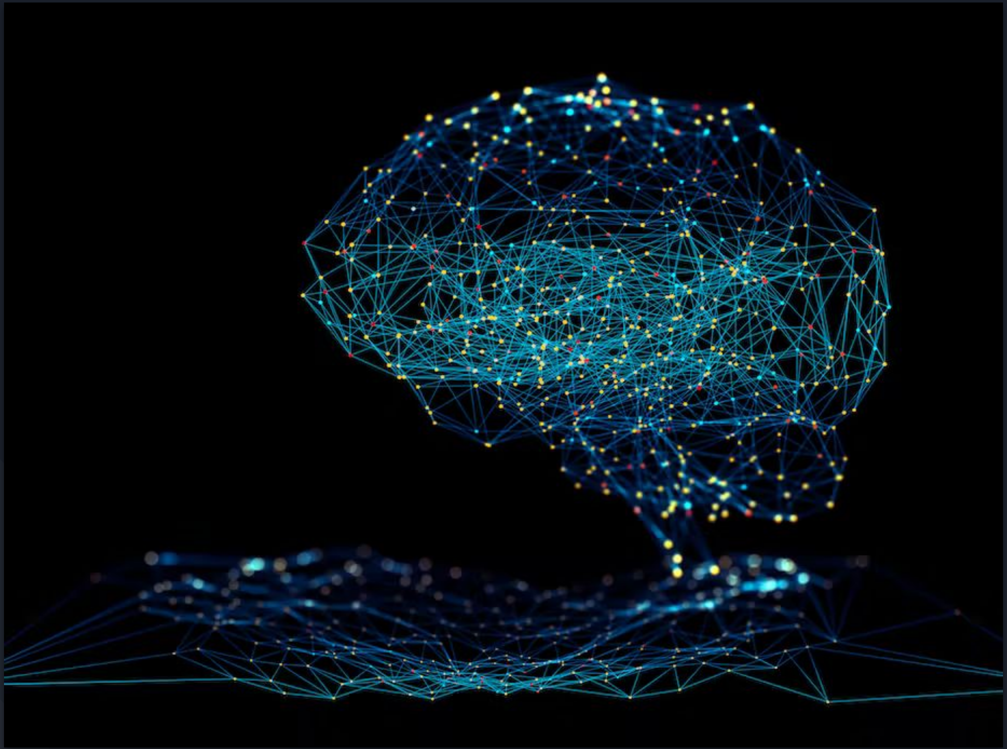
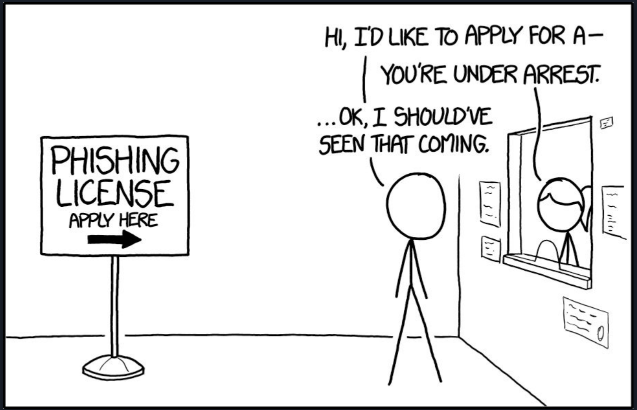
Applying Neural Networks To The Classification Of Email Spam



By Kaelyn Isaac Parris

Introduction


PHISHING LICENSE
APPLY HERE
HI, I'D LIKE TO APPLY FOR A—
YOU'RE UNDER ARREST.
...OK, I SHOULD'VE SEEN THAT COMING.

- UBE
  - Unsolicited Bulk Email
- According to AAG 2022 Spam Statistics
  - 3.4 billion spam emails sent daily
  - 48% of all email in 2022 was spam
- A classic problem in machine learning
- Traditionally solved with non-neural net methods
  - Early application of Naive Bayes - (Heckerman et al. 1998)
- Aiming to build on the work of (Magdy, Abouelseoud, and Mikhail 2022)
  - 3 classes to predict
    - Ham, Spam, & Phishing
  - 99% precision
- Supervised learning - 2 classes to predict

Can Neural Nets measure up to traditional predictive statistical methods?

