# Entry 2 - Define the Process

In Entry 1, I decided to take the smallest problem I can find and solve just one aspect. Only after I’ve solved that single problem am I allowed to move on to the next. Jason Brownlee of [Machine Learning Mastery](https://machinelearningmastery.com/start-here/) recommends small projects that take [no more than 5-15 hours](https://machinelearningmastery.com/self-study-machine-learning-projects/) from inception to presentation of the results. He also points out that with a good process in place, a specific problem can be addressed in [1-2 hours](https://machinelearningmastery.com/practice-machine-learning-with-small-in-memory-datasets-from-the-uci-machine-learning-repository/).

After pouring hours at a time into self-study, debugging, reserach, troubleshooting, and a host of other things, 1-2 hours to solve a specific problem sounds pretty good.

The first problem to solve is naturally **how do I pick what to work on first**?

## The Problem

There are a lot of skills involved in doing data science. To name a few:

**Mathematics**

* Statistics
* Probability
* Linear Algebra
* Calculus

**Databases**

* Infrastructure
* SQL
* Entity Relationship Modeling
* Normalization

**Technology**

* Version Control
* Virtual Environments/Containers
* Command Line Interface
* Repositories

**Programming**

* Principals and theory
* Specific language (Python or R)
* Data structures
* Algorithms
* Regex

**Data Handling**

* Analysis
* Mining
* Wrangling/Cleaning
* Feature Engineering
* Time Series

**Specialty Areas**

* Graph
* Natural Language Processing
* Computer Vision
* Behavior Analysis
* Entity Recognition

**Storytelling**

* Communication
* Visualization

**Big Data**

* Cloud Computing
* Streaming vs Batch

**Domain Knowledge**

* Industry Processes
* Opportunities & Challenges

Most of the fundamental skills like proramming and math are interrelated with the higher level skills like visualization and algorithms. **How do I detangle those skills, work my way back to the beginning, and find a logical place to start?**

If I start with too high a skill, like building a predictive model, I’ll spend more than the recommended 5-15 hours, let alone the target goal of 1-2 hours, working through the requirements to slap together even a basic solution. For example, when creating a model, first you have to get the data (which comes with it’s own set of pitfalls). Then that data has to be preprocessed:

* What to do with missing values?
  + Most algorithms choke if there’s no value for a specific feature of a specific observation
  + Others, like decision trees, could care less
* What about categorical variables?
  + Most algorithms prefer numeric values
  + The fact that you have seven different kinds of mushroom mean nothing to them, but if you just change them into numbers the algorithm makes assumptions like button mushrooms come before shiitake just like 3 comes before 8
* How do you figure out which of your nearly 600 features are actually useful?
  + Plugging them all in could cause issues with [collinearity](http://www.stat.tamu.edu/~hart/652/collinear.pdf)
  + Or your computer could chug away forever trying to make use of all 600
* Which algorithm do you pick?
  + Scikit-learn lists [17 different kinds of algorithms](https://scikit-learn.org/stable/supervised_learning.html), many of which have multiple implimentations
  + Algorithms can be resource prohibitive, so running every one of them under the sun is unrealistic
* How do you measure success?

## 

## The Failure

Obviously, I’m getting ahead of myself again. Missing values, variable types, measuring success: these are all important issues to address. They should be considered fundamentals when it comes to building a model - not the generic fundamentals of solving a machine learning problem.

## The Options

Jason makes an eloquent argument for choosing or developing a [systematic process](https://machinelearningmastery.com/process-for-working-through-machine-learning-problems/). Having spent hours upon hours working through data in a one-off manner, streamlining some of these decisions and processes sounds like a stellar idea.

Having a clear process gives direction on what to do, when to do it, and what comes next. It also provides a framework for incorporating [deliberate practice](https://jamesclear.com/deliberate-practice-theory) into the pipeline; concentrating on one single aspect to improve. This will allow me to generate the [quantity](https://jamesclear.com/repetitions) of work that should produce better quality as antidoctially suggested in [Art & Fear](https://www.amazon.com/gp/product/0961454733/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=0961454733&linkCode=as2&tag=jamesclear-20&linkId=CYEZ57AX7IODGHWX).

The suggestion was based on an experiment conducted by a photography professor at the University of Florida who varied the way one of his classes was graded. One half was graded on the quantity of photos they produced: 100 photos = A, 90 photos = B, etc. The other half was graded on quality: only one photo had to be submitted, but it had to be as close to perfect as possible. The quantity group outperformed the quality group because they spent the semester trying things out and learning by doing. The quality group got so focused on what a good photo constituted that they didn’t actually take many photographs. Christopher Baus would call these two groups the [doers and the talkers](https://baus.net/doersandtalkers/).

I’ve spent quite enough time in the talker/theory group. Time to become a doer.

## The Proposed Solution

Jason recommends a [5 step process](https://machinelearningmastery.com/process-for-working-through-machine-learning-problems/) on his Machine Learning Mastery website. This doesn’t seem quite detailed enough for me.

At the 2019 ODSC East conference, Andreas Muller did a [four part series](https://github.com/amueller/ml-workshop-1-of-4) on Machine Learning with Scikit Learn. In his introduction, he recommended checking out [Hands on Machine Learning with Scikit-Learn & TensorFlow](https://www.amazon.com/Hands-Machine-Learning-Scikit-Learn-TensorFlow/dp/1491962291) by Aurelien Geron. I own the book (yes the first edition, because that’s the one I bought during my graduate studies and I own quite enough data science books already thank you very much) and this remembered recommendation prompted me to see what Aurelien had to say.

Chapter Two outlines an 8 step process, which better aligns with my experience of what’s needed for a machine learning solution.

The 8 steps are:

1. Frame the problem and look at the big picture
2. Get the data
3. Explore the data to gain insights
4. Prepare the data to better expose the underlying data patterns to Machine Learning algorithms
5. Explore many different models and short-list the best ones
6. Fine-tune the models and combine them into a great solution
7. Present the solution
8. Launch, monitor, and maintain the system

### Added bonus

The added bonus of using these steps is that each one helps pick the next mini-challenge to tackle.

## Next up

Frame the Problem.