# Prerequisites and Prework

### Is Machine Learning Crash Course right for you?

I have little or no machine learning background.

I have some background in machine learning, but I'd like a more current and complete understanding.

I know machine learning really well, but I know little or nothing about TensorFlow.

Please read through the following Prerequisites (#prerequisites) and Prework (#prework) sections before beginning Machine Learning Crash Course, to ensure you are prepared to complete all the modules.

## Prerequisites

Machine Learning Crash Course does not presume or require any prior knowledge in machine learning. However, to understand the concepts presented and complete the exercises, we recommend that students meet the following prerequisites:

- **Mastery of intro-level algebra.** You should be comfortable with variables and coefficients, linear equations, graphs of functions, and histograms. (Familiarity with more advanced math concepts such as logarithms and derivatives is helpful, but not required.)
- **Proficiency in programming basics, and some experience coding in Python.** Programming exercises in Machine Learning Crash Course are coded in Python (https://www.python.org/) using TensorFlow (https://www.tensorflow.org/). No prior experience with TensorFlow is required, but you should feel comfortable reading and writing Python code that contains basic programming constructs, such as function definitions/invocations, lists and dicts, loops, and conditional expressions.

See the Key Concepts and Tools (#key-concepts) section below for a detailed list of math and programmi
ots used in Machine Learning Crash Course, with reference materials for each.

# Prework

mming exercises run directly in your browser (no setup required!) using the Colaboratory
://colab.research.google.com) platform. Colaboratory is supported on most major browsers, and is most
ghly tested on desktop versions of Chrome and Firefox. If you'd prefer to download and run the exercises
, see these instructions (/machine-learning/crash-course/running-exercises-locally) for setting up a local
nment.

## Problem Framing

If you're new to machine learning, we recommend starting your journey by taking Introduction to Machine Learning Problem Framing (https://developers.google.com/machine-learning/problem-framing/). This one-hour course teaches you how to identify appropriate problems for machine learning.

## Getting Started with pandas

The programming exercises in Machine Learning Crash Course use the pandas (http://pandas.pydata.org/) library for manipulating data sets. If you're unfamiliar with pandas, we recommend completing the Quick Introduction to pandas (https://colab.research.google.com/notebooks/mlcc/intro_to_pandas.ipynb?utm_source=mlcc&utm_campaign=colab-external&utm_medium=referral&utm_content=pandas-colab&hl=en) tutorial, which illustrates the key pandas features used in the exercises.

# Key Concepts and Tools

Machine Learning Crash Course discusses and applies the following concepts and tools. For more information, see the linked resources.

## Math

### Algebra

- Variables (https://www.khanacademy.org/math/algebra/introduction-to-algebra/alg1-intro-to-variables/v/what-is-a-variable)
, coefficients (https://www.khanacademy.org/math/cc-sixth-grade-math/cc-6th-equivalent-exp/cc-6th-parts-of-expressions/v/expression-terms-factors-and-coefficients)
, and functions (https://www.khanacademy.org/math/algebra/algebra-functions)
- Linear equations (https://wikipedia.org/wiki/Linear_equation) such as $y = b + w_1x_1 + w_2x_2$
- Logarithms (https://wikipedia.org/wiki/Logarithm), and logarithmic equations such as $y = ln(1 + e^z)$
- Sigmoid function (https://wikipedia.org/wiki/Sigmoid_function)

### Linear algebra

- Tensor and tensor rank (https://www.tensorflow.org/programmers_guide/tensors)
- Matrix multiplication (https://wikipedia.org/wiki/Matrix_multiplication)

### Trigonometry

- Tanh (https://reference.wolfram.com/language/ref/Tanh.html) (discussed as an activation function (https://developers.google.com/machine-learning/glossary#activation_function); no prior knowledge needed)

### Statistics

- Mean, median, outliers (https://www.khanacademy.org/math/probability/data-distributions-a1/summarizing-center-distributions/v/mean-median-and-mode)
, and standard deviation (https://wikipedia.org/wiki/Standard_deviation)
- Ability to read a histogram (https://wikipedia.org/wiki/Histogram)