Ham, Spam, & Phishing

- Terms “Ham” and “Spam” originate from Monty Python sketch
- Repetitive chanting of “spam”
- Ham derived as the opposite of spam
- Phishing refers to “fishing” for personal information

Data

- Sources of data:
    - Spambase
        - Pre-Processed
        - Establish comparison with standard approaches
        - 4601 emails
        - 50/50 split
    - Enron Corpus
        - Used curated data from authors of “Spam Filtering with Naive Bayes - Which Naive Bayes?,” (Metsis, Androutsopoulos, and Paliouras 2006)
        - 1935 Ham Emails
        - 8,800 Spam
    - Complemented with SpamAssassin to reach 50/50 split
        - Total 11,226 emails
        - Nearly 50/50 split

How To Process An Email

(Guzella and Caminhas 2009)

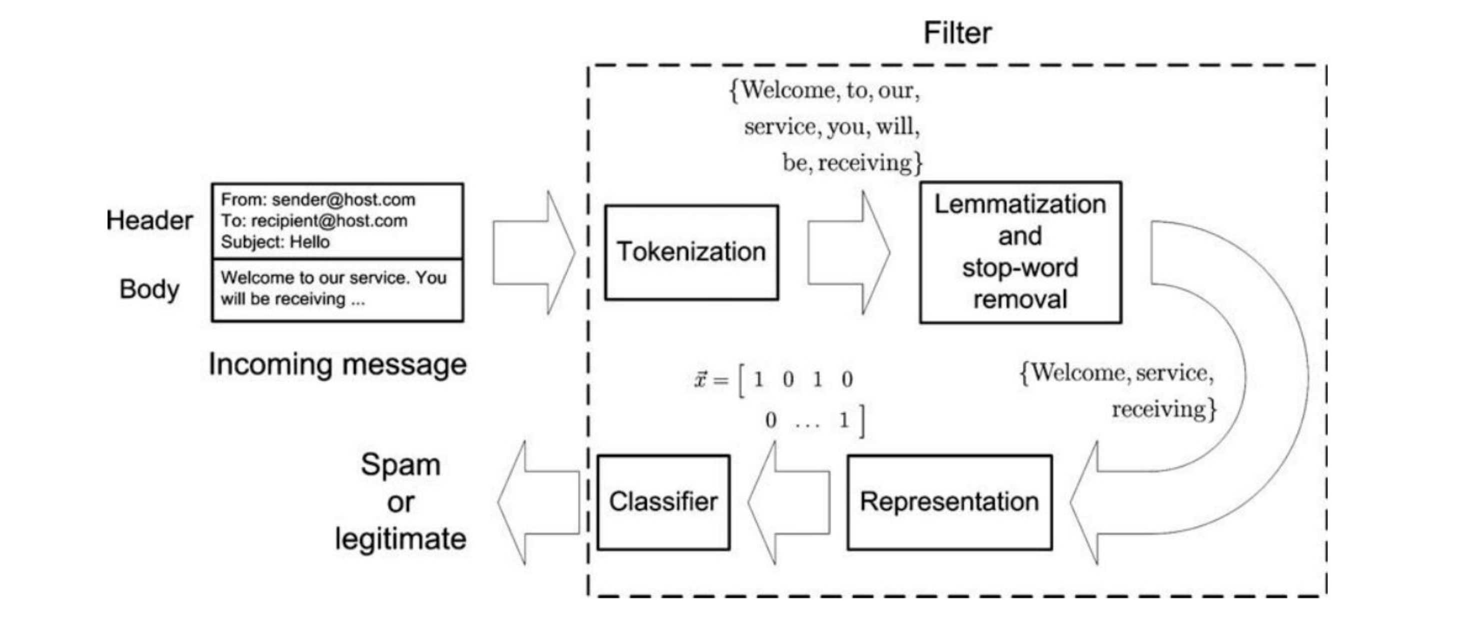
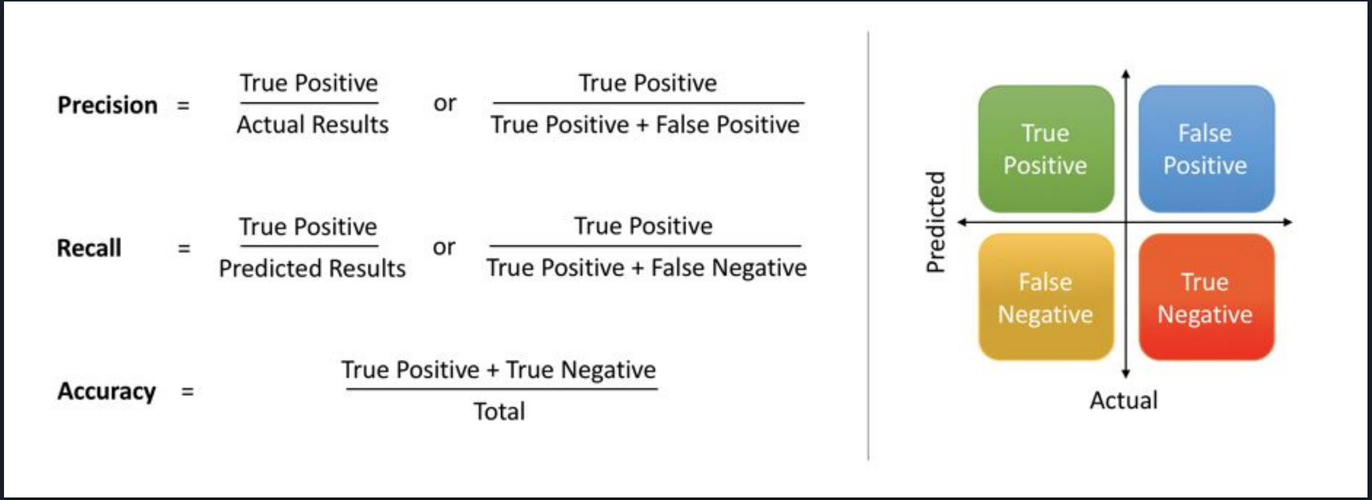

