**Becoming a Machine Learning Engineer | Step 3: Pick Your Tool**

As engineers, we spend our careers learning to use new and better tools so that we can build products that bring real value. The tools we use can change often, but they all serve a function and have use cases. Even though machine learning in its modern form is very new, as an industry goes, there is a wide variety of tools at an engineer’s disposal. At the end of this article, you should have a greater understanding of some available tools and whether any of them is right for you.

**Main Tools**

**WEKA | Waikato Environment for Knowledge Analysis**

WEKA is a modern machine learning workbench with many great features every ML engineer will need to explore data and apply algorithms. All of this great functionality without having to write a single line of code. Whether you are a programmer or not, I recommend trying out different problems with WEKA.

I also use it a lot when familiarizing myself with new data sets, which should be in everyone's problem solving process [(Becoming a Machine Learning Engineer | Step 2: Pick a process)](https://medium.com/towards-data-science/becoming-a-machine-learning-engineer-step-2-pick-a-process-942eef6ba8dd). The built in algorithms, Graphing tools, easy data import, and the easy to use GUI allow me to explore data much faster than writing a quick script in python.

[**WEKA Download**](https://www.cs.waikato.ac.nz/ml/weka/downloading.html)**|**[**Great WEKA tutorial**](https://www.ibm.com/developerworks/library/os-weka1/index.html)

**Python + Libraries**

Python is an accessible programming language and the fastest growing right now regarding users, documentation, and libraries. With amazing libraries such as NumPy, SciPy, Tensorflow, Pandas, Flask, and much more, you can just about do anything you want with python.

The only downside to python is that can be slower than other languages when you don’t leverage libraries efficiently. Take for example matrices and NumPy, you should very rarely ever have to use loops to modify values and if you must then keep them to an absolute minimum.

[**University of Michigan Python Coursera Course**](https://www.coursera.org/learn/python)

**R**

R is a workhorse for statistical analysis and by extension machine learning. It is a platform to understand and explore your data using statistical methods and graphs. On top of that is has a large number of machine learning algorithms, and advanced implementations written by the developers of the algorithm.

It is great tools for all developers, but if you’re new to machine learning, I would suggest going deep with python. Not only can you develop on offs but you can also bring your project to production, which is not possible using just R.

[**Johns Hopkins R Coursera Course**](https://www.coursera.org/learn/r-programming)

**Other great tools**

**Matlab/Octave**

MATLAB/Octave is excellent for representing and working with matrices. These softwares are very popular in universities, and it is easy for non-programmers to get in to, but you can ignore this if you plan to practice machine learning in the real world.

[Matlab Download](https://www.mathworks.com/downloads/) | [Octave Download](https://www.gnu.org/software/octave/)

**C**

In production environments, it is common to see algorithms prototyped in R or python and then implemented in C for execution speed and system reliability. If you plan to get serious about implementing machine learning algorithms it might make sense for you to focus on C.

[C tutorial](http://www.learn-c.org/)

**What tools should you use?**

Python and WEKA :)