncations within the messages are not taken under consideration, and no calculation is utilized to explore for words. Furthermore, three other tokens are created based on the number of dollar signs, the number of numeric strings, and the whole number of characters within the message. The inspiration behind counting message length as a include is that marketers point to use as much space as conceivable without surpassing the 160 character restrain for SMS. Within the introductory information examination, a polynomial occasion demonstrate with Laplace smoothing was utilized. Extricating tokens from all messages within the dataset come about in 7,789 highlights, in spite of the fact that not all of them were valuable for classification. Tokens that happened less than five times or more than 500 times were evacuated, as they were either as well uncommon or as well common to contribute to the message substance. These edges were chosen by testing the execution of the NB classification calculation on diverse values. As a result, 1,552 highlights were produced from the remaining tokens.

The execution of the NB calculation was assessed by utilizing distinctive preparing set sizes and partitioning the dataset into 70% for preparing and 30% for testing. The comes about of applying the NB calculation to the dataset utilizing extricated highlights were plotted in Figure 1. It was watched that the NB calculation accomplished great by and large precision. The ten-fold cross-validation for this calculation on the given information set appeared 1.5% blunders by and large, with 93% spam messages blocked (SC) and 0.74 ham messages blocked (BH).

$$SC = \frac{\text{FN Cases (False Negative)}}{\text{No of Spams}}$$

$$BH = \frac{\text{FP Cases (False Positive)}}{\text{No of Hams}}$$

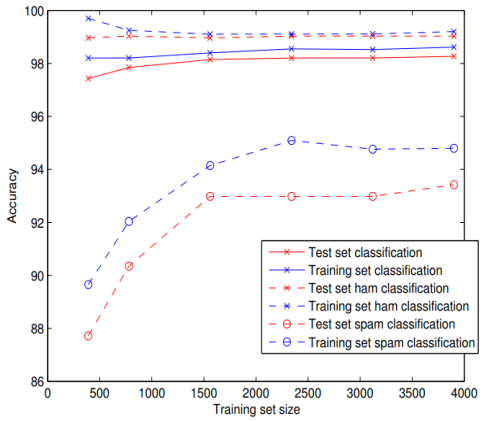

Fig. 1. Learning curve for naive Bayes algorithm applied to the dataset and evaluated using cross validation (30% of initial dataset is our test set

Based on the comes about of our examination, we have watched that the length of the message (i.e., the number of characters utilized) can be a exceedingly enlightening include for spam classification. By sorting highlights based on their common data (MI) criteria, it shows up that this highlight has the most noteworthy MI with target names. Moreover, upon dissecting the misclassified tests, we have taken note that messages with a length underneath a certain limit are as a rule genuine messages (i.e., hams), but they may get misclassified as spam due to certain tokens comparing to alphabetic words or numeric strings within the message. When looking at the learning bend, able to see that once the NB calculation is prepared on the extricated highlights, the preparing mistake and test blunder are moderately near to each other. Based on the examination of the comes about, it was watched that the length of the message substance (number of characters utilized) can be an critical highlight for spam classification. By sorting the highlights based on their shared data (MI) criteria, it was found that this highlight has the most noteworthy MI with target names. Besides, upon analysing the misclassified tests, it was found that messages with a length underneath a certain limit are as a rule ham, but due to the tokens comparing to alphabetic words or numeric strings within the message, they can be classified as spam. Looking at the learning bend, it was watched that after preparing the NB calculation on the extricated highlights, the preparing set blunder and test set blunder are near to each other, showing that there's no issue of overfitting and gathering more data may not result in much enhancement within the algorithm's execution. Therefore, to diminish predisposition and progress the classifier, more pertinent highlights got to be included to the list of tokens.

To achieve this, we included five unmistakable banners to gather the tokens. These banners decide whether the message length in characters is ≤ 40, ≤ 60, ≤ 80, ≤ 120, and ≤ 160. Furthermore, we included the string of non-alphabetic characters and images, barring dab, comma, colon, and shout check, to our tokens. For occasion, a string of characters such as "://" may recommend the nearness of a web address, or a character such as "@" may show the nearness of a mail address within the message. The coming about highlights is sifted once more if they are as well uncommon or as well common within the dataset. At long last, we conclusion up with a list of 1582 highlights.