Filter
Header
Body
From: sender@host.com
To: recipient@host.com
Subject: Hello
Welcome to our service. You will be receiving ...
Incoming message
Tokenization
{Welcome, to, our, service, you, will, be, receiving}
Lemmatization and stop-word removal
{Welcome, service, receiving}
Representation
$\vec{x} = [1\ 0\ 1\ 0\ 0\ \dots\ 1]$
Classifier
Spam or legitimate

Fig. 1. An illustration of some of the main steps involved in a spam filter.

Choosing A Metric:

Precision chosen to prioritize the reduction of false positives.



$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \text{ or } \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

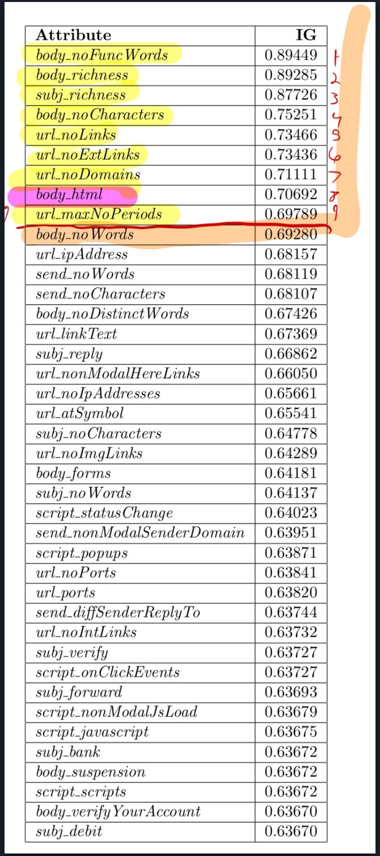
$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \text{ or } \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\textbf{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$

True Positive	False Positive
False Negative	True Negative

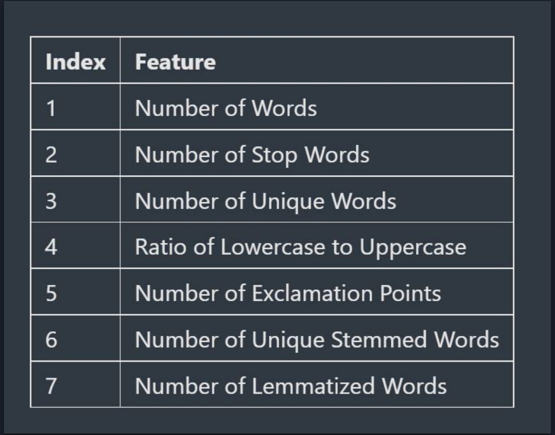
Feature Selection

- (Toolan and Carthy 2010) identify over 40 features used in the literature
  - How to determine features?
    - Entropy/Information Gain
      - Entropy: impurity in dataset
      - Information Gain: Which features reduce entropy the most?
  - For Spam/Ham classification
    - 9 features with the highest information gain



