The Classification of Spam vs Ham Emails

Using Naive Bayes and Token Vectorization

Abstract

- Spam email filtering is very important because it avoids tons of unnecessary, fraudulent, and dangerous emails from getting to our emails and only showing us ham (regular) emails. With this, there is a huge amount of time saved because we aren't sorting through mail that often.

- We decided to try to see if we could create a classification algorithm (Naive Bayes) to be able to accurately classify whether an email was either spam or ham.

Data

- We obtained our data through Kaggle.
  - There are a total of **5,728 emails.**
    - 4,360 ham emails
    - 1,368 spam emails
  - **Two columns**
    - email (the only feature)
    - Classifier (0 = Ham, 1 = Spam)

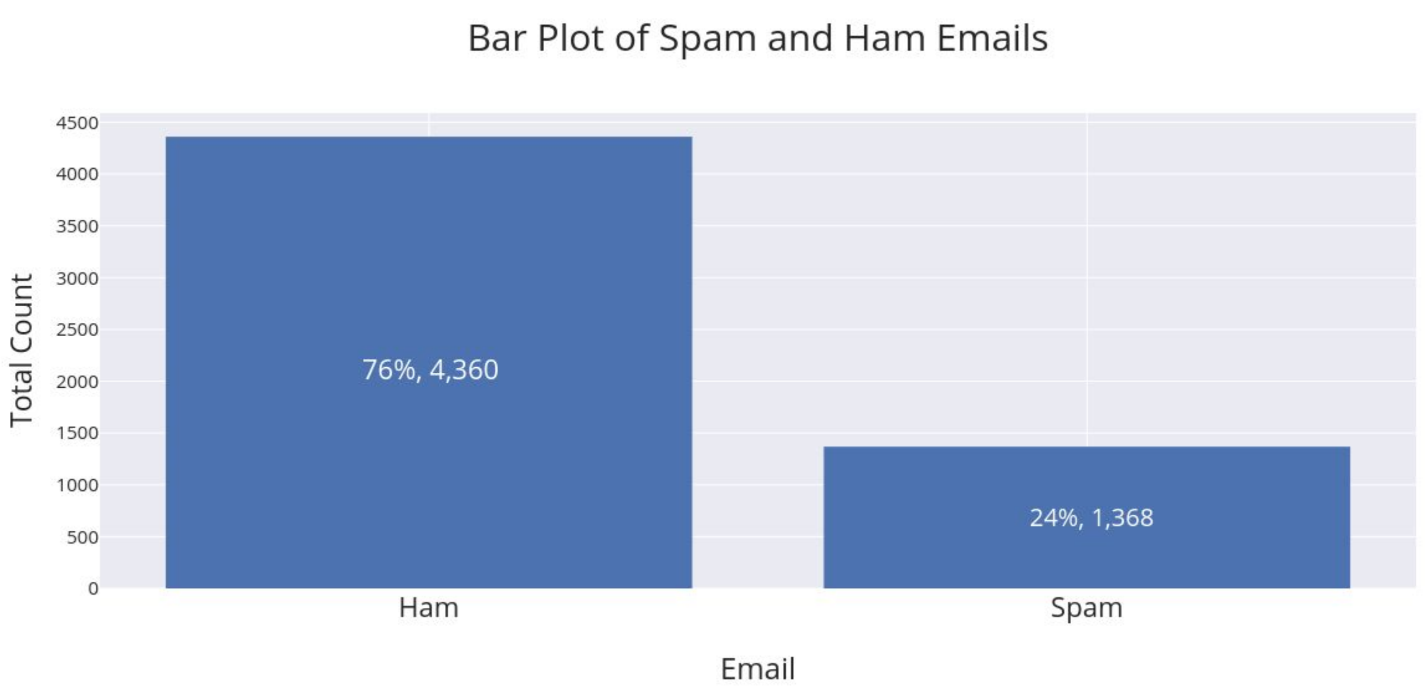
- **20 of our personal emails** for real world testing.

Data Preprocessing

- **Token Vectorization:**

	Nigerian	me	help	your	you	Hey	needs	are	Jon	out	Prince	able	to
Nigerian Prince needs your help!	1	0	1	1	0	0	1	0	0	0	0	0	0
Hey Jon, are you able to help me out?	0	0	1	0	1	1	0	1	1	1	0	1	1

  - **Total Emails**: 5,728, **Total Words**: 37,303
- **Create a 20% Ratio Split:** 20% for Testing, 80% for Training



Naive Bayes

- **Remember Naive Bayes?**

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- **P(spam | TokenVectorizer) =**
  - $\frac{\text{P(spam) * P(TokenVectorizer | spam)}}{\text{P(TokenVectorizer)}}$

- **P(ham | TokenVectorizer) =**
  - $\frac{\text{P(ham) * P(TokenVectorizer | ham)}}{\text{P(TokenVectorizer)}}$

Results

Confusion Matrix:

Dataset:

854	10
4	278

Personal:

3	4
5	8

ROC_AUC:

- Dataset: 0.987
- Personal: 0.521

- Our dataset is able to accurately classifity 99% of the test emails.
- The model on our personal email is able to accurately classify 56% of times.
- Our model looks good on paper but is not performing all that well on our personal emails out in the real world.
- The classifier can accurately classify ham emails but is classifying targeted emails as spam.

Improvement:

- There needs to be other features such as sender email, images, emojis detection, where you mark a sender as not spam and more.
- A bigger dataset, with millions of emails.