K-NEAREST NEIGHBORS (KNN)

The KNN algorithm is utilized to classify new data by analysing the k nearest neighbors. In this paper, a value of k=6 is used. The distances from neighbors can be calculated using different metrics, such as Euclidean or Manhattan distance. The class of the new data can be determined by considering the majority of the neighbouring classes or by using calculated distances. Unlike generalizing techniques, such as regression or decision trees, KNN is a non-generalizing technique because it stores all the training data in memory. This can lead to memory issues when handling large datasets, but fast index structures like sphere trees and KD trees can be used to address this problem.

NAIVE BAYES

In this section, we applied the NB algorithm to the final set of extracted features. The high accuracy, simplicity, and speed of this algorithm make it a desirable choice for spam detection problems. In the context of the NB algorithm with the polynomial event model, we assumed that the length of the message corresponds to the hypothesis of an independent Bernoulli variable, with each character of a text message being either spam or ham. We applied the NB algorithm with the polynomial event model and Laplace smoothing to the dataset, and using a 10-fold cross-validation, we achieved an overall error of 1.12%, 94.5% SC, and 0.51% BH. Using Bayesian a priori data and applying the NB algorithm with the same event type reduced SC (93.7%) and BH (0.44%) with a small margin, but the overall error remained the same. It is expected that the Bayesian model will improve the algorithm in cases of high variance. The learning curve for the polynomial NB applied to the final extracted features is shown in Figure 2. The errors for different datasets in this graph were generated using cross-validation with a 70% sample as the training set. As shown in the figure, the test set error and the training set error are close to each other and within an acceptable range, indicating that there is no overfitting in the model. To reduce bias and improve the accuracy of the algorithm, more complex models will be explored in the following sections.

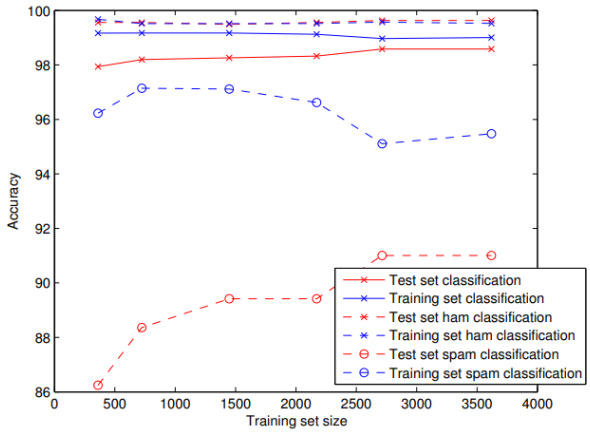

Fig. 2. Learning curve for multinomial NB algorithm applied to final features

USED LIBRARIES

1. **Numpy:** NumPy is a Python package/library that offers support for big, multi-dimensional arrays and matrices, as well as an extensive array of mathematical functions to manipulate these arrays. It is a crucial library for scientific computation in Python and is commonly utilized with other libraries for tasks such as data analysis, machine learning, and numerical computing.
2. **Sklearn:** Sklearn is an open-source machine learning library intended for Python programming language that provides a range of classification, regression, and clustering algorithms. These algorithms include popular methods like support vector machines, random forests, gradient boosting, k-means, and DBSCAN. The library is designed to work smoothly with other Python libraries used for numerical and scientific computation, such as NumPy and SciPy.
3. **Multinomial Naive Bayes:** The Multinomial Naive Bayes classifier is a good choice for classification problems that involve discrete features such as word counts used for text classification. Typically, the Multinomial distribution requires integer feature counts. However, in practice, fractional counts like tf-idf may also work well.
4. **Streamlit:** Streamlit is a Python-based open-source framework for building web applications. It allows developers to create and deploy data-driven apps quickly and easily by using Python scripts.

SYSTEM SPECIFICATIONS

1. HARDWARE REQUIRMENTS :

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented

- **Processor:** Intel I5
- **Ram:** 4GB
- **Hard Disk:** 40 GB

2. Software Requirements:

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide abasis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team‘s and tracking the team’s progress throughout the development activity.

