



Spam mail detection

Author : Hardik Varma

Abstract

As we probably are aware E-mail spam is an ongoing issue for each individual. It is perhaps the greatest issue in the realm of the web. The email spam is nothing it's a notice of any organization/item or any sort of infection which is getting by the email customer post box with no notice. To take care of this issue the diverse spam filtering procedure is utilized. The spam filtering strategies are used to ensure our mailbox for spam sends. In this venture, we are using the Naïve Bayesian Classifier for spam grouping. The Naïve Bayesian Classifier is a straightforward and productive strategy for spam grouping. Here we are utilizing the Kaggle dataset for the arrangement of spam and non-spam messages. The outcome is to build the precision of the framework. The general precision of 97.63% accomplished by my model.

List Terms—E-mail spam, Classification, Feature Extraction, Naïve Bayesian Classifier.



Introduction

These days, email gives numerous approaches to send a large number of promotions at no expense to the sender. Therefore, much-spontaneous mass email, otherwise called spam email spread generally and turns into a genuine danger to the Internet as well as to society. For example, when the user received a large amount of e-mail spam, the chance of the user forgot to read a non-spam message increase. As a result, many e-mail readers have to spend their time removing unwanted messages. E-mail spam also may cost money to users with dial-up connections, waste bandwidth, and may expose minors to unsuitable content.

Over the past many years, many approaches have been provided to block e-mail spam. For filtering, some email spam is not being labeled as spam because the e-mail filtering does not detect that email as spam. Some existing problems are about accuracy for email spam filtering that might introduce some error.

A few ML calculations have been utilized in spam email filtering, however, Naïve Bayes calculation is especially well-known in commercial and open-source spam filters. This is because of its simplicity, which makes it easy to implement and just needs short training time or fast evaluation to filter email spam. The filter requires training that can be provided by a previous set of spam and non-spam messages. It monitors each word that happens just in spam, in non-spam messages, and in both.

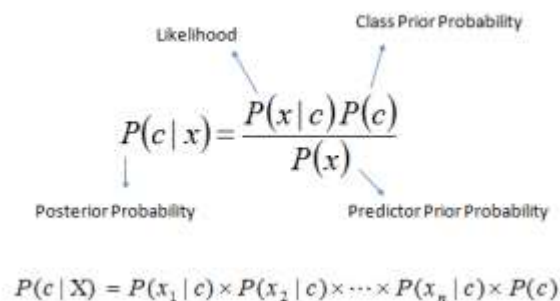


Move towards main contain now, let's understand the naïve Bayes algorithm first.

What is the Naive Bayes algorithm?

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.



$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$

Here,

- $P(c|x)$ is the posterior probability of class (c , target) given predictor (x , attributes).
- $P(c)$ is the prior probability of class.
- $P(x|c)$ is the likelihood which is the probability of predictor given class.
- $P(x)$ is the prior probability of predictor.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naïve Bayes is known to outperform even highly sophisticated classification methods.

Text classification/ Spam Filtering/ Sentiment Analysis: Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments).

$$P(\text{SPAM} / \text{Word}) = [P(\text{Word} / \text{SPAM}) \times P(\text{SPAM})] / P(\text{Word})$$

$$P(\text{Word}) = P(\text{Word} / \text{Spam}) * P(\text{Spam}) + P(\text{Word} / \text{Not Spam}) * P(\text{Not Spam})$$

$$\prod_i P(W_i / \text{SPAM}) \times P(\text{SPAM}) / P(\text{Word})$$

Here we will utilize naïve Bayes classifier like this.

Along these lines, for this, I took one dataset of messages from Kaggle. That dataset contains both spam and ham (not spam) messages.

<https://drive.google.com/open?id=11CsLTQRaV7QusmP2k7x55uXQsdrLV1ky>

dataset.

This informational index prepared one model with a naïve Bayes classifier. With the utilization of flask, I deploy that model to the site page. Furthermore, made one HTML page. after this, I have that site page from free facilitating.

<https://spam-mail-detection.herokuapp.com/>

This is website which I have host.

