SMS Spam Detection

[1]T.Sowmya Krishna, [2]P.Jyothsna, [3]P.Pravallika

[4]T.Pavani

[1,2,3,4] UG Student, Department of Computer Science Engineering,

Dr K V Subba Reddy College Of Engineering For Women, Kurnool, Andhra Pradesh, India

Article Info

Page Number: 470-475

Publication Issue:

Vol. 71 No. 2 (2022)

Abstract

The number of people who use mobile devices is on the rise. SMS, or short message service, is a text messaging service that is available on both basic phones and smartphones. As a result, SMS traffic dramatically increased. The number of spam messages also went up. Junk or unwanted messages, also known as spam, can be used to harm users by wasting their time and stealing valuable information. Spammers attempt to send spam messages for financial or business reasons, such as market growth, information about lottery tickets, credit card information, fraudulent messages or advertisements, etc. Effective spam detection is a crucial method for determining whether an SMS is spam or not. Special attention should be paid to spam classification. For SMS spam detection, we used a variety of machine learning methods in this paper. Using our dataset, we develop a model for detecting spam. The Multinomial Nave Bayes model outperforms previous models in spam detection with an accuracy of 96.8%, as demonstrated by our experiments. For all of our implementations, we used Python

Article History

Article Received: *25 December 2021*

Revised: *20 January 2022*

Accepted: *24 February 2022*

Publication: *28 March 2022*

1. Introduction

As mobile phones and mobile networks gain in popularity, the Short Message Service (SMS) has become a popular means of communication. However, SMS spam is also affecting SMS users. The term "sms spam," which also goes by the name "drunk message," refers to any irrelevant messages that are sent through mobile networks. We have noticed a rise in the number of unsolicited commercial messages sent to mobile phones via text messaging as the platform's popularity has grown.

Spam messages are increasingly popular for a number of reasons. First, there are a lot of people who use mobile phones around the world, so there are a lot of people who could be targeted by spam messages. Second, the low cost of sending spam messages could be advantageous to the spammer. Last but not least, the majority of mobile phone spam classifiers are unable to accurately and efficiently identify spam messages because they lack sufficient computational resources.

One of the most talked-about subjects of recent decades is machine learning, and numerous classification applications based on machine learning can be found in a variety of fields of study. Particularly, spam detection is a relatively mature area of research that has several tried-and-true approaches. The dataset is a substantial text file in which the message's label and text message string begin each line. Machine learning techniques like Naive Bayes, SVM, Decision Tree, and others are applied to the samples after the data has been pre-

processed and features have been extracted, and their results are compared. The project's best classifier's performance is then determined. Then, using sklearn li in Python, various machine learning algorithms are applied library

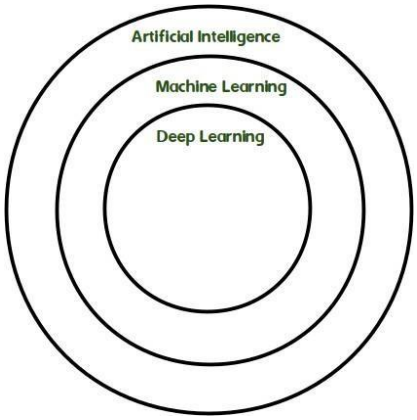

Fig.1 Supervised Machine Learning

2. Literature Review

Machine learning is undoubtedly, one of the most exciting subsets of Artificial intelligence. It completes the task of learning from data with specific inputs to the machine. It's important to understand what makes machine learning work and thus, how it can be used in the future.

The machine learning process, starts with inputting training data into the selected algorithm and the Training data being known or unknown data to develop the final machine learning algorithm,. The type of training data input does impact the algorithm, and that concept will be covered further momentarily.

Now input data is fed into the machine learning algorithm to test whether the algorithm works correctly. The prediction and results are then checked against each other. If t prediction and results don't match, the algorithm is re-trained multiple times until the data scientist gets the desired outcome. This enables the machine learning algorithm to continually learn on its own and produce the optimal answer, gradually increasing the accuracy over time

Classification models classify the input data. Classification techniques predict discrete responses. For example, the email is genuine, or spam. Typical applications include medical imaging, speech recognition and credit scoring.

For example, they can help to predict whether an online customer will purchase a product. Output can be yes or no: buyer or no buyer. But the methods of classification are not limited to two classes.

For example, a classification method can help assess whether a given image contains a car or a truck.

The simplest classification algorithm is logistic regression, which sounds like a regression method, but it is not. Logistic regression estimates the probability of occurrence of an event based on one or more inputs. The chart below shows the marks of past students and whether

they were admitted. Logistic regression allows us to draw a line that represents the decision boundary

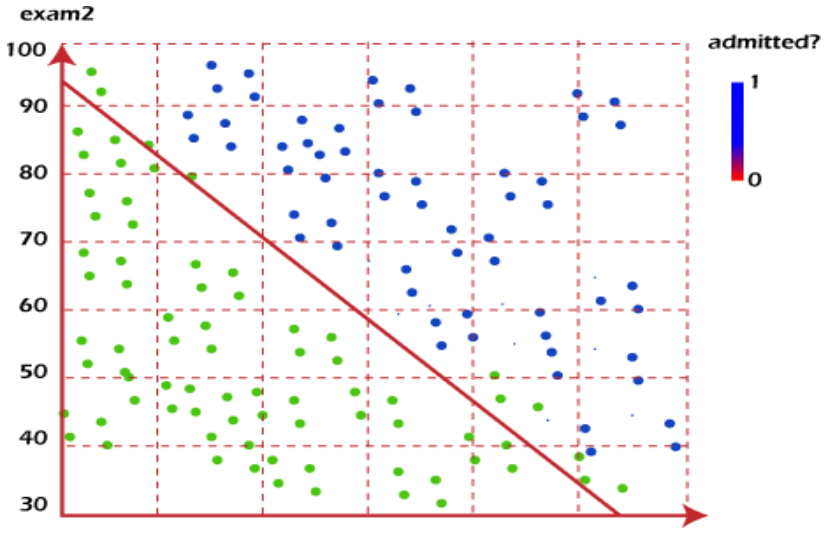

Fig.2 Classification Graph

3. Proposed System

An electronic file known as a Jupyter notebook contains both text descriptions and programming code. Additionally, embedded charts, plots, images, videos, and links can be included in Jupyter notebooks. A web browser like Firefox or Google Chrome is required to operate Jupyter notebooks. Although many different programming languages' code can be found in Jupyter notebooks, Python code is typically found in many of them. The app.py file contains the same kind of Python code as the Python code in a Jupyter notebook.

Markdown-formatted explanations and clarifications of the programming code can be found in the text description sections of Jupyter notebooks. The markdown file extension is.md. A Jupyter notebook's Markdown sections can format text to be bold, italic, form tables and lists, display code listings, and render images. An open-source Integrated Development Environment called Spyder can be thought of as a combination of the Python REPL and a Python module.py file with a markdown.md file sandwiched in between code sections. It was developed by scientists, is written in Python, and is only intended for use by engineers, data analysts, and scientists. Scientific Python Development IDE is another name. It provides an Editor for writing code, a console for evaluating it and viewing the results at any time, a variable defined during evaluation, and numerous other features for effective application or program development.

Spyder is a powerful Python-based scientific environment created by and for scientists, engineers, and data analysts. It is one of a kind because it combines the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package with the advanced editing, analysis, debugging, and profiling capabilities of a comprehensive development tool. In addition, a number of well-known scientific applications, such as NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy, and others, are built into Spyder. Spyder has a lot of built-in features, but third-party plugins can

make it even more powerful. Spyder can also be used as a PyQt5 extension library, allowing you to incorporate features like the interactive console and advanced editor into your own software. Flask is a Python-based web application framework. It is developed by Armin Ronacher, who is in charge of Pocco, a global group of Python enthusiasts. The Jinja2 template engine and Werkzeug WSGI toolkit are the foundations of Flask. Pocco projects are both.

Flask makes suggestions but does not enforce project layout or dependencies. The developer is in charge of selecting the libraries and tools they want to use. The community has provided a number of extensions that make it simple to add new features.

In order to set up your environment, two packages are required. Virtual so that a user can simultaneously create multiple Python environments. As a result, compatibility issues between different versions of the libraries, including Flask, can be avoided

```
from sklearn.model_selection import train_test_split

# split into 70:30 ration
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

# describes info about train and test set
print("Number transactions X_train dataset: ", X_train.shape)
print("Number transactions y_train dataset: ", y_train.shape)
print("Number transactions X_test dataset: ", X_test.shape)
print("Number transactions y_test dataset: ", y_test.shape)
```

```
Number transactions X_train dataset:  (4136, 35000)
Number transactions y_train dataset:  (4136,)
Number transactions X_test dataset:  (1035, 35000)
Number transactions y_test dataset:  (1035,)
```

Fig.3 Code Of The Project

Fig.4 Output Of The Project

4. Conclusion

The use of text messaging to identify unwanted and virus-infected email messages and prevent them from reaching the user's inbox has led to an increase in the SMS spam problem. In order to identify and eliminate spam messages, we are employing a variety of machine language classification algorithms. The most effective classification method is Multinomial Naive Bayes. A text classification-based multinomial naive bayes model was used to filter spam. My objective was to compare and observe the text behavior of incoming email messages in order to comprehend their customers and increase the accuracy rate based on text sanitization. According to the research conducted during the writing of this document, spam filtering relies heavily on the text classification process. Together with sanitizing the words, a good text mechanism classification can improve the effectiveness of spam classification and aid in future research into how spam behaves. Because of the model, this method can be applied to a variety of datasets with high accuracy for spam detection. The model was created to count the frequency of text, identify, split, sanitize, categorize it, and so on. In and of itself, the proposed algorithm enhances the Multinomial Naive Bayes. The reason for this is that once the distinction between spam and ham is established. In summary, Naive Bayes has gained popularity due to its simplicity in interpretation and the possibility of optimizing algorithms for improved performance in comparison to other more complex black box models like Neural Networks

References

1. Kaspersky (2014). Spam and Phishing Statistics Report Q1-2014. [Online]. Available: https://usa.kaspersky.com/resource-center/threats/spam-statistics-report-q1-2014 [Accessed July. 30, 2020]
2. Vergelis M, Shcherbakova T, Sidorina T. (2019). Spam and phishing in Q1 2019. [Online]. Available: https://securelist.lat/spam-and-phishing-in-q1-2019/88830 [Accessed July. 30, 2020]
3. https://www.techsoupcanada.ca/en/learning_center/10_sfm_explained
4. Cunningham P, Nowlan N, Delany S, Haahr M. (2003). A Case-Based Approach to Spam Filtering that Can Track Concept Drift. [Online]. Available: https://www.researchgate.net/publication/2474902_A_Case-Based_Approach_to_Spam_Filtering
5. _that_Can_Track_Concept_Drift [Accessed Sept. 2, 2020]
6. Christina V, Karpagavalli S, Suganya G.(2010).Email Spam Filtering using Supervised Machine Learning Techniques. [Online]. Available: https://www.researchgate.net/publication/50235326_Email_Spam_Filtering_using_Supervised_Ma chine_Learning_Techniques [Accessed Sept. 2, 2020]
7. Luo Q, Lui B, Yan J, He Z. (2011). Design and Implement a Rule-Based Spam Filtering System Using Neural Network. [Online]. Available: https://ieeexplore.ieee.org/document/6086218 [Accessed Sept. 2, 2020]
8. Malarvizhi R, Saraswathi K. (2013). Content-Based Spam Filtering and Detection Algorithms- An Efficient Analysis & Comparison. [Online]. Available: https://www.ijettjournal.org/volume- 4/issue-9/IJETT-V4I9P198.pdf

9. Pelletier L, Almahana J, Choulakian V. (2004). Adaptive filtering of spam. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1344731 [Accessed Sept. 5, 2020]
10. Himani B, Mahesh H. (2012). A review on support vector machine for data classification [Online].Available:https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1039.2508&rep=rep 1&type=pdf [Accessed Sept. 5, 2020]
11. Christina V, Karpagavalli S, Suganya G. (2010) Email Spam Filtering using Supervised Machine Learning Techniques [Online].Available: http://www.enggjournals.com /ijcse/doc/IJ CSE 10-02-09-151.pdf [Accessed Sept. 7, 2020]