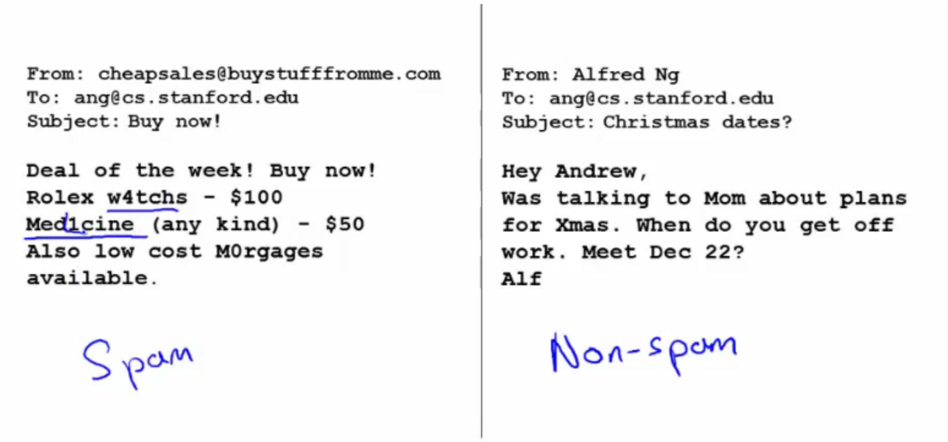
Spam Classifier

This section teaches how to put a machine learning system together. Suppose we have an email dataset, we would like to distinguish between spam and non-spam classes. For this purpose, we need to create a vector for each email. Normally vectors contains around 10,000 to 50,000 entities gathered by most common worlds used in spam emails. Once the vectors are ready, we can start classifying the emails to spam or ham (non-spam) email. Imagine these emails below:



```
From: cheapsales@buystufffromme.com
To: ang@cs.stanford.edu
Subject: Buy now!

Deal of the week! Buy now!
Rolex w4tchs - $100
Med1cine (any kind) - $50
Also low cost M0rgages
available.
```

Spam

```
From: Alfred Ng
To: ang@cs.stanford.edu
Subject: Christmas dates?

Hey Andrew,
Was talking to Mom about plans
for Xmas. When do you get off
work. Meet Dec 22?
Alf
```

Non-spam

Usually in spam emails, words are misspelled (for example medicine is written as med1cin). Here is how we represents features in email in order to classify them to class 0 (non-spam) or class 1 (spam emails):

- Choose hundreds of words that represent of email being spam and put them in a long vector
    - Like misspelled word
    - Using email header
    - Spammers usually obscure origin of email
    - Unusual routes emails
    - Is Discount similar to discounts in spam emails?!
- See how many of these worlds appear in email and define a feature vector x (if the word repeats more than once, we only consider it once)

$$x = \left[\begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 1 \\ \vdots \end{array}\right] \begin{array}{l} \text{andrew} \\ \text{buy} \\ \text{deal} \\ \text{discont} \\ \vdots \\ \text{now} \\ \vdots \end{array}$$

Error Analysis

You may come up with lots of ideas in machine learning problem. Best approach is to:

- Spend first 24 hours to develop an initial simple algorithm
- Implement and test it
- Plot learning curve to see how can you improve the performance
- Manually examine few samples and see why the algorithm did not work
    - For example, in spam classifier you may get 100 wrong in 500 examples. Go through those email and see what are you missing
- Evaluate your algorithm. A single number (accuracy) can tell us how well the algorithm is doing. We need to do error analysis on cross validation set, not test set