Email Classification HAM or SPAM

Kris Ghimire, Rajesh Satluri, Suchismita Moharana

June 6, 2021

Abstract

An increase in the volume of spam email has become a major issue for all internet users. In this research paper, researchers are going to build Naïve Bayes a popular machine learning technique to filter emails into ham and spam.

1. Introduction:

If you login to any of your mailbox, you will quickly notice that more than half of your emails are unwanted and often unsolicited junk that is filling up storage. Often time these emails are phishing mail intended to steal information. These annoying and useless emails are called spam. Filtering out only the email we want and discarding unnecessary spam is not an easy task. It is excessively time consuming and almost impossible to manually identify only the ham out of thousands of emails we receive. To overcome this challenge and automate the process of identifying ham and spam, in this research paper, we will use Natural Language Processing (NLP) pipeline to build a machine learning classifier to classify email into ham and spam. The figure below (figure 1) shows the general pipeline of how the classification process works.

Figure 1

2. Methods

Dataset:

	EmailText	Classification	Norm_EmailText
0	> I think keeping the number of middle letter...	0	> think keeping number middle letters consiste...
1	URL: http://jeremy.zawodny.com/blog/archives/0...	0	url: date: 2002-10-03t21:59:32-08:00 me, rss r...
2	URL: http://boingboing.net/#85521686 Date: Not...	0	url: date: supplied president's niece longer f...
3	> Martin Mentioned: > >I've used this a few t...	0	> martin mentioned: > >i've used times thoroug...
4	\nMortgage companies make you wait...They Dem...	1	mortgage companies make wait...they demand int...

Table 1

The dataset provided for this case study is slightly different than the usual format. It would be much easier if we always get dataset in csv format. However, data comes in different forms and one of the biggest challenge of Data Scientist is to get the data together. The data were provided in five different folders easy_ham, easy_ham2, hard_ham, spam2 and spam, all of which contained multiple files of raw text and html format emails. The first step was to get only the relevant content from all the given files. We have identified that there are different email content type of email exists. Especially it is tricky to parse Multipart/* email content types. We have adopted recursive search approach to identify inner content types and parse the email content. Used BeautifulSoup and HTMLparser and email package to extract only the contents relevant for NLP pipeline to classify Spam or Not Spam and created dataframe (Table 1).

Out of 9000+ email files, there were 17 of them which we could not read due to encoding issue, this is on windows machine. But when we ran on Mac OS, only 3 email files were not able to read. It is very small volume as compared to 9000+ and we are ignoring these 17 or 3 emails. Skipping process is automated and the same model can be running on heterogeneous platforms. There are some duplicates in the dataset, we decided to build two models, one including duplicates and another without duplicates. We didn't see much difference in either of them in term of model performance metrics.

Upon visualizing target features, (figure 2) we noticed unequal distribution of classes, ham and spam which is quite common and always a challenge in classification problems. As with most machine learning algorithms, uneven distribution of class ratio in Naïve Bayes classifier could leads to an inaccurate estimate of class prior which could then potentially decrease the predictive performance.

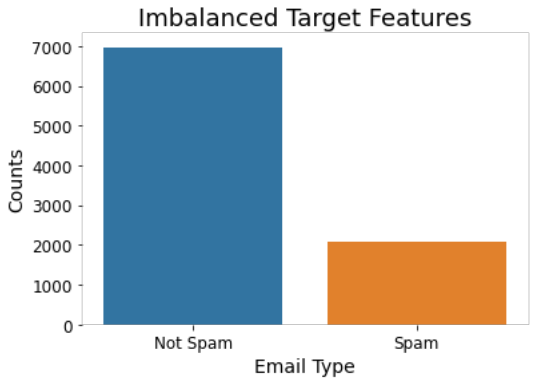

Figure 2

Data Exploration: Text cleaning also known as text normalizing is crucial before exploration the data in NLP pipeline. We applied majority of text normalization process to extract meaningful information out of text. Some of which includes, case conversions, removing stop words, removing characters that are not alpha numeric, lemmatization, tokenization etc. We wanted to explore the words that are most frequently used in spam email. As expected with all spam email text, we can notice in the figure 3 below that word 'FREE' is the most used word

followed by other common spam words such as please, money, get, click, site, one etc. as top ten frequent used spam words.