Hardik Dhanak – Thomas Nguyen – Shyam Patel – Frank Chambergo – Gian Elim

INST 414: Data Science Techniques – Final Project – Fall 2019

The Classification of Spam vs Ham Emails

Using Naive Bayes and Token Vectorization

Abstract

- Spam email filtering is very important because it avoids tons of unnecessary, fraudulent, and dangerous emails from getting to our emails and only showing us ham (regular) emails. With this, there is a huge amount of time saved because we aren't sorting through mail that often.

- We decided to try to see if we could create a classification algorithm (Naive Bayes) to be able to accurately classify whether an email was either spam or ham.

Data

- We obtained our data through Kaggle.
  - There are a total of 5,728 total emails
    - 4,360 ham emails
    - 1,368 spam emails
  - Two columns
    - email (the only feature)
    - Classifier (0 = Ham, 1 = Spam)

- 20 of our personal emails for real world testing.

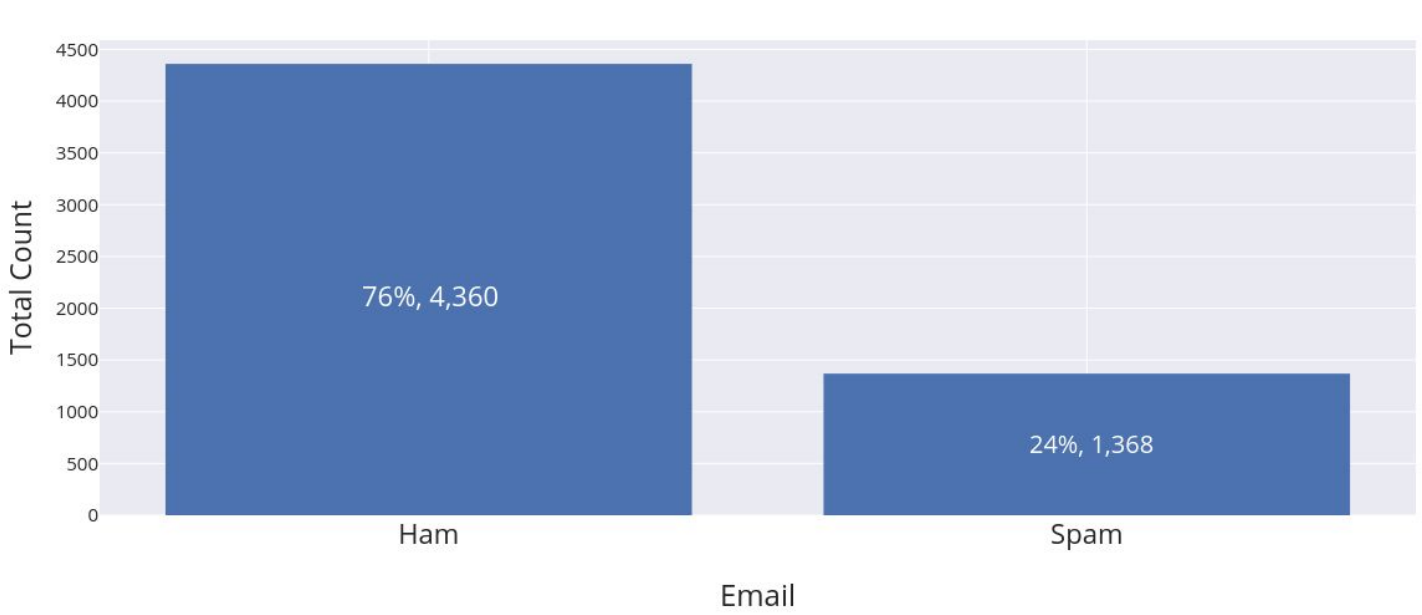
Data Preprocessing

- **Token Vectorization:**

	Nigerian	me	help	your	you	Hey	needs	are	Jon	out	Prince	able	to
Nigerian Prince needs your help!	1	0	1	1	0	0	1	0	0	0	0	0	0
Hey Jon, are you able to help me out?	0	0	1	0	1	1	0	1	1	1	0	1	1

  - **Total Emails**: 5,728, **Total Words**: 37,303
- **Create a 20% Ratio Split:** 20% for Testing, 80% for Training

Bar Plot of Spam and Ham Emails



Naive Bayes

- **Remember Naive Bayes?**

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

- **P(spam | TokenVectorizer) =**
  - $\frac{\text{P(spam) * P(TokenVectorizer | spam)}}{\text{P(TokenVectorizer)}}$

- **P(ham | TokenVectorizer) =**
  - $\frac{\text{P(ham) * P(TokenVectorizer | ham)}}{\text{P(TokenVectorizer)}}$

Results

Dataset	**Precision**	**Recall**	**F1-Score**
Ham	1.00	0.99	0.99
Spam	0.97	0.99	0.98
Accuracy	0.98	0.99	0.98
Weight Avg	0.99	0.99	0.99

Pretty good right? But how does it do in the real world?

Personal	Precision	Recall	F1-Score
Ham	0.38	0.43	0.40
Spam	0.67	0.62	0.64
Accuracy	0.52	0.52	0.55
Weight Avg	0.56	0.55	0.56

- **Confusion Matrix:**

Dataset:

854	10
4	278

Personal:

3	4
5	8

- **ROC_AUC:**
  - Dataset: 0.987
  - Personal: 0.521

Hardik Dhanak – Thomas Nguyen – Shyam Patel – Frank Chambergo – Gian Elim
INST 414: Data Science Techniques – Final Project – Fall 2019