

**Syracuse University**

**IST-664 Final Project  
Email Spam Corpora**

ThulasiRam Ruppakrishnan

IST 664

Professor Michael Larche

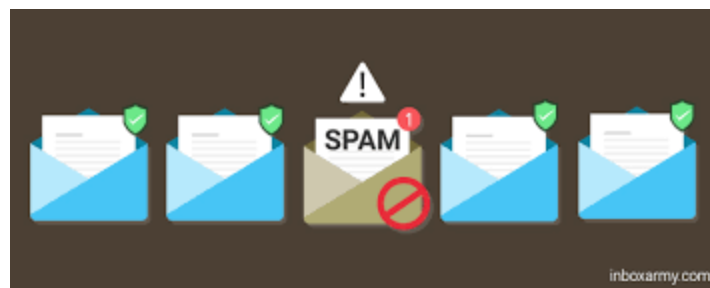
## Contents

Introduction .....	3
Analysis and Models .....	5
About the data.....	5
Models.....	8
Results .....	14

# Introduction

Email spam, also referred to as junk email, is unsolicited messages sent in bulk by email (spamming). The name comes from Spam luncheon meat by way of a Monty Python sketch in which Spam is ubiquitous, unavoidable, and repetitive. Email spam has steadily grown since the early 1990s, and by 2014 was estimated that it made up around 90% of email messages sent.

Since the expense of the spam is borne mostly by the recipient, it is effectively postage due advertising. This makes it an excellent example of a negative externality. The legal definition and status of spam varies from one jurisdiction to another, but laws and lawsuits have nowhere been particularly successful in stemming spam.



Most email spam messages are commercial in nature. Whether commercial or not, many are not only annoying, but also dangerous because they may contain links that lead to phishing web sites or sites that are hosting malware - or include malware as file attachments.

Spammers collect email addresses from chat rooms, websites, customer lists, newsgroups, and viruses that harvest users' address books. These collected email addresses are sometimes also sold to other spammers.

## Spam Detection

There are currently different approaches to spam detection. These approaches include blacklisting, detecting bulk emails, scanning message headings, greylisting, and content-based filtering:

- **Blacklisting** is a technique that identifies IP addresses that send large amounts of spam. These IP addresses are added to a Domain Name System-Based Blackhole List and future email from IP addresses on the list are rejected. However, spammers are circumventing these lists by using larger numbers of IP addresses.
- **Detecting bulk emails** is another way to filter spam. This method uses the number of recipients to determine if an email is spam or not. However, many legitimate emails can have high traffic volumes.

- **Scanning message headings** is a fairly reliable way to detect spam. Program written by spammers generate headings of emails. Sometimes, these headings have errors that cause them to not fit standard heading regulations. When these headings have errors, it is a sign that the email is probably spam. However, spammers are learning from their errors and making these mistakes less often
- **Greylisting** is a method that involves rejecting the email and sending an error message back to the sender. Spam programs will ignore this and not resend the email, while humans are more likely to resend the email. However, this process is annoying to humans and is not an ideal solution.

Current spam techniques could be paired with **content-based spam filtering** methods to increase effectiveness. Content-based methods analyze the content of the email to determine if the email is spam. The goal of this project was to analyze machine learning algorithms and determine their effectiveness as content-based spam filters.







# Analysis and Models





## About the data

The dataset is one produced for detecting Spam emails from the Enron public email corpus. In addition to some small numbers of Spam already in the corpus, additional spam emails were introduced into each user's email stream in order to have a sufficient number of spam examples to train a classifier. The non-Spam emails are labeled "ham". (See this paper for details: [http://www.aueb.gr/users/ion/docs/ceas2006\\_paper.pdf](http://www.aueb.gr/users/ion/docs/ceas2006_paper.pdf) ) The dataset that we have was gleaned from their web site at <http://www.aueb.gr/users/ion/data/enron-spam/>.

Although there are 3 large directories of both Spam and Ham emails, only the first one is used here with 3,672 regular emails in the "ham" folder, and 1,500 emails in the "spam" folder.