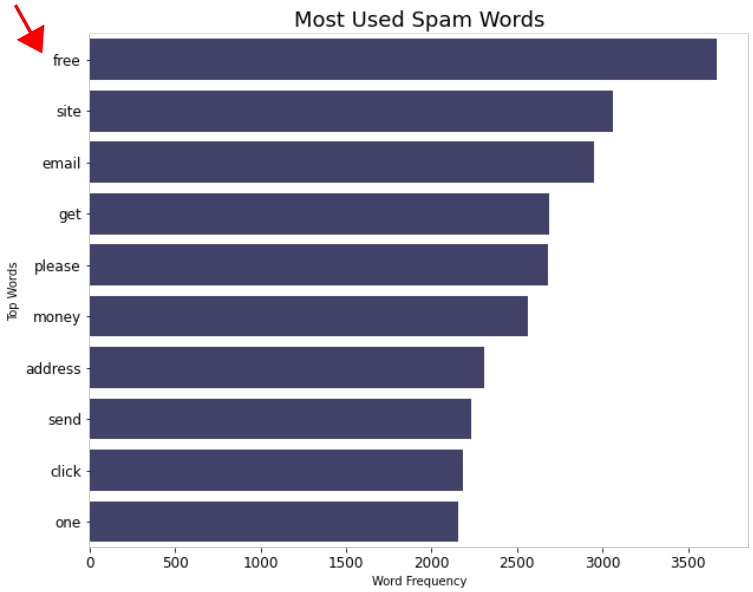

Figure 3

Visualizing not spam (ham) will not provide any insight to us as we are mostly focusing on understanding about spam email and how can eliminate receiving spam junk. However, just for the exploratory purpose we made word cloud of ham text as shown in figure 4 below.

Figure 4

Independent Assumption: One of the key assumptions of Naïve Bayes classifier which is what we are focusing on using for email classification is that it naively make 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.

Bag of words feature extraction: Machine learning algorithms need numeric data to perform calculation. Text data will mean nothing to an algorithm. There are various methods of feature extraction. A most popular and simple to understand is a method of feature extraction using bag-of-words (BOW). A BOW is a representation of text data that describes the occurrence of words

within word corpus. It is called bag because there is no order of occurrence of words. We will use sklearn CountVectorizer which makes implementation of both tokenization and occurrence counting in a single class.

Train Test Split: While dataset is split into training and test set (80/20 split) to keep test data separate and to perform modeling only on training set.

NLP pipelines: We will follow NLP pipeline to build the classification. Figure 5.