- **Python IDE:** Anaconda Jupyter Notebook
- **Programming Language:** Python

SYSTEM ARCHITECTURE

- Data Collection: This component is responsible for collecting SMS messages data for both spam and non-spam messages. The dataset should be well-balanced and representative of the real-world scenario.
- Data Preprocessing: This component is responsible for cleaning and processing the collected data to remove irrelevant information and normalize the data for machine learning algorithms. This process may involve various stages such as breaking down the input data into tokens, removing commonly used words (stop words), reducing words to their base form (stemming), and selecting relevant features from the data.
- Feature Engineering: This component is responsible for selecting and creating the most relevant features from the preprocessed data to be used as input for machine learning algorithms. Some common features used in SMS spam detection include the length of the message, the frequency of specific words or phrases, and the presence of certain characters or symbols.
- Model Selection: This component is responsible for selecting the appropriate machine learning algorithm to classify SMS messages as spam or non-spam. Some common algorithms used for SMS spam detection include logistic regression, support vector machines, and Naive Bayes.
- Model Training: This component is responsible for training the selected machine learning algorithm using the preprocessed data and the selected features.
- Model Evaluation: This component is responsible for evaluating the performance of the trained machine learning algorithm using metrics such as accuracy, precision, recall, and F1 score.
- Deployment: This component is responsible for deploying the trained machine learning algorithm into a real-world SMS spam detection system. This may involve integrating the model with a mobile application or a messaging platform.

PROPOSED SYSTEM

The dataset is subjected to the NB algorithm using features obtained from a different training set size. The learning curve performance is evaluated by splitting the dataset into a 70D44 training set and a 30% test set. The algorithm demonstrates a high overall accuracy. Notably, the length of the SMS (i.e., the number of characters used) is a valuable feature for spam classification. This feature has the highest mutual information (MI) for the target labels when the features are sorted based on their MI. Other pre-processing steps, such as tokenization, stop word removal, stemming,

and feature extraction, can also be performed. Furthermore, upon reviewing misclassified samples, it was observed that text messages with a length below a certain threshold are often labelled as ham, but due to the presence of tokens containing alphabetic characters or numerical strings in the message, they may be misclassified as spam. The learning curve analysis reveals that after training the algorithm on the extracted features, the error rates for the training set and the test set are similar. Based on the current learning curve, it appears that high variance is not an issue and collecting more data may not significantly improve the performance of the learning algorithm. Instead, we should focus on reducing bias to improve the classifier. One way to achieve this is by exploring additional tokenization features that may lead to more accurate classifications.

Advantages of Proposed System

- Less complex than the previous process
- Ability to learn and extract complex functionality.
- Good accuracy
- With simplicity and fast processing time, the proposed algorithm gives better results execution time.
- The implementation of machine learning and deep learning techniques has shown to be effective in predicting values.
- Correct prediction

RESEARCH OBJECTIVE

1. To develop a more accurate and effective SMS spam detection system: The aim is to enhance the precision and speed of current SMS spam identification systems by utilizing machine learning methodologies. This could involve exploring different algorithms, feature engineering techniques, and data pre-processing methods to develop a more robust and accurate system.

2. The aim of this objective is to assess the performance of various machine learning algorithms in detecting SMS spam. The focus is on comparing the accuracy, precision, recall, and F1 score of algorithms such as decision trees, logistic regression, and support vector machines to determine which one is the most effective.

3. To investigate the impact of different features on SMS spam detection performance: The objective here would be to explore how different features (such as message length, number of URLs, and use of specific keywords) impact the performance of SMS spam detection algorithms. This could involve conducting feature selection and feature engineering experiments to identify the most informative features.

4. To develop a real-time SMS spam detection system: The objective here would be to develop a machine learning-based SMS spam detection system that can operate in real-time. This could involve exploring different machine learning models that are computationally efficient and can quickly classify incoming messages as spam or not spam.

FURTHER ENHANCEMENTS

- Ensemble Methods: Ensemble methods can improve the performance of machine learning models by combining multiple models. This can include methods like bagging, boosting, and stacking, which can help to reduce overfitting and increase accuracy.

- Deep Learning: The utilization of deep learning methods like neural networks can enhance the precision of SMS spam detection. This strategy has proven to be successful in various natural language processing assignments, involving the classification of text.

- Transfer Learning: Transfer learning can be applied in SMS spam detection by using pre-trained models. For instance, pre-trained language models such as BERT (Bidirectional Encoder Representations from Transformers) can be fine-tuned for SMS spam detection, which can result in improved accuracy while reducing the amount of required training.