Name	Date modified	Type	Size
 0001.1999-12-10.farmer.ham.txt	2/29/2020 9:34 AM	Text Document	1 KB
 0002.1999-12-13.farmer.ham.txt	2/29/2020 9:34 AM	Text Document	5 KB
 0003.1999-12-14.farmer.ham.txt	2/29/2020 9:34 AM	Text Document	1 KB
 0004.1999-12-14.farmer.ham.txt	2/29/2020 9:34 AM	Text Document	2 KB

### Ham list

 0006.2003-12-18.GP.spam.txt	2/29/2020 9:35 AM	Text Document	2 KB
 0008.2003-12-18.GP.spam.txt	2/29/2020 9:35 AM	Text Document	1 KB
 0017.2003-12-18.GP.spam.txt	2/29/2020 9:35 AM	Text Document	1 KB
 0018.2003-12-18.GP.spam.txt	2/29/2020 9:35 AM	Text Document	4 KB

### Spam List



	doc	bow	label	compound	neg	neu	pos	treebank_tag	pos_tag
0	ham/0001.1999-12-10.farmer.ham.txt	[christmas, tree, farm, pictures]	ham	0.0000	0.000	1.000	0.000	[(christmas, NN), (tree, NN), (farm, NN), (pic...	[(christmas, NNS), (tree, VBP), (farm, NN), (p...
1	ham/0002.1999-12-13.farmer.ham.txt	[vastar, resources, gary, production, high, is...	ham	-0.9153	0.066	0.911	0.023	[(vastar, NN), (resources, NNS), (gary, NN), (...	[(vastar, NN), (resources, NNS), (gary, JJ), (...
2	ham/0003.1999-12-14.farmer.ham.txt	[calpine, daily, gas, nomination, calpine, dai...	ham	0.0000	0.000	1.000	0.000	[(calpine, NN), (daily, JJ), (gas, NN), (nomin...	[(calpine, JJ), (daily, JJ), (gas, NN), (nomin...
3	ham/0004.1999-12-14.farmer.ham.txt	[issue, fyi, note, stella, forwarded, stella, ...	ham	0.8689	0.000	0.902	0.098	[(issue, NN), (fyi, NN), (note, NN), (stella, ...	[(issue, NN), (fyi, VBZ), (note, VBP), (stella...
4	ham/0005.1999-12-14.farmer.ham.txt	[meter, 7268, nov, allocation, fyi, forwarded,...	ham	0.5106	0.000	0.945	0.055	[(meter, NN), (7268, NN), (nov, NN), (allocati...	[(meter, NN), (7268, CD), (nov, JJ), (allocati...

**Table 1.1 Bag of Words**

#### Most Informative Features

V_forwarded = True	ham : spam = 193.4 : 1.0
V_hou = True	ham : spam = 182.7 : 1.0
V_prescription = True	spam : ham = 137.4 : 1.0
V_nom = True	ham : spam = 123.7 : 1.0
V_pain = True	spam : ham = 115.7 : 1.0
V_ect = True	ham : spam = 113.3 : 1.0
V_2001 = True	ham : spam = 105.3 : 1.0
V_2005 = True	spam : ham = 95.8 : 1.0
V_health = True	spam : ham = 84.1 : 1.0
V_bob = True	ham : spam = 81.0 : 1.0
V_sex = True	spam : ham = 79.1 : 1.0
V_spam = True	spam : ham = 75.8 : 1.0
V_medications = True	spam : ham = 75.8 : 1.0
V_featured = True	spam : ham = 70.8 : 1.0
V_differ = True	spam : ham = 69.1 : 1.0
V_thousand = True	spam : ham = 64.1 : 1.0
V_creative = True	spam : ham = 64.1 : 1.0
V_713 = True	ham : spam = 61.5 : 1.0
V_subscribers = True	spam : ham = 59.1 : 1.0
V_farmer = True	ham : spam = 56.3 : 1.0
V_alj = True	spam : ham = 54.1 : 1.0
V_cheap = True	spam : ham = 53.5 : 1.0
V_pro = True	spam : ham = 52.5 : 1.0
V_epson = True	spam : ham = 49.1 : 1.0
V_tap = True	ham : spam = 47.6 : 1.0
V_ami = True	ham : spam = 46.6 : 1.0
V_julie = True	ham : spam = 43.1 : 1.0
V_susan = True	ham : spam = 37.8 : 1.0
V_adobe = True	spam : ham = 37.5 : 1.0

**Table 1.2 Most Informative Features**

# Models

In this exercise, models are developed using Naïve Bayes, SVM and Kmeans Clustering to compare their efficiency and accuracy in classifying deception and sentiments on a text document.

