AI SMART

SPAM DETECTOR

USING

AI

SUBMITTED

BY

M.Rubika

Au812921104036

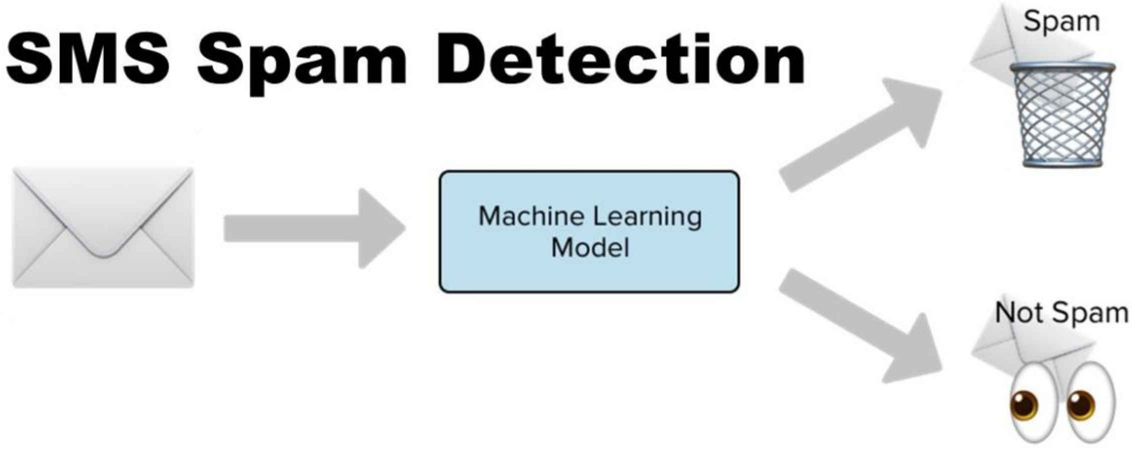
rubikamanoharan2003@gmail.com

INNOVATION

Problem Description

Understanding the problem is a crucial first step in solving any machine learning problem. In this article, we will explore and understand the process of classifying emails as spam or not spam. This is called Spam Detection, and it is a binary classification problem.

The reason to do this is simple: by detecting unsolicited and unwanted emails, we can prevent spam messages from creeping into the user's inbox, thereby improving user experience.

