Code	**SUPPORT_VECTOR_MACHINE.PY**
Author	Nathaniel Heatwole, PhD (heatwolen@gmail.com) (GitHub) (LinkedIn)
Summary	Uses a linear support vector machine (SVM) to separate two groups of data, both from scratch and using sklearn. Also fits quadratic and cubic SVMs using sklearn.
Methods/ process	Support vector machine (SVM): - Supervised learning method for generating a boundary between groups of data. - Classification threshold (hyperplane) can be linear or non-linear. - *Support vectors* are the *margin* boundaries. They have the same functional form and slope as and are equidistant about the boundary vector. - Optimization: - *Hard margin*: no points permitted in region between the support vectors (the "demilitarized zone," so to speak), and margin width is maximized. - *Soft margin*: number of points inside the margin and/or how far inside they are located is minimized (for overlapping groups). Steps (from scratch): 1. Randomly select a point from each of the two group in the training data. 2. Compute the slope of the line connecting those two points, and take the negative inverse. This new slope vector is orthogonal to the line connecting the two points, and is therefore an efficient initial guess for the SVM slope.[1] 3. Fit a line with this slope through each point in the training data. 4. Identify the support vectors, using the group-level extremities of the y-intercepts (from the previous step), and considering the relative orientation of the groups. 5. Compute the margin (distance) between these two support vectors. 6. Optimize the slope value (from step #2) to find a local maximum margin width, using parametric variation (for some fixed number of iterations). 7. Repeat this entire process for many different sets of randomly selected points. 8. Choose the SVM parameter set yielding the global maximum margin.
Training data	Randomly generated points (synthetic data) consisting of two groups.
Output	Plots: - Training data - Linear SVM (from scratch) - Linear SVM (sklearn) - Quadratic SVM (sklearn) - Cubic SVM (sklearn) Summary: - SVM parameters (from scratch, sklearn)
Result	The outputs from scratch and using the built-in functionality in sklearn align well.

[1] This is because, rather than *connecting* the data, as a linear fit does, SVMs best *separate* the groups.