E.g. Build a spam classifier:

How to spend your time to make it have lower error?

– collect lots of data

– Develop sophisticated features based on email routing information.

– Develop sophisticated features for message body.

– Develop sophisticated algorithm to detect misspellings.

**When start build a learning algorithm or working on a ML, Recommended approach:**

– Start with a simple algorithm that you can implement quickly. Implement it and test it on your cross-validation data.

– Plot learning curves to decide if more data, more features, etc. are likely to help.

– Error analysis: Manually examine the examples(in cross validation set) that your algorithm made errors on. See if you spot any systematic trend in what type of examples it is making error on.

Error analysis may not be a helpful for deciding if this is likely to improve performance. Only solution is to try it and see if it works.

**Skewed Classes**

It means that the number of examples of one class is much more greater than others class.

Error metric of skewed class:

y = 1 in presence of rare class that we want to detect(y = 1 is the class that have less examples)

Precision/recall

|  |  |  |  |
| --- | --- | --- | --- |
|  | Actual class | | |
| Predicted  class |  | 1 | 0 |
| 1 | True positive | False positive |
| 0 | False negative | True negative |

Precision = true positive / (true positive + false positive)

recall = true positive / (true positive + false negative)

How to compare precision/recall numbers?

F1 Score(F Score):

2 \* (P\*R) / (P + R)

Using Large Data Sets

Use a learning algorithm with many parameters(that could low bias)

Use a very large training set(might be unlikely to overfit.In other words, it could low variance)