- Active Learning: Active learning is a method that can enhance the accuracy of SMS spam detection models by selectively choosing which samples to annotate during the training process. By doing so, it is possible to reduce the amount of labelled data required while still improving the model's accuracy.

- Multilingual SMS Spam Detection: SMS spam detection can be enhanced to support multiple languages. This can involve developing models for specific languages or using multilingual models that can handle different languages.

- Real-Time Processing: Real-time processing can be used to classify SMS messages in real-time, as they are received. This can be important for systems that need to make fast decisions, such as blocking spam messages from reaching users.

CONCLUSION

The table presented in the study shows the results of applying various classification models to the SMS Spam dataset. The authors found that the Bayesian Naive Polynomial with Laplace smoothing and Linear Multiplier SVM are among the most effective classifiers for SMS spam detection. The SVM classifier, which was the best in the original article cited by the authors, achieved an overall accuracy of 97.64%. The next best classifier in their study was the Naive Bayes classifier, which achieved an overall precision of 97.50%. Compared to previous work, the authors' classifier reduced the overall error by more than half. This improvement was attributed to the addition of important features such as message length, setting a threshold for the length, and analysing learning curves and misclassified data.

REFERENCES

1. Press Release, Growth Accelerates in the Worldwide Mobile Phone and Smartphone Markets in the Second Quarter, According to IDC, http: //www.idc.com/getdoc.jsp?containerId=prUS24239313
2. Tiago A. Almeida, Jos Mara G. Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of SMS spam filtering: new collection and results. In Proceedings of the 11th ACM symposium on Document engineering (DocEng '11). ACM, New York, NY, USA, 259-262. DOI=10.1145/2034691.2034742 http://doi.acm.org/10.1145/2034691.2034742
3. http://en.wikipedia.org/wiki/Short Message Service
4. http://en.wikipedia.org/wiki/Mobile phone spam
5. Adaboost, http://en.wikipedia.org/wiki/AdaBoost
6. SMS Spam Collection Data Set from UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection
7. Scikit-learn Ensemble Documentation, http://scikit-learn.org/stable/ modules/ensemble.html
8. T. G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, Multiple Classifier Systems, pages 1-15. LNCS Vol. 1857, Springer, 2001.
9. SMS Spam Collection v.1, http://www.dt.fee.unicamp.br/~tiago/ smsspamcollection

AUTHORS' BIOGRAPHY

Anikait Kapoor is a passionate student from Navi Mumbai, Maharashtra, currently pursuing a Bachelor's degree in Computer Science and Engineering with a specialization in Artificial Intelligence and Machine Learning at Apeejay School Nerul. With a focus on AI, deep learning, machine learning, and neural networks, he has embarked on various projects aimed at tackling real-world challenges. His commitment to innovation and problem-solving drives him to explore cutting-edge technologies in the field. As an aspiring engineer, he is dedicated to mastering AI and contributing to its advancements. With a solid foundation in problem-solving and a thirst for knowledge, Anikait Kapoor is poised to make significant strides in the domain of artificial intelligence, shaping a future where technology serves humanity in transformative ways.

Debavushan Saikia, a native of Assam, embarked on his academic journey at A-New High School, followed by his 12th-grade education at Luit Valley Academy. Currently pursuing his Undergraduate degree in Computer Science with a specialization in Artificial Intelligence and Machine Learning at Chandigarh University, Debavushan is dedicated to advancing his expertise in cutting-edge technologies. With a passion for the ever-evolving field of AI and ML, he is poised to make significant contributions to the realm of technology. Debavushan Saikia's academic pursuits reflect his commitment to innovation and excellence in the dynamic world of computer science.

Ishaan Dhawan is a budding scholar in machine learning, currently pursuing a Bachelor's in Engineering in Computer Science with a specialization in Machine Learning from Chandigarh University. He' has passed his 12th from Guru Nanak Khalsa Senior Secondary School.

Ishaan's research focuses on machine learning applications, particularly in pattern recognition and predictive modelling. As a co-author of a published research paper, he contributed to data preprocessing and algorithm development, playing a key role in achieving significant outcomes.

Passionate about using technology for social good, Ishaan is dedicated to advancing machine learning research and innovation.