<https://github.com/Kritikal30/Spam-Mail-Detection>

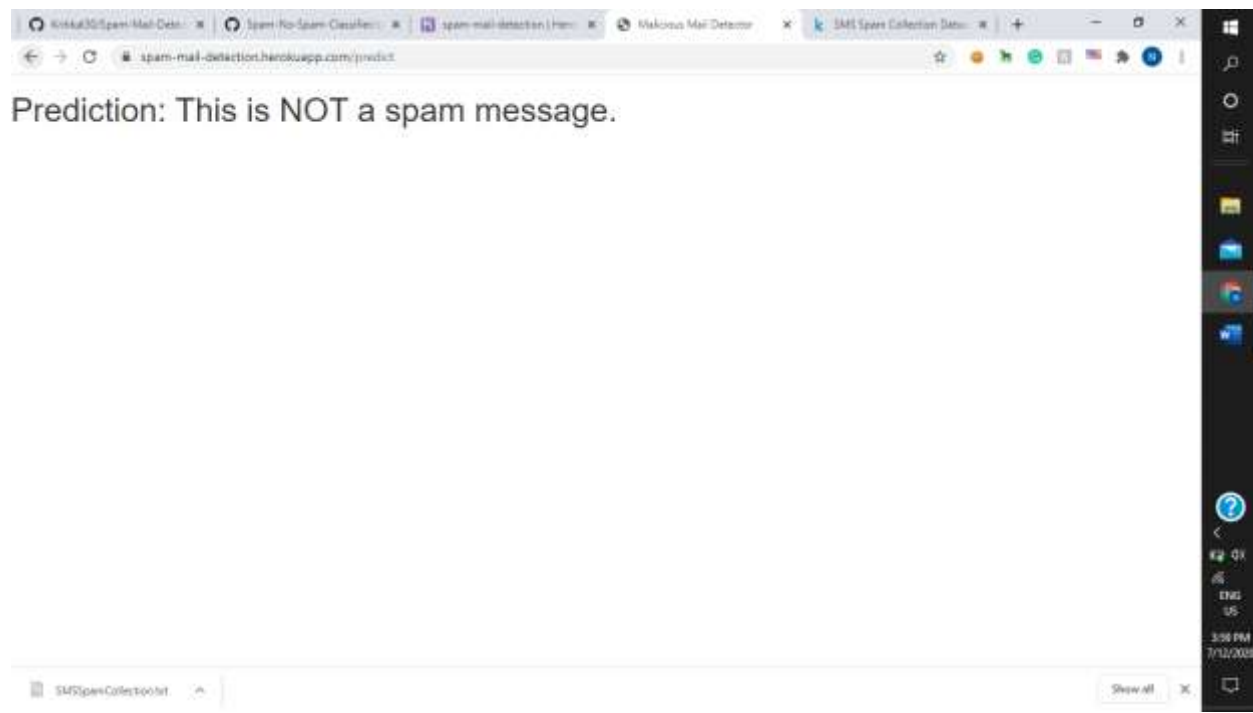
you can check my GitHub repo for more details.

The screenshot shows the Heroku dashboard for the 'spam-mail-detection' application. The top navigation bar includes 'Personal', 'spam-mail-detection', and buttons for 'Open app' and 'More'. The main content area is divided into several sections:

- Installed add-ons:** A message states 'There are no add-ons for this app. You can add add-ons to this app and they will show here. [Learn more](#)'.
- Dynamic formation:** A message states 'This app is using free dynos'.
- Web:** A message states 'gunk.com: app:app' with a 'ON' status.
- Collaborator activity:** A list of activities for the user 'led.nat@301@gmail.com' (1 deploy):
 - Deployed: 28 Oct 2021 at 7:30 PM - v3
 - Build succeeded: 28 Oct 2021 at 7:30 PM - [View build log](#)
 - Enable Logplex: 28 Oct 2021 at 7:37 PM - v2
 - Initial release: 28 Oct 2021 at 7:37 PM - v1

The bottom of the dashboard shows the file 'TMSPamCollector.txt' and a 'Show all' button.

The screenshot shows the 'spam-mail-detection' application interface. The top navigation bar includes 'spam-mail-detection', 'Heroku', and 'Mahouai Mail Detector'. The main content area is titled 'Enter Your message here:' and features a large text input field with the placeholder text 'Enter Your Message Here...'. Below the input field is a 'Predict' button. The bottom of the interface shows the file 'TMSPamCollector.txt' and a 'Show all' button.



Conclusion

E-mail spam filtering is an important issue in the network security and machine learning techniques; Naïve Bayes classifier that used has a very important role in this process of filtering e-mail spam. The quality of performance Naïve Bayes classifier is also based on datasets that used. As can see, dataset that have fewer instances of e-mails and attributes can give good performance for Naïve Bayes classifier. Naïve Bayes classifier also can get highest precision that give highest percentage spam message manage to block if the dataset collects from single e-mail accounts. So, we can see, why performance of Naïve Bayes classifier is good when used SPAMBASE dataset.

Reference

<https://youtu.be/bjsJOl8gz5k>

<https://youtu.be/O2L2Uv9pdDA>

<https://towardsdatascience.com/spam-filtering-using-naive-bayes-98a341224038>