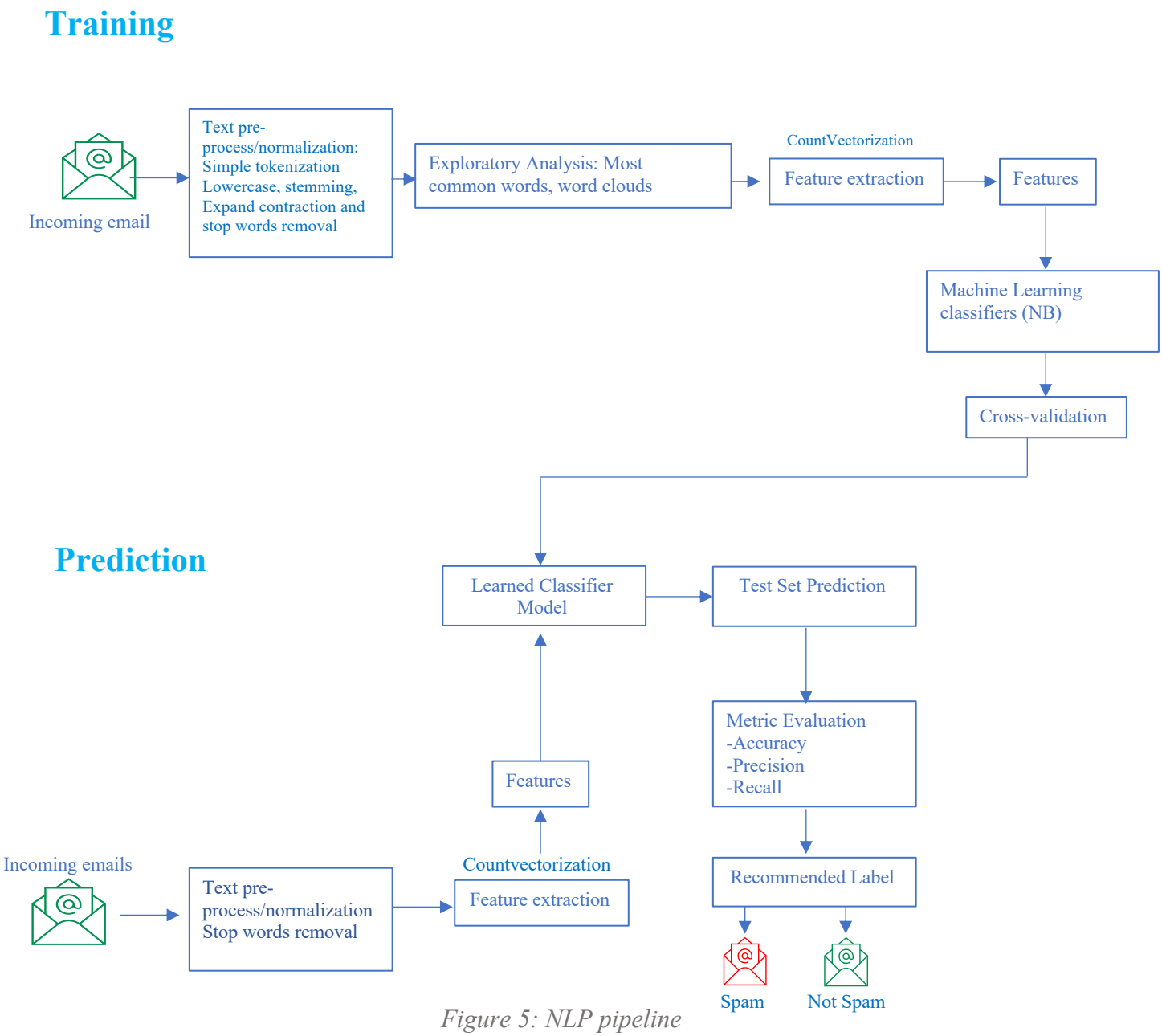

Figure 5: NLP pipeline

Imbalance target features: Dataset had imbalanced target features. Modeling on imbalanced target features would give evaluation metrics that are not accurate. We will implement Naïve Bayes based 'ComplementNB' classification technique to balance the dataset.

Cross-Validation: We will use 10- fold cross validation k=10 because with stratifiedKFold, the class distribution in the dataset is preserved in the training and test splits. Shuffle is set to true so that the splitting will be random.

Naïve Bayes (NB) Classification: Naïve Bayes is slightly different from other Machine Learning (ML) algorithms. With NB algorithms we apply statistical distributions to Bayes rule (Figure 6) and use its mathematics to produce results.

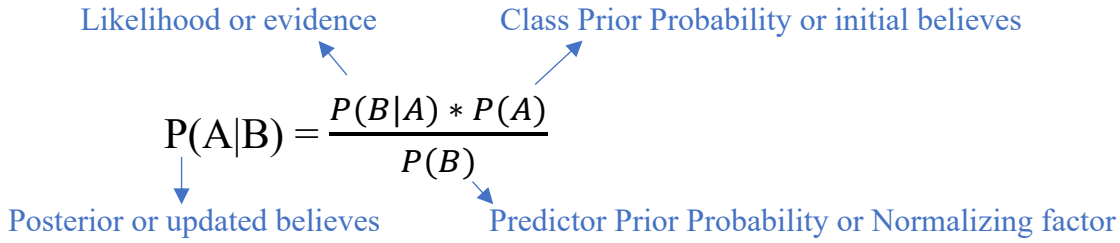

Figure 6: Bayes Rule

We would read above NB formula as the probability that A occurs give B is the probability that B occurs given A times the probability of occurring A divided by the probability of occurring B.

To explain NB in simple term, what we have is, we have our prior or old believes (probability distribution), when the new evidence occurs, we update the believes. Basically, we take that new evidence and multiply it by our prior believes to get our posterior or updated beliefs.

3. Result:

Model 1:

Models were build using several approaches. First, we built model 1 on the data as it is without balancing. The 10- fold cross-validation gave mean accuracy of about 97.7 % and standard deviation of 0.0064. Model did a good job on both majority as well as minority class as verified by the Roc score of 0.97 as well as precision and recall score (Figure 7, 8, 9).

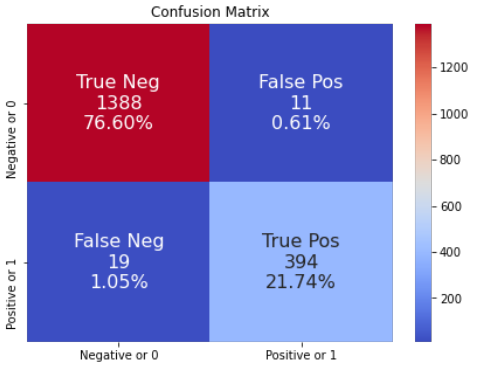

Figure 7 Confusion metrics

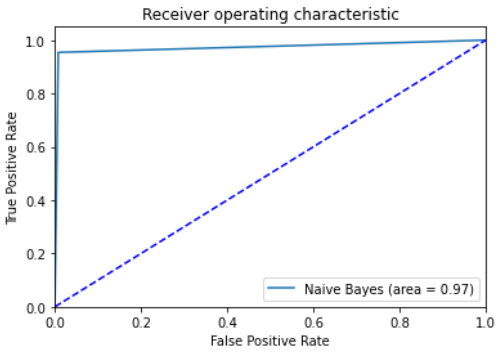

Figure 8: ROC Curve

	precision	recall	f1-score
0	0.99	0.99	0.99
1	0.97	0.95	0.96
accuracy			0.98
macro avg	0.98	0.97	0.98
weighted avg	0.98	0.98	0.98

Figure 9. Classification Report

Model – 2:

We built second model, model -2 using ComplementNB algorithm. ComplementNB, according to sklearn documentation is particularly suited for imbalanced data sets. Balanced data model produced slightly better accuracy, 97.8% (10-f cross-validated) compared to model 1 shown in Figure 10,11. Precision score (figure 12) suggests that model 2 is better at classifying not ham email than spam email.

