

Chapter 21

Gaussian Naive Bayes Tutorial

Naive Bayes is a simple model that uses probabilities calculated from your training data to make predictions on new data. The basic Naive Bayes algorithm assumes categorical data. A simple extension for real-valued data is called Gaussian Naive Bayes. In this chapter you will discover how to implement Gaussian Naive Bayes from scratch. After reading this chapter you will know:

- The Gaussian Probability Density Function and how to calculate the probability of real values.
- How to learn the properties for a Gaussian Naive Bayes model from your training data.
- How to use a learned Gaussian Naive Bayes model to make predictions on new data.

Let's get started.

21.1 Tutorial Dataset

A simple dataset was contrived for our purposes. It is comprised of two input variables $X1$ and $X2$ and one output variable Y . The input variables are drawn from a Gaussian distribution, which is one assumption made by Gaussian Naive Bayes. The class variable has two values, 0 and 1, therefore the problem is a binary classification problem.

Data from class 0 was drawn randomly from a Gaussian distribution with a standard deviation of 1.0 for $X1$ and $X2$. Data from class 1 was drawn randomly from a Gaussian distribution with a mean of 7.5 for $X1$ and 2.5 for $X2$. This means that the classes are nicely separated if we plot the input data on a scatter plot. The raw dataset is listed below:

X1	X2	Y
3.393533211	2.331273381	0
3.110073483	1.781539638	0
1.343808831	3.368360954	0
3.582294042	4.67917911	0
2.280362439	2.866990263	0
7.423436942	4.696522875	1
5.745051997	3.533989803	1
9.172168622	2.511101045	1
7.792783481	3.424088941	1
7.939820817	0.791637231	1

Listing 21.1: Gaussian Naive Bayes Tutorial Data Set.

You can clearly see the separation of the classes in the plot below. This will make the data relatively easy to work with as we implement and test a Gaussian Naive Bayes model.

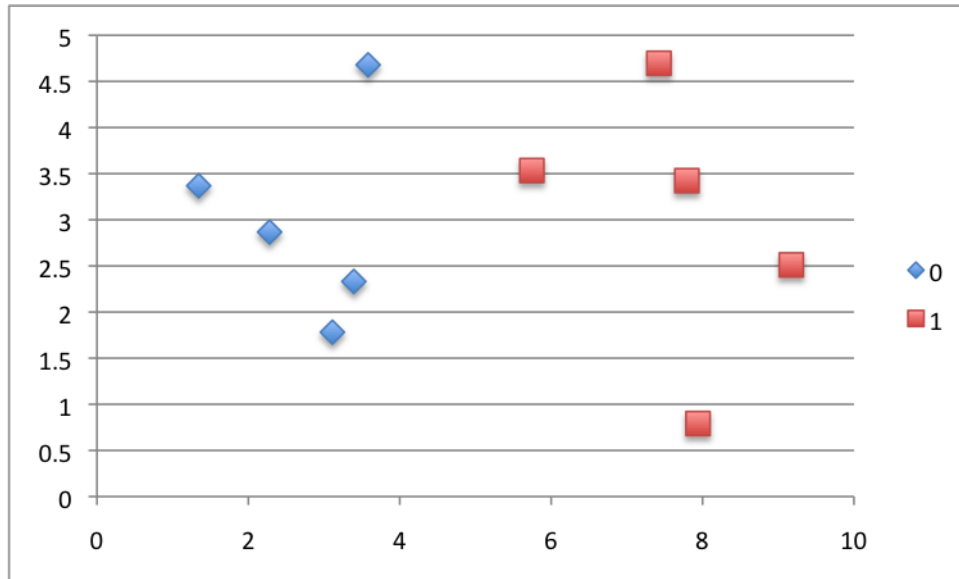


Figure 21.1: Gaussian Naive Bayes Tutorial Dataset.

21.2 Gaussian Probability Density Function

The Gaussian Probability Density Function (PDF) will calculate the probability of a value given the mean and standard deviation of the distribution from which it came. The Gaussian PDF is calculated as follows:

$$pdf(x, mean, sd) = \frac{1}{\sqrt{2 \times \pi \times sd}} \times e^{-\left(\frac{x-mean}{2 \times sd}\right)^2} \tag{21.1}$$

Where $pdf(x)$ is the Gaussian PDF, $mean$ and sd are the mean and standard deviation calculated above, π is the numerical constant PI, e is Euler’s number raised to a power and x is the input value for the input variable. Let’s look at an example. Let’s assume we have real values drawn from a population that has a mean of 0 and a standard deviation of 1. Using the Gaussian PDF we can estimate the likelihood of range of values.

X	PDF(x)
-5	1.48672E-06
-4	0.00013383
-3	0.004431848
-2	0.053990967
-1	0.241970725
0	0.39894228
1	0.241970725
2	0.053990967
3	0.004431848

4	0.00013383
5	1.48672E-06

Listing 21.2: Test of the Gaussian Probability Density Function.

You can see that the mean has the highest probability of nearly 0.4 (40%). You can also see values that are far away from the mean like -5 and +5 (5 standard deviations from the mean) have a very low probability. Below is a plot of the probabilities values.

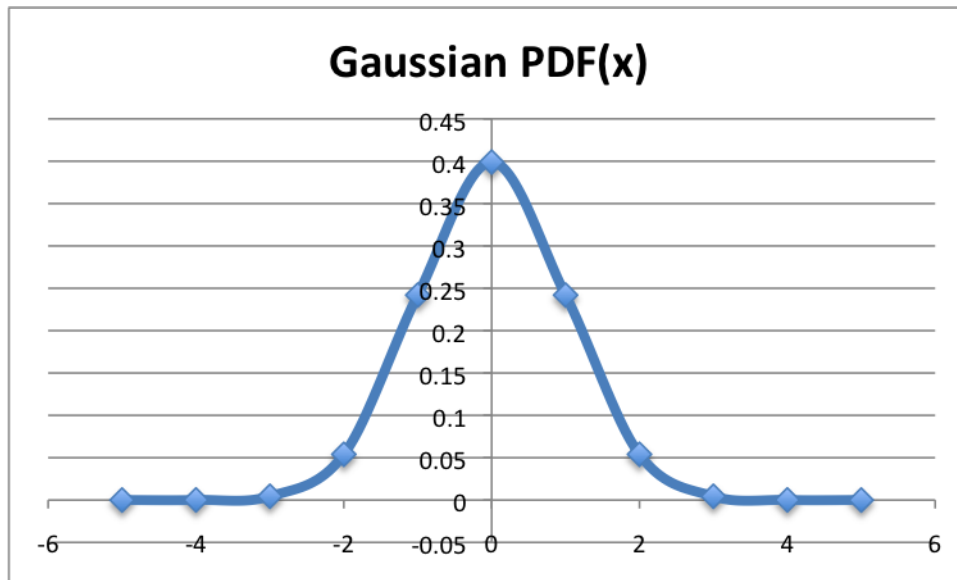


Figure 21