Classifier vs Estimator

**classifier**: This specifically refers to a type of function (and use of that function) where the response (or range in functional language) is discrete. Compared to this a **regressor** will have a continuous response. Once we may have built a classifier, it is expected to predict for us from within a finite range of classes which class a vector of data is likely to indicate.

Estimator

This isn't a word with a rigorous definition but it usually associated with finding a current value in data. in machine learning it is most frequently used in conjunction with parameter estimation or density estimation. In both cases there is an assumption that data we currently have comes in a form that can be described with a function. With parameter estimation, we believe that the function is a known function that has additional parameters such as rate or mean and we may estimate the value of those parameters. In density estimation we may not even have an assumption about the function but we will attempt to estimate the function regardless. Once we have an estimation we may have at our disposal a model. The estimator then would be the method of generating estimations, for example the method of [maximum likelihood](http://en.wikipedia.org/wiki/Maximum_likelihood).

Bayesian Classification

Naive Bayes classifiers are built on Bayesian classification methods. These rely on Bayes's theorem, which is an equation describing the relationship of conditional probabilities of statistical quantities. In Bayesian classification, we're interested in finding the probability of a label given some observed features, which we can write as P(L | features). Bayes's theorem tells us how to express this in terms of quantities we can compute more directly:

P(L | features)=

All we need now is some model by which we can compute P(features | Li) for each label. Such a model is called a generative model because it specifies the hypothetical random process that generates the data. Specifying this generative model for each label is the main piece of the training of such a Bayesian classifier. The general version of such a training step is a very difficult task, but we can make it simpler through the use of some simplifying assumptions about the form of this model.

This is where the "naive" in "naive Bayes" comes in: if we make very naive assumptions about the generative model for each label, we can find a rough approximation of the generative model for each class, and then proceed with the Bayesian classification. Different types of naive Bayes classifiers rest on different naive assumptions about the data.

## Gaussian Naive Bayes

In this classifier, the assumption is that data from each label is drawn from a simple Gaussian distribution.

One extremely fast way to create a simple model is to assume that the data is described by a Gaussian distribution with no covariance between dimensions. This model can be fit by simply finding the mean and standard deviation of the points within each label, which is all you need to define such a distribution.