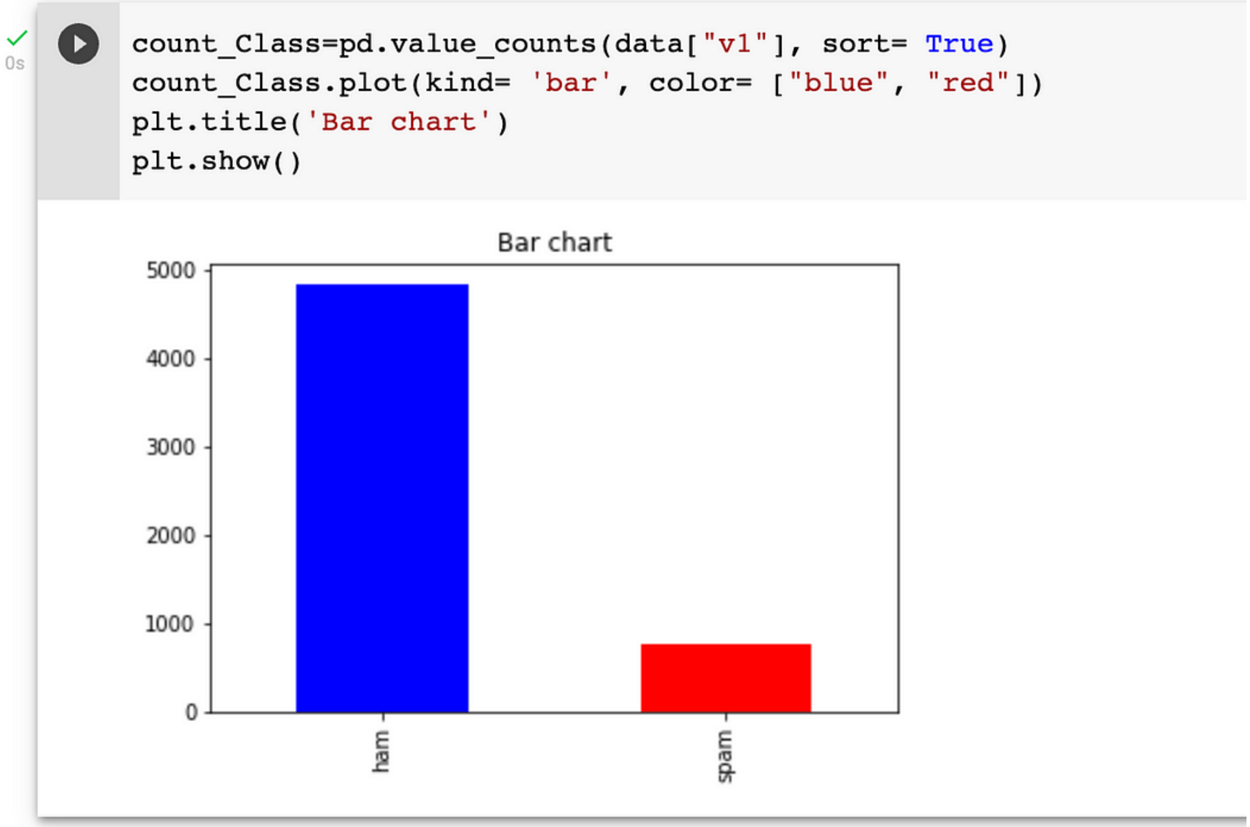


MACHINE LEARNING ALORITHM

NAÏVE BAYES

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features. The probability of an event 'A' occurring, given the condition 'B', is calculated

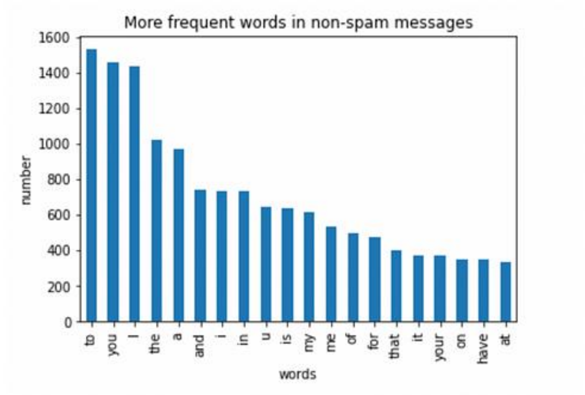
Visualizing the number of spam and non-spam(ham) messages

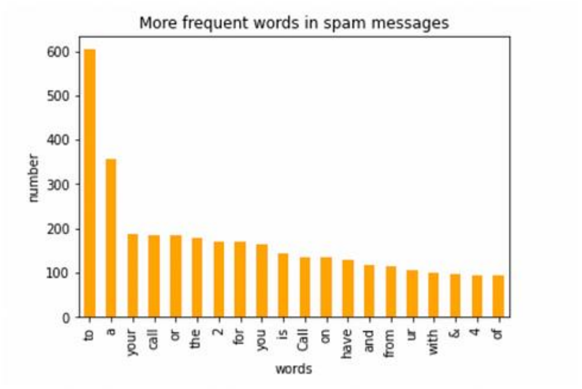


```
count_Class=pd.value_counts(data["v1"], sort= True)
count_Class.plot(kind= 'bar', color= ["blue", "red"])
plt.title('Bar chart')
plt.show()
```

As we can see, there are around 4800 ham messages and around 800 messages spam messages. The classification of spam messages will have the spam classification status — 1, as it is the event of our interest and occurs less often than the ham messages.

Frequent words in spam and non-spam messages.





We can see that the majority of frequent words in both classes are stop words such as ‘to’, ‘a’, ‘or’ and so on. These common words won’t be significant while classifying whether a message is significant or not. We need to analyze the presence of more relevant words in a spam message. So, to remove the common words, we can use **stopwords** function in Natural Language Toolkit library (nltk).

