# **Naive Bayes Classifier: A Geometric Analysis of the Naivete.**

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), **naive Bayes classifiers** are a family of simple "[probabilistic classifiers](https://en.wikipedia.org/wiki/Probabilistic_classification)" based on applying [Bayes' theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem) with strong (naïve) [independence](https://en.wikipedia.org/wiki/Statistical_independence) assumptions between the features. They are among the simplest [Bayesian network](https://en.wikipedia.org/wiki/Bayesian_network) models. But they could be coupled with [Kernel density estimation](https://en.wikipedia.org/wiki/Kernel_density_estimation) and achieve higher accuracy levels.

Naïve Bayes has been studied extensively since the 1960s. It was introduced (though not under that name) into the [text retrieval](https://en.wikipedia.org/wiki/Information_retrieval) community in the early 1960s, and remains a popular (baseline) method for [text categorization](https://en.wikipedia.org/wiki/Text_categorization), the problem of judging documents as belonging to one category or the other (document categorization)(such as [spam or legitimate](https://en.wikipedia.org/wiki/Spam_filtering), sports or politics, etc.) with [word frequencies](https://en.wikipedia.org/wiki/Bag_of_words) as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine).It also finds application in automatic [medical diagnosis](https://en.wikipedia.org/wiki/Medical_diagnosis).

Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. [Maximum-likelihood](https://en.wikipedia.org/wiki/Maximum-likelihood_estimation) training can be done by evaluating a [closed-form expression](https://en.wikipedia.org/wiki/Closed-form_expression),which takes [linear time](https://en.wikipedia.org/wiki/Linear_time), rather than by expensive [iterative approximation](https://en.wikipedia.org/wiki/Iterative_method) as used for many other types of classifiers.

In the [statistics](https://en.wikipedia.org/wiki/Statistics) and [computer science](https://en.wikipedia.org/wiki/Computer_science) literature, naive Bayes models are known under a variety of names, including **simple Bayes** and **independence Bayes**. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naïve Bayes is not (necessarily) a [Bayesian](https://en.wikipedia.org/wiki/Bayesian_probability) method.

The curse of dimensionality is the bane of all classification problems. What is the curse of dimensionality? As the number of features (dimensions) increase linearly, the amount of training data required for classification increases exponentially. If the classification is determined by a single feature we need a-priori classification data over a range of values for this feature, so we can predict the class of a new data point. For a feature *x* with 100 possible values, the required training data is of order O(100). But if there is a second feature *y* as well that is needed to determine the class, and *y* has 50 possible values, then we will need training data of order O(5000) – i.e. over the grid of possible values for the pair “*x, y*“. Thus the measure of the required data is the volume of the feature space and it increases exponentially as more features are added.

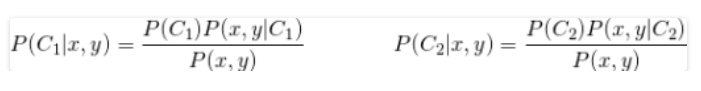
Are there some simplifying assumptions we can make to decrease the amount of data required, even while keeping all the features? In the above example we said we needed training measurements of order O(5000). But naive bayes classifier needs measurements of order only O(150) – i.e. just a linear increase, not an exponential one! That is fantastic but we know there is no free lunch and naive bayes classifier should be making some simplifying (*naive*?) assumptions. That is the purpose of this post – examine the impact of *naivete* in the naive bayes classifier that allows it to side-step the curse of dimensionality. On a practical note, we do not want the algebra to detract us from appreciating what we are after, so we stick to 2 dimensions *x,y* and 2 classes *C1* and *C2*.

**Decision Boundary**

Decision boundary is a curve in our 2-dimensional feature space that separates the two classes *C1* and *C2*. In Figure 1, the zone where *y – f(x) > 0* indicates class *C1* and *y – f(x) < 0* indicates *C2*. Along the decision boundary *y = f(x),* and the probability of belonging to either class is equal. Equal probability of either class is the criterion to obtain the decision boundary.

## **1. Naive Bayes Classification**

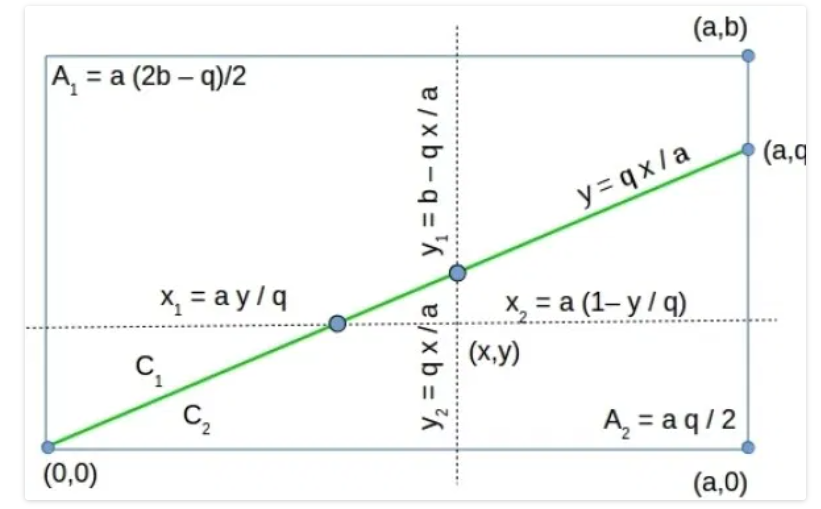
Naive bayes classification is based on [Bayes rule](https://en.wikipedia.org/wiki/Bayes%27_theorem) that relates conditional and marginal probabilities. It is well described in literature so we simply write the equations down for a 2 class (*C1* and *C2*) situation with 2 features *x,y*.



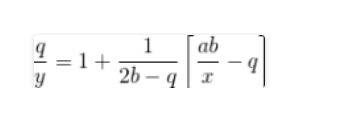
For any new measurement *x,y* we would compute *P(C1|x,y)* and *P(C2|x,y)*, as per Equation [1](http://xplordat.com/2018/08/06/naive-bayes-classifier-a-geometric-analysis-of-the-naivete-part-1/#id2895037934) and pick the class with the larger value. As the denominator is the same it is easier to just compute their ratio so we do not have to evaluate *P(x,y)*.

*P(Ci)* is the probability for any measurement to fall into the class *Ci .* It is computed easily enough from the training data as the relative abundance of *Ci* samples. It is the computation of *P(x,y|Ci)* that is fraught with the challenge of data requirements we talked about. To do it right we need to estimate the joint probability distribution of *x,y* in each class *Ci* and that requires training data on a grid of *x,y* values. That is where the *naive* part of naive bayes comes in to alleviate the curse of dimensionality.

## **2. A Linear Decision Boundary**

We start with the linear case in Figure 4 where a straight line *y = qx/a* separates the two classes in a rectangular feature space. When *q* equals *b* we get balanced classes.

*Evaluating x1 , y1 , x2 and y2 for a linear decision boundary y = q x /a*

The class areas *A1, A2* and the lengths *x1, x2, y1,* and *y2* follow directly from the geometry. . Using them in Equation [7](http://xplordat.com/2018/08/06/naive-bayes-classifier-a-geometric-analysis-of-the-naivete-part-1/#id3826515635) and simplifying we get the decision boundary as predicted by naive bayes to be a hyperbola.

With the closed-form solution in hand for the predicted separation boundary, we can look at the impact of class size and asymptotics thereof when the size of class A1 keeps increasing while A2 stays constant. This is achieved through increasing *b* while holding *a* and *q* constant.