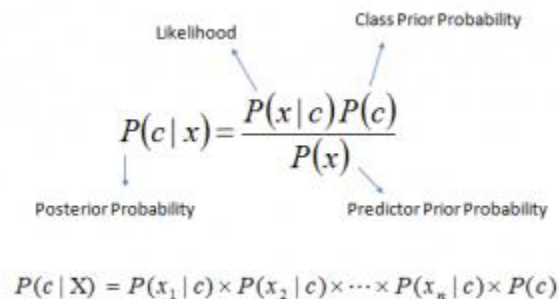
## Naïve Bayes Classification

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 specific feature in a class is unrelated to the presence of any other feature. For example, a fruit may 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'.

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

## Bayes theorem

Bayes theorem provides a way of calculating posterior probability  $P(c|x)$  from  $P(c)$ ,  $P(x)$  and  $P(x|c)$ . Look at the equation below:



The diagram shows the Bayes' Theorem equation with labels pointing to its components. The equation is 
$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$
 Labels: 'Likelihood' points to  $P(x|c)$ , 'Class Prior Probability' points to  $P(c)$ , 'Posterior Probability' points to  $P(c|x)$ , and 'Predictor Prior Probability' points to  $P(x)$ .

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

- $P(c|x)$  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.

## Classification based on conditional probability



To classify whether players will play or not based on weather condition using Naïve Bayes classification approach

Likelihood table Frequency Table are derived by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Table 2.1

Using Naive Bayesian equation, the posterior probability for each class is calculated. The class with the highest posterior probability is the outcome of prediction.

Say if we want to find out if the Players will play when the weather is sunny?

To solve the above discussed method of posterior probability.

$$P(\text{Yes} | \text{Sunny}) = P(\text{Sunny} | \text{Yes}) * P(\text{Yes}) / P(\text{Sunny})$$

Here we have  $P(\text{Sunny} | \text{Yes}) = 3/9 = 0.33$ ,  $P(\text{Sunny}) = 5/14 = 0.36$ ,  $P(\text{Yes}) = 9/14 = 0.64$

Now,  $P(\text{Yes} | \text{Sunny}) = 0.33 * 0.64 / 0.36 = 0.60$ , which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes.

### Model 1.1: Spam Classification using Naïve Bayes with Unigram Features

**Table 2.1** shows the classification report and the confusion matrix of the Naïve Bayes with Unigram Feature set

	Precision	Recall	F1
spam	0.987	0.846	0.911
ham	0.921	0.994	0.956

	h	s
	a	p
	m	a
		m
ham	<63.8%>	5.5%
spam	0.4%	<30.3%>

(row = reference; col = test)

**Table 2.1 Classification report and Confusion Matrix**

### Model 1.2: Spam Classification using Naïve Bayes with Bigram Features

**Table 2.2** shows the classification report and the confusion matrix of the Naïve Bayes with Bigram Feature set

	Precision	Recall	F1
spam	0.987	0.846	0.911
ham	0.921	0.994	0.956

	h	s
	a	p
	m	a
	m	m
ham	<63.8%>	5.5%
spam	0.4%	<30.3%>

(row = reference; col = test)

**Table 2.2 Classification report and Confusion Matrix**

### Model 1.3: Spam Classification using Naïve Bayes with POS Features

**Table 2.3** shows the classification report and the confusion matrix of the Naïve Bayes with POS Feature set

	Precision	Recall	F1
spam	0.990	0.854	0.917
ham	0.925	0.995	0.959

	h	s
	a	p
	m	a
	m	m
ham	<64.1%>	5.2%
spam	0.3%	<30.4%>

(row = reference; col = test)

**Table 2.3 Classification report and Confusion Matrix**

### Model 1.4: Spam Classification using Naïve Bayes with SL Features

**Table 2.4** shows the classification report and the confusion matrix of the Naïve Bayes with SL Feature set

	Precision	Recall	F1
spam	0.997	0.848	0.916
ham	0.921	0.998	0.958

			s	
		h	p	
		a	a	
		m	m	
-----+				
ham		<63.8%>	5.5%	
spam		0.1%	<30.6%>	
-----+				
(row = reference; col = test)				