### Calculus (*optional, for advanced topics*)

- Concept of a derivative (https://wikipedia.org/wiki/Derivative) (you won't have to actually calculate derivatives)

- Gradient (https://www.khanacademy.org/math/multivariable-calculus/multivariable-derivatives/gradient-and-directional-derivatives/v/gradient) or slope

- Partial derivatives (https://wikipedia.org/wiki/Partial_derivative) (which are closely related to gradients)

- Chain rule (https://wikipedia.org/wiki/Chain_rule) (for a full understanding of the backpropagation algorithm (https://developers.google.com/machine-learning/crash-course/backprop-scroll/) for training neural networks)

## Python Programming

### Basic Python

The following Python basics are covered in The Python Tutorial (https://docs.python.org/3/tutorial/):

- Defining and calling functions (https://docs.python.org/3/tutorial/controlflow.html#defining-functions), using positional and keyword (https://docs.python.org/3/tutorial/controlflow.html#keyword-arguments) parameters

- Dictionaries (https://docs.python.org/3/tutorial/datastructures.html#dictionaries), lists (https://docs.python.org/3/tutorial/introduction.html#lists), sets (https://docs.python.org/3/tutorial/datastructures.html#sets) (creating, accessing, and iterating)

- `for` loops (https://docs.python.org/3/tutorial/controlflow.html#for-statements), `for` loops with multiple iterator variables (e.g., `for a, b in [(1,2), (3,4)]`)

- `if/else` conditional blocks (https://docs.python.org/3/tutorial/controlflow.html#if-statements) and conditional expressions (https://docs.python.org/2.5/whatsnew/pep-308.html)

- String formatting (https://docs.python.org/3/tutorial/inputoutput.html#old-string-formatting) (e.g., `'%.2f' % 3.14`)

- Variables, assignment, basic data types (https://docs.python.org/3/tutorial/introduction.html#using-python-as-a-calculator) (`int`, `float`, `bool`, `str`)

- The `pass` statement (https://docs.python.org/3/tutorial/controlflow.html#pass-statements)

### Intermediate Python

The following more advanced Python features are also covered in The Python Tutorial (https://docs.python.org/3/tutorial/):

- List comprehensions (https://docs.python.org/3/tutorial/datastructures.html#list-comprehensions)
- Lambda functions (https://docs.python.org/3/tutorial/controlflow.html#lambda-expressions)

## Third-Party Python Libraries

Machine Learning Crash Course code examples use the following features from third-party libraries. Prior familiarity with these libraries is not necessary; you can look up what you need to know when you need it.

### Matplotlib (http://matplotlib.org/contents.html) (for data visualization)

- `pyplot` (http://matplotlib.org/api/pyplot_api.html) module
- `cm` (http://matplotlib.org/api/cm_api.html) module
- `gridspec` (http://matplotlib.org/api/gridspec_api.html) module

### Seaborn (http://seaborn.pydata.org/index.html) (for heatmaps)

- `heatmap` (http://seaborn.pydata.org/generated/seaborn.heatmap.html) function

### pandas (http://pandas.pydata.org/) (for data manipulation)

- `DataFrame` (http://pandas.pydata.org/pandas-docs/stable/dsintro.html#dataframe) class

### NumPy (http://www.numpy.org/) (for low-level math operations)

- `linspace` (https://docs.scipy.org/doc/numpy-1.10.0/reference/generated/numpy.linspace.html) function
- `random` (https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.random.html#numpy.random.random) function
- `array` (https://docs.scipy.org/doc/numpy/reference/generated/numpy.array.html) function

- `arange` (https://docs.scipy.org/doc/numpy/reference/generated/numpy.arange.html) function

**scikit-learn (http://scikit-learn.org/) (for evaluation metrics)**

- metrics (http://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics) module

## Bash Terminal / Cloud Console

To run the programming exercises on your local machine or in a cloud console, you should be comfortable working on the command line:

- Bash Reference Manual (https://tiswww.case.edu/php/chet/bash/bashref.html)
- Bash Cheatsheet (https://github.com/LeCoupa/awesome-cheatsheets/blob/master/languages/bash.sh)
- Learn Shell (http://www.learnshell.org/)

Help Center (https://support.google.com/machinelearningeducation)