# Using Python Logging

1. Get the seceon\_logging.py from seceon-ape/MachineLearning/src/main/python and ensure it is in the same directory as the target file
2. In your file, import seceon\_logging and create a logger object using the getLogger function, with the following parameter options:
   1. log\_name : name of the log, loggers with the same name are considered to be the same object. The default is “machine-learning-python”.
   2. filename : the name of the log file that will be created by the logger. The default is “ML\_python”. Note: The date will be concatenated with whatever name is chosen.
   3. log\_level : The level you want the log to be set. Each level will output all logs of its level and higher and will not log lower levels. The levels from lowest to highest are logging.DEBUG, logging.INFO, logging.WARN, logging.ERROR, and logging.CRITICAL
   4. backupCount : The number of files that should be kept. After the maximum number of files is reached, the oldest log file will automatically be deleted. The default is 1, which means that the current log file and the previous day’s files will be kept.
   5. path : the location the log files should be saved to. The default will be the current directory from wherever the call is made.
3. The logger object should be able to make calls to log using your\_object.debug(“sample text”), your\_object.info(“sample text”), your\_object.warn(“sample text”), your\_object.error(“sample text”), your\_object.critical(“sample text”).
4. To add the logger to functions, just create a parameter for the function where the default value is logger=None. At the beginning of the function include the lines:

if logger is None:

logger = seceon\_logging.getLogger()

This allows for a logger to be passed in from different functions or files while still creating a log if not.

# Logging with the Timer

1. Add the Timer.py file from seceon-ape/MachineLearning/src/main/python to the directory of your target file.
2. In your file, import Timer.py.
3. For whatever function you want to time / log, create your code like this:

with Timer.Timer() as t:

\* Run code here \*

loggingObject.debug(“Code ran. Time: %s“ % (t.secs))

Make sure the log is written outside of the with loop, or the call to get the secs attribute will not work and the code will return an error.