



[Course](#) > [Evaluat...](#) > [Confus...](#) > [Choosi...](#)

Choosing The Right Estimator

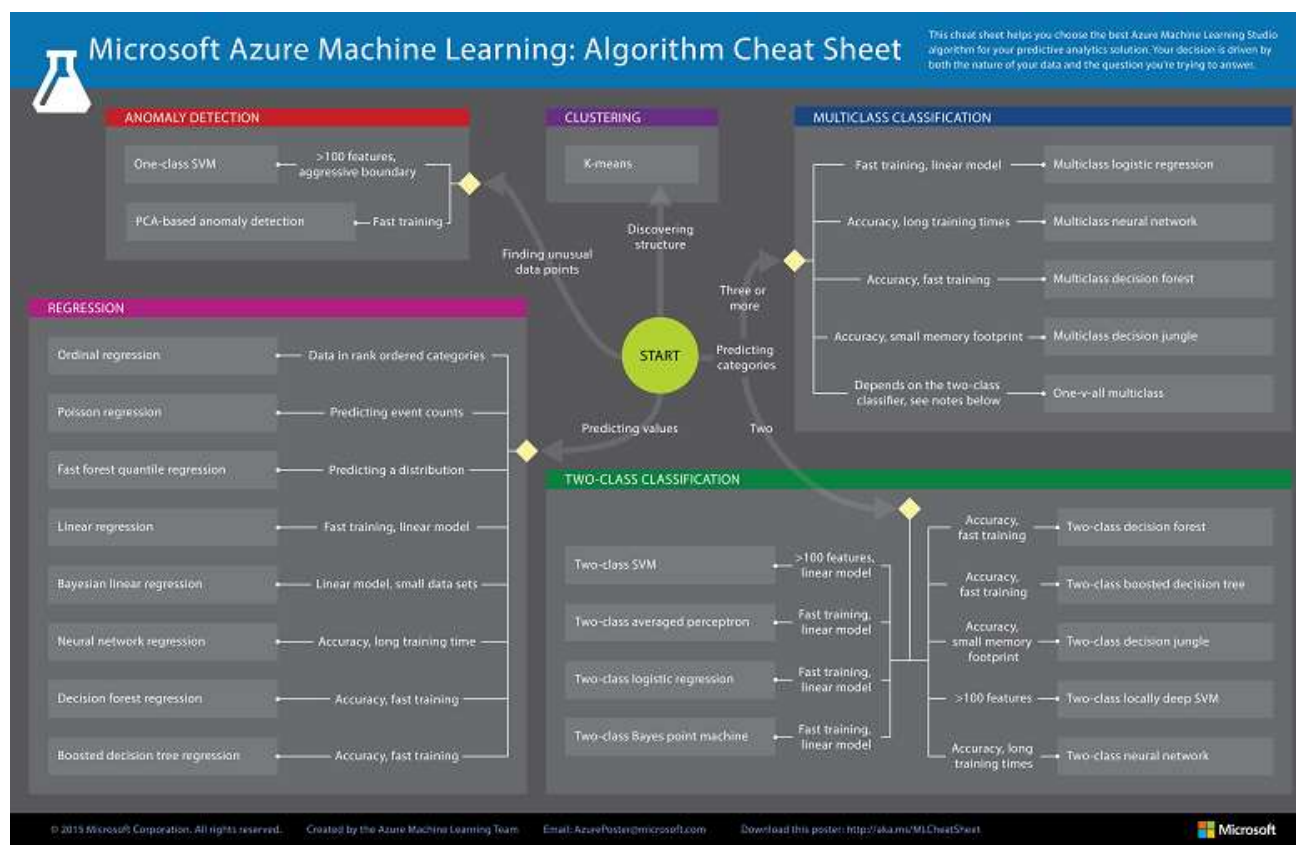
In this course, the terms model, class, estimator, and predictor have been used interchangeably, but never introduced formally. A model is any formula or algorithm designed to represent the mechanics of your data. SciKit-Learn implements many machine learning models as Python language classes, so they can be instantiated and used as objects. To keep the API clean, most of these classes follow a similar paradigm and interface, as a result of inheriting from a single estimator base-class. This is why SciKit-Learn's documentation refers to all of the machine learning methods in their library as estimators. Lastly, when dealing with data that come with attributes you want to learn to predict, such as supervised learning problems, the estimators designed to handle this are called predictors.

Having made it this far in the course, you now have a good foundation regarding the selection of estimators we have covered so far: isometric feature mapping, principle component analysis, k-means, k-nearest neighbors, linear regression, support vector machines, decision trees, and random forest. This is only a taste of what SciKit-Learn has to offer. With so many different categories of machine learning out there, many of which offer solutions to the same types of problems, it can get confusing trying to decide which one is most suitable for your data purposes. Developing an understanding of when to use which algorithm is an important step in your machine learning journey. Typically, algorithm choice is dictated by a balance of factors

- The dimensionality of your data
- The geometric nature of your data
- The types of features used to represent your data
- The number of training samples you have at your disposal
- The required training and prediction speeds needed for your purposes
- The predictive accuracy level desired
- How configurable you need your model to be
- And much more

As you continue to learn the ropes, a few groups have put together visualizations that will guide you to ensuring you've made an informed choice. The first one we'd like to draw your attention to is Microsoft's Azure [Machine Learning Algorithm Cheat Sheet](#) documentation page. Read through the entire article, from the definitions to the comments. This course has been structured around SciKit-Learn, but there are hundreds of other machine learning packages out there. Having exposure to a few of them, and gaining familiarity with the jargon and verbiage they use is beneficial to your future as a data scientist. Moreover, closer to the bottom of the Microsoft Azure page, they also have succinct notes regarding the details of some of the algorithms we've covered and some we haven't, such as logistic regression, boosted trees, neural networks, naive bayes, and other specialized algorithms.

The page also has a [cheat-sheet](#) available for download, a flow-chart that describes the circumstances under which you should use the different machine learning algorithms provided on the Azure platform:



More immediately applicable to this course is a similar [cheat-sheet](#) created by Andreas Mueller, the release manager of, and serial contributor to SciKit-Learn. Rumor has it that this diagram was initially created for laughs before gaining viral acceptance and usage. Since your machine learning experience in this course until now has been based on SciKit-Learn rather than of AzureML, you might find this chart more immediately helpful:

