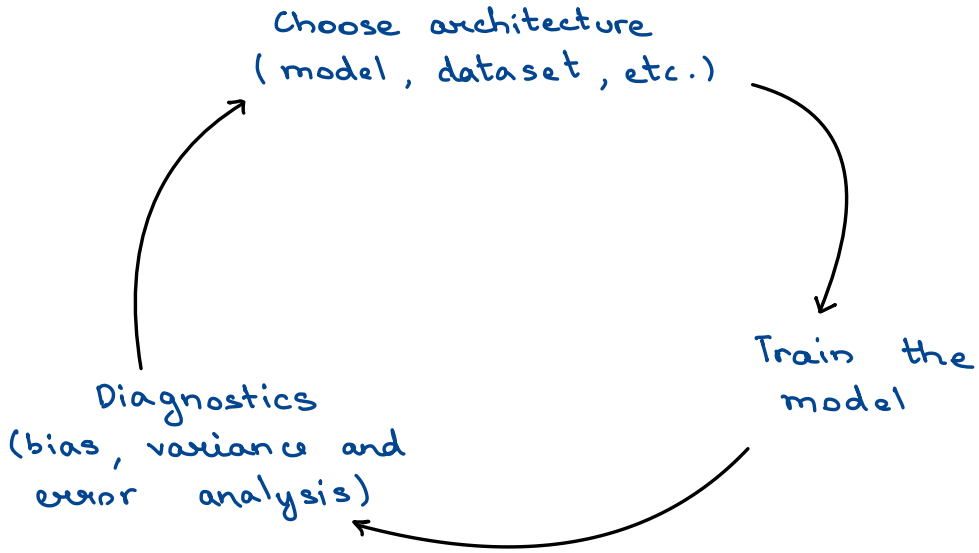
This loop represents what a process of ML Development feels like :-



In most cases, it will take multiple iterations of the loop to get the optimum performance.

Let's again look at the email spam classifier.

From: cheapsales@buystufffromme.com
To: Andrew Ng
Subject: Buy now!

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

deliberate misspellings

spam

From: Alfred Ng
To: Andrew Ng
Subject: Christmas dates?

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

not spam

First we decide the architecture:

It could be a supervised learning model

$\vec{x}$ = features of email
y = spam (1) or not spam (0)

basically we're listing many english words and checking if they are in the email and classify them using 1s and 0s.

We'll list approx. 10,000 words from the dictionary and compute $x_1, x_2, x_3, \ldots, x_{10,000}$

We'll check if any of those 10,000 words appear in our mail.

$$\vec{x} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ \\ \\ \\ \vdots \end{bmatrix} \begin{matrix} a \\ andrew \\ buy \\ deal \\ discount \\ \\ \\ \end{matrix}$$

There are more different ways to construct a feature vector.

We could even count the amt. of times these words have appeared, but 1s and 0s for indication works just fine.

How to try to reduce your spam classifier's error?

- Collect more data. E.g., "Honeypot" project.
- Develop sophisticated features based on email routing (from email header). ②
- Define sophisticated features from email body. E.g., should "discounting" and "discount" be treated as the same word.
- Design algorithms to detect misspellings. E.g., w4tches, med1cine, m0rtgage.

③

→ More data can be collected by many different methods. One such method that works for email spam is that we can create multiple fake email ids and purposefully get scammers to email us. Once they email those fake ids, we get to know more about how a spam email works.

② Email routing checks how an email has travelled across different servers around the globe to reach the designated email address using email headers. Sometimes the path that an email has taken may tell us if its from a spammer or not.

③ Spending time on writing sophisticated code and designing algorithms for certain misclassifications should only be done based on error analysis.

- Pharma 21
- Deliberate misspellings (w4tches, med1cine) 3
- Unusual email routings 7
- Steal passwords (phishing) 18
- Spam messages embedded in images 5

Since pharma and phishing have the most errors, we have to write better code and address these errors instead of focusing on deliberate misspellings because fixing those might not have a lot of impact.