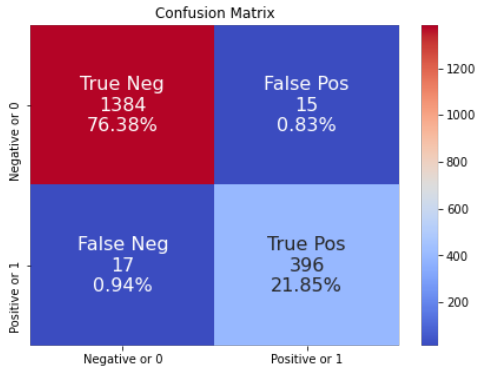

Figure 10: Confusion metrics

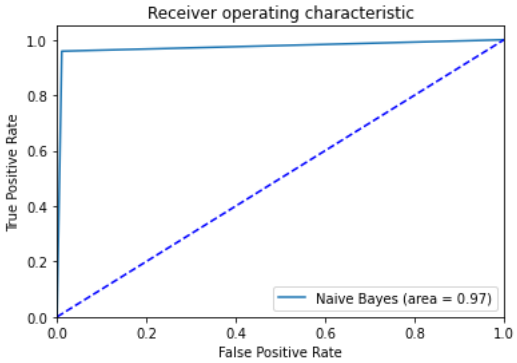

Figure 11: ROC Curve

	precision	recall	f1-score
0	0.99	0.99	0.99
1	0.96	0.96	0.96
accuracy			0.98
macro avg	0.98	0.97	0.97
weighted avg	0.98	0.98	0.98

Figure 12: Classification Report

Model- 3

Finally, we have built third model, model -3 using ComplementNB by dropping duplicates. We noticed slight dip in the accuracy with 10-fold cross validated mean accuracy to be 96.2% and standard deviation 0.0077. We noticed no change in ROC score of 0.97. Figure 13,14, 15.

Another interesting metrics to note is precision and recall. The model produced decline in recall score 0.95 for spam email of while recall is 0.98 for not spam.

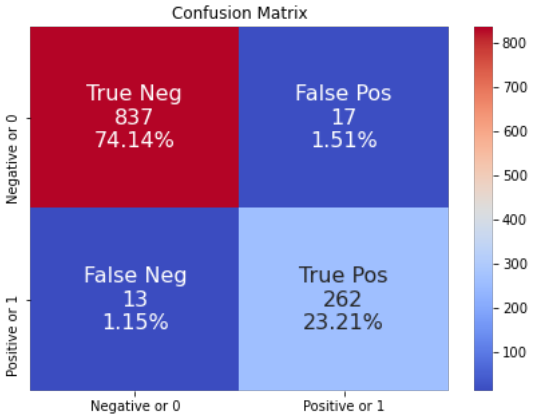

Figure 13 Confusion metrics

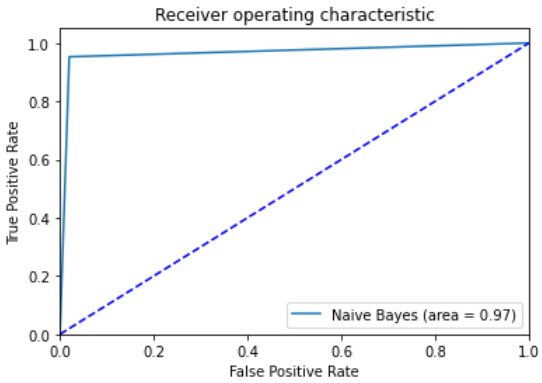

Figure 14. ROC Curve

	precision	recall	f1-score
0	0.98	0.98	0.98
1	0.94	0.95	0.95
accuracy			0.97
macro avg	0.96	0.97	0.96
weighted avg	0.97	0.97	0.97

Figure 15: Classification Report

4. Conclusion:

In this research paper, we built three different NB classifiers to meet the business requirement of filtering spam email with high precision and accuracy. We implemented several NLP techniques to clean and normalize the text data to feed to ML classifier.

Out of all three models, model 3 would be a better model even though all the models have similar accuracy, as duplication of data used in other two model could provide misleading model evaluation metrics.
Although the NB model (model 3) has been quite effective in filtering spam with high precision and accuracy, further text cleaning/normalization, feature extraction and advanced Deep Learning-based NLP could be applied and compared with Naïve Bayes model in various aspects.