INST 414
Hardik Dhanah, Thomas Nguyen, Shyam Patel, Frank Chambergo, Gian Elim

Final Paper

Introduction:

Spam filtering is a must right now because we are in a data-driven world and there are constantly spam emails coming in that need to be filtered out to make ensure one's ability to work quickly and always have access to the right emails without having to search through many spam ones to find the one ham email. Spam filtering is one of the most valuable tools across all ages with a device. Since Technology has developed such at a significant rate, college students, businesses, and users with internet access who have personal emails are a major impact. Spam filter makes everyone's life easier because on average there are approximately 14.5 billion spam emails sent per day. Filtering out these spam emails to obtain your own email with whom you signed up for is better than getting unknown emails.

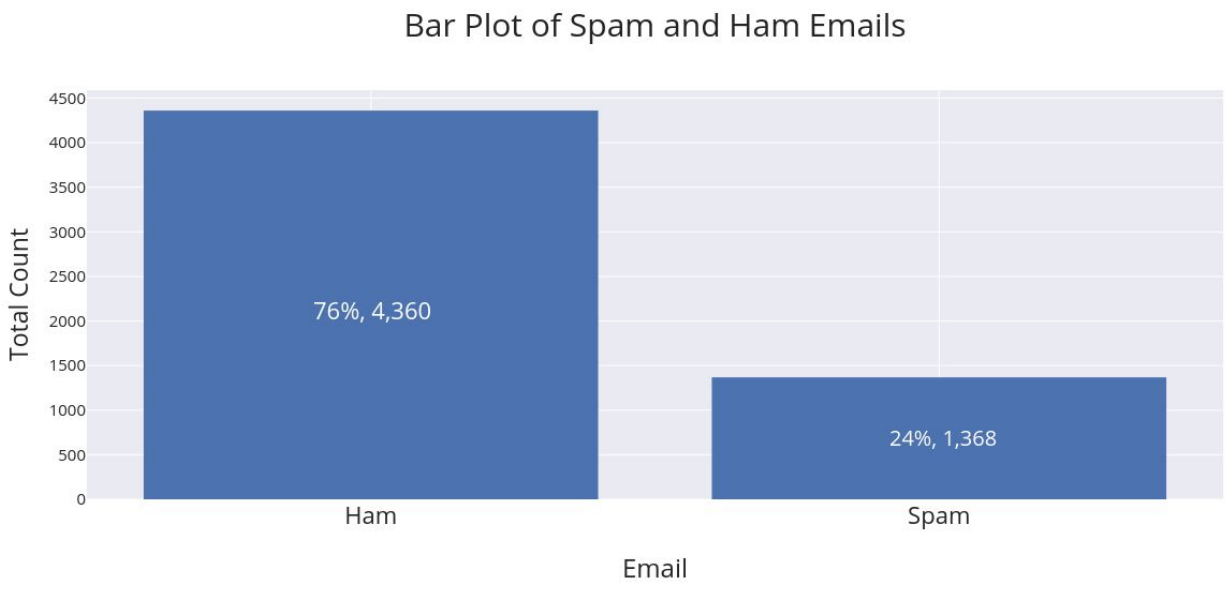
Use of Machine Learning:

We thought Machine Learning would be great for such spam filtering because instead of having us manually mark each individual email as spam, we could use a machine learning algorithm, specifically a classification algorithm. With a classification algorithm, we can train it with current emails and whether or not they are spam or not. Then with this training, it will go out and try to predict with an email it sees is spam or ham in accordance with its training.

Data:

We converted this problem into a machine learning problem by finding a dataset through Kaggle that gave us 5,728 emails of which 4,360 were ham emails and 1,368 were spam emails. There were only two columns, one just the email and the second one as the classifier, 0 being ham and 1 being spam. In addition, we used 20 of our personal emails to test out how our algorithm worked in the real world. The data collection was straight forward for us because we were able to find a dataset that had simply the email's text contents and whether or not it was spam or ham.

Visualization of the total count of Ham and Spam emails.



Data Preprocessing

To get a working machine learning algorithm, we first needed to get the data preprocessed, and we didn't have to clean the data, though we had to token vectorize it. By token vectorizing, each word is given its own number value and then is represented as the following:

- **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

What is going on here is that each word from each sentence is given its own column and then an array is created to see if that sentence has those following words. 0 is no it does not have that word and 1 is that it does. "Nigerian Prince needs your help!" becomes "1,01,1,0,1,0,0,0,0,0,0".

Model Creation:

To create our model, we used the Naive Bayes classification algorithm. Naive Bayes works well with spam classification and token vectorization. The following formula represents

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

the probability of A given B = Probability of B given A * probably of A all divided by probably of B. We would then translate this, over to our formulas as the following:

P(spam | TokenVectorizer) =

- $\frac{P(spam) * P(TokenVectorizer \mid spam)}{P(TokenVectorizer)}$

P(ham | TokenVectorizer) =

- $\frac{P(ham) * P(TokenVectorizer \mid ham)}{P(TokenVectorizer)}$

With this, we were able to then run our Naive Bayes model. What the model would do is, whether or not an email is Spam or Ham given that those specific words from an array in token vectorization.

We created a 20% Ratio Split: 20% for Testing data, 80% for Training data.

Results:

Training Model (Results from Test Model):

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	0.99	0.99	0.99

Confusing Matrix:

854	10
4	278

ROC AUC: 0.987

Testing Trained Model on Personal Emails:

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	0.56	0.55	0.56

Confusing Matrix:

3	4
5	8

ROC AUC: 0.521

As a result, our training model accurately predicted spam/ham emails 99% of the time. Although our training model has high accuracy, when we tested our trained model on our personal emails, the model wasn't able to perform as well it did on the testing set. It only was able to accurately predict whether the email was spam/ham 56% of the time. The confusing matrix shows that, for the testing set, it was accurately able to predict 1,132/1,416 and only misclassified 14/1,416 while on our personal emails it was able to only predict 11/20 and misclassified 9/20. The ROC AUC was .987 for the testing set and .521 for personal emails. Being than .987 is close to 1, the testing set was a good classifier while for the personal emails it wasn't.

Essentially, our model looked good on the test set, but it wasn't able to perform as accurately as it was on the test set. This is because the emails that it was trained on was only 5,728 and spam filters in the modern world have billions of emails. More regular emails receive marketing emails that this specific classifier is saying that are spam when they aren't simply because that's how it's trained. Though, if it was trained on that only, we don't think it would still work well because the text is not the only feature modern spam filters use. They use email address, pictures, emojis as well as others to classify them, whereas we are using just text of the emails.