**Table 2.4 Classification report and Confusion Matrix**

#### Model 1.5: Spam Classification using Naïve Bayes with Negative Features

**Table 2.5** shows the classification report and the confusion matrix of the Naïve Bayes with Negative Feature set

	Precision	Recall	F1
spam	1.000	0.834	0.910
ham	0.912	1.000	0.954

			s	
		h	p	
		a	a	
		m	m	
-----+				
ham		<63.2%>	6.1%	
spam		.	<30.7%>	
-----+				
(row = reference; col = test)				

**Table 2.5 Classification report and Confusion Matrix**

### Model 1.6: Spam Classification using Naïve Bayes with All Features

**Table 2.6** shows the classification report and the confusion matrix of the Naïve Bayes with All Feature set

	Precision	Recall	F1
spam	0.993	0.845	0.913
ham	0.919	0.997	0.956

			s	
		h	p	
		a	a	
		m	m	
-----+				
ham		<63.7%>	5.6%	
spam		0.2%	<30.5%>	
-----+				
(row = reference; col = test)				

**Table 2.6 Classification report and Confusion Matrix**

# Results

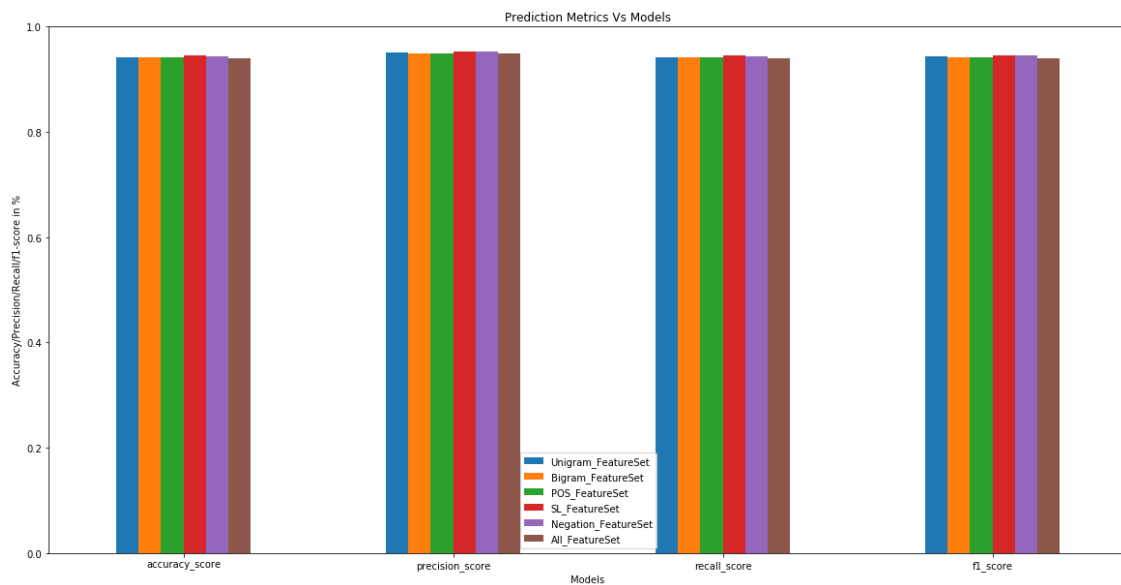
Accuracy, precision and recall using Naive Bayes are tabulated in **Table 3.1** for spam detection

	model	accuracy_score	precision_score	recall_score	f1_score
5	All Feature set	0.942	0.950208	0.942	0.943169
0	Unigram Feature set	0.941	0.948517	0.941	0.942135
1	Bigram Feature set	0.941	0.948517	0.941	0.942135
2	POS Feature set	0.945	0.951929	0.945	0.946025
3	SL Feature set	0.944	0.952143	0.944	0.945128
4	Negation Feature set	0.939	0.949111	0.939	0.940352

**Table 3.1 Model performance comparison for Spam detection**

After analyzing various feature set to detect spams, it is observed that POS feature set is outperforming when compared to other feature sets but with a low margin of 0.01 difference.

Please refer **Figure 3.1** which compares the accuracy of different models and its accuracy, precision and recalls.



**Figure 3.1 Prediction Metrics comparison for Spam detection**

The above inference shows that POS Feature set is performing better than other feature set models but with only marginal difference.