Attribute	IG	
body_noFuncWords	0.89449	1
body_richness	0.89285	2
subj_richness	0.87726	3
body_noCharacters	0.75251	4
url_noLinks	0.73466	5
url_noExtLinks	0.73436	6
url_noDomains	0.71111	7
body_html	0.70692	8
url_maxNoPeriods	0.69789	9
body_noWords	0.69280	
url_ipAddress	0.68157	
send_noWords	0.68119	
send_noCharacters	0.68107	
body_noDistinctWords	0.67426	
url_linkText	0.67369	
subj_reply	0.66862	
url_nonModalHereLinks	0.66050	
url_noIpAddresses	0.65661	
url_atSymbol	0.65541	
subj_noCharacters	0.64778	
url_noImgLinks	0.64289	
body_forms	0.64181	
subj_noWords	0.64137	
script_statusChange	0.64023	
send_nonModalSenderDomain	0.63951	
script_popups	0.63871	
url_noPorts	0.63841	
url_ports	0.63820	
send_diffSenderReplyTo	0.63744	
url_noIntLinks	0.63732	
subj_verify	0.63727	
script_onClickEvents	0.63727	
subj_forward	0.63693	
script_nonModalJsLoad	0.63679	
script_javascript	0.63675	
subj_bank	0.63672	
body_suspension	0.63672	
script_scripts	0.63672	
body_verifyYourAccount	0.63670	
subj_debit	0.63670	

Created 7 features for first test:



Index	Feature
1	Number of Words
2	Number of Stop Words
3	Number of Unique Words
4	Ratio of Lowercase to Uppercase
5	Number of Exclamation Points
6	Number of Unique Stemmed Words
7	Number of Lemmatized Words

Models

(Precision Metric)

Dummy Classifier

- 50%

Naive Bayes (Gaussian)

- 82% on preprocessed
- 63% With Chosen Features

DecisionTree:

- 96%

Neural Net:

- 93-96%
- Runtime ~4 minutes
  - (early stopping)

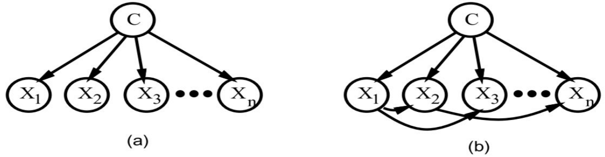
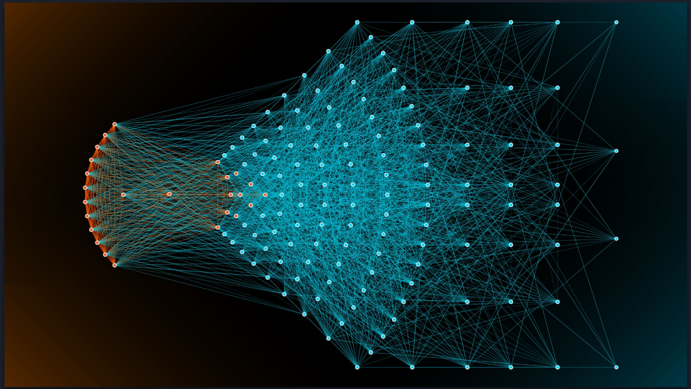
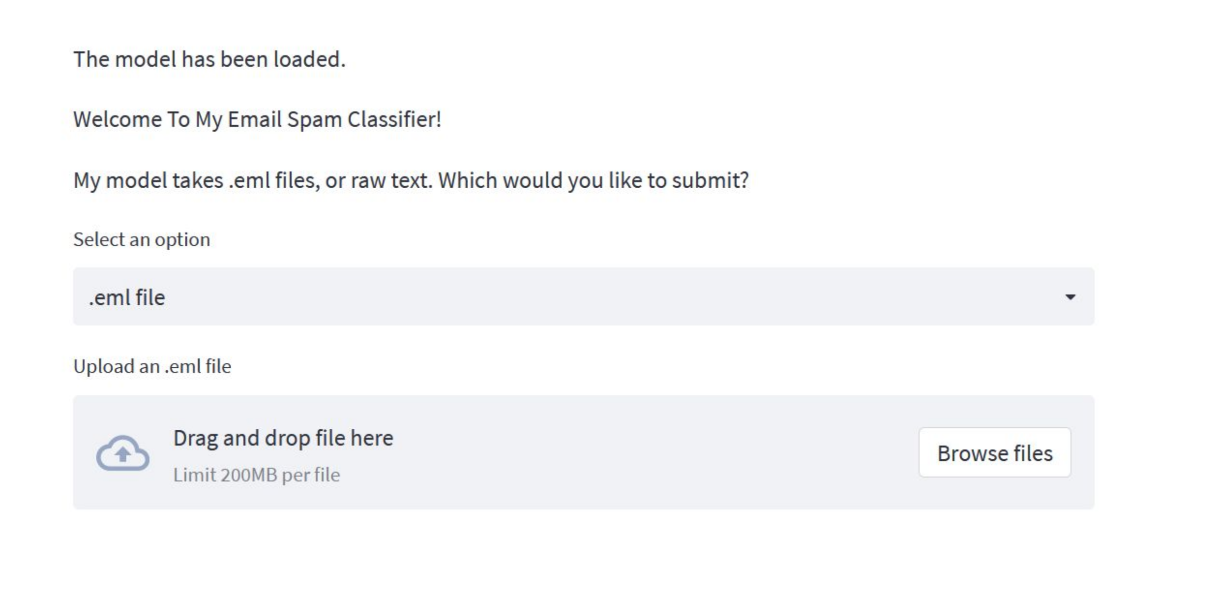


Figure 1: Bayesian networks corresponding to (a) a Naive Bayesian classifier; (b) A more complex Bayesian classifier allowing limited dependencies between the features.

(Heckerman et al. 1998)



Model Deployment



The model has been loaded.

Welcome To My Email Spam Classifier!

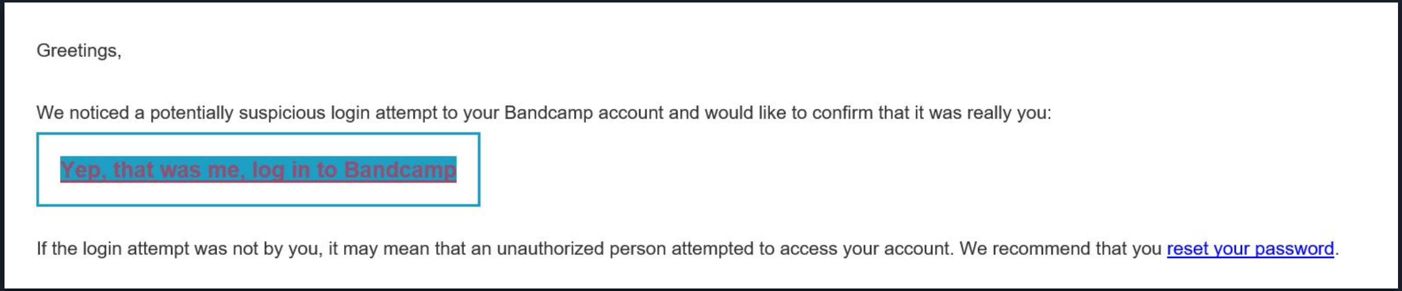
My model takes .eml files, or raw text. Which would you like to submit?

Select an option

.eml file

Upload an .eml file

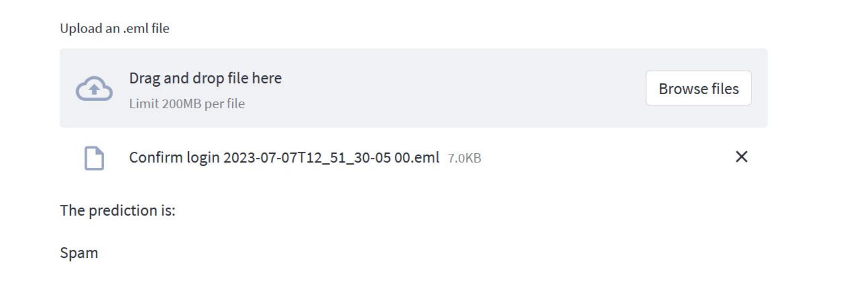
Dude, Where's My HTML?



Greetings,

We noticed a potentially suspicious login attempt to your Bandcamp account and would like to confirm that it was really you:

If the login attempt was not by you, it may mean that an unauthorized person attempted to access your account. We recommend that you reset your password.



Confirm login 2023-07-07T12_51_30-05 00.eml 7.0KB

The prediction is:

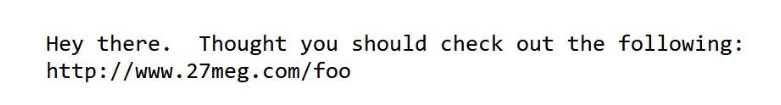
Spam

This file is not spam!

What’s Next?

- Process 9 Features
  - Currently in progress
- Current Issues:
  - HTML!
- Spammers have sneakier methods!
  - Images!

The spam of the future:



```
Hey there.  Thought you should check out the following:
http://www.27meg.com/foo
```

(“A Plan for Spam”, 2002)

Thanks For Listening!

Questions?

Contact info:

Kaelyn Isaac Parris
kaelyn_parris@protonmail.com
Kaelyn Parris | LinkedIn