RANDOM FOREST

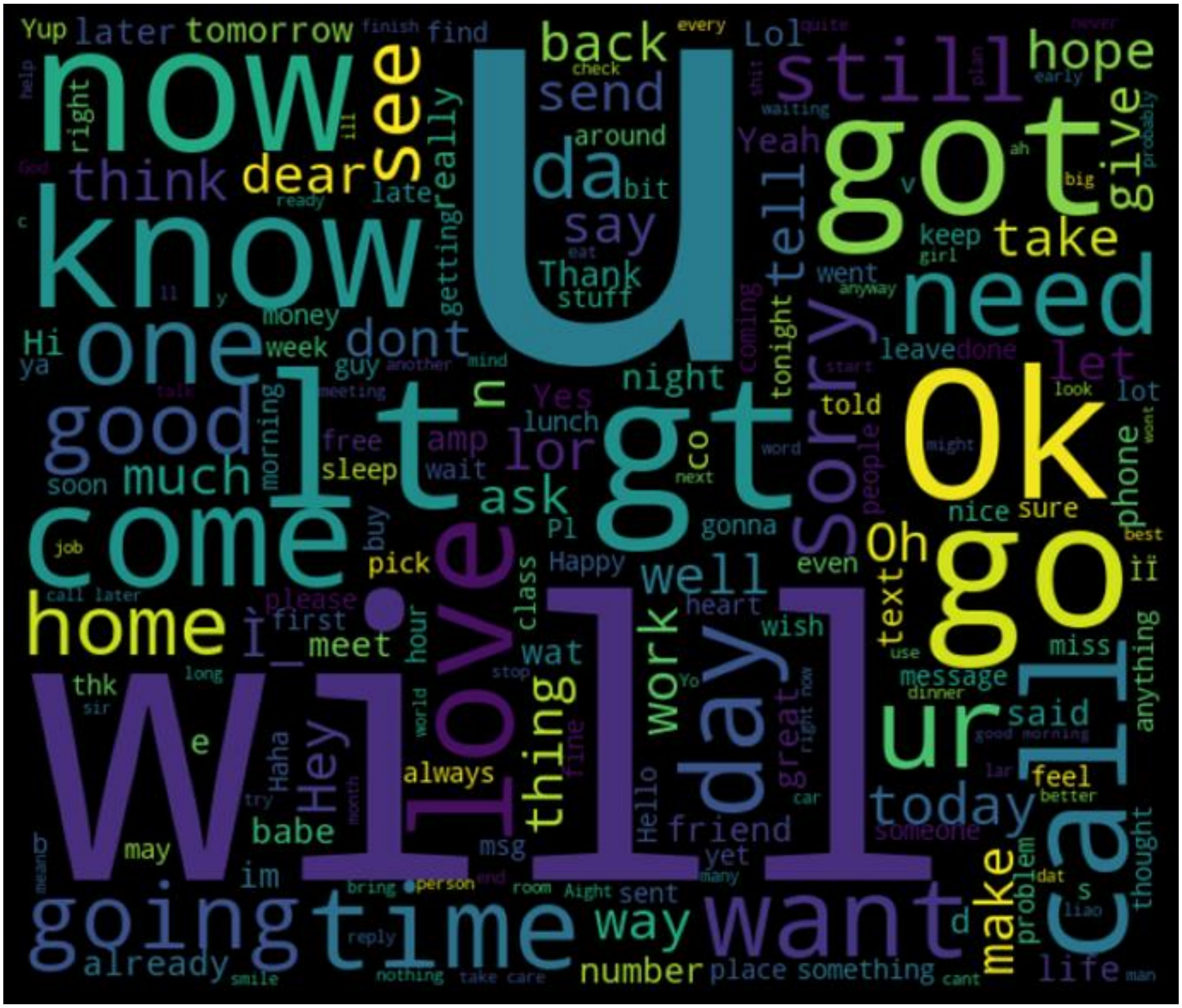
Spam sms detection is an important application of machine learning algorithms to filter out unwanted emails. There are several algorithms out there for this type of classification in the area of natural language processing. Usually spam emails have some typical words that make it quite obvious that the email is a spam. In this article we will walk through the text processing from spam and non-spam emails using nltk package. Particularly we will see the stemming and lemmatization procedure for NLP. We will also implement NB classifier as well SVC and Random Forest Classifier to detect spam emails and compare the classifiers in terms of accuracy. Let's dive in to it.

"NLTK is a leading platform for building Python programs to work with human language data". It is quite straightforward to process and tokenize the texts using nltk such as stemming and lemmatization which we will see later.

First we need to import the necessary packages.

Data Label

After importing the csv file containing the texts for spam and non-spam labels, I have created two data frames: one for real emails and the other for spam email which we will utilize for analysis.



By implementing these innovative techniques and strategies you can improve the effectiveness and adaptability of your AI spam detector in combating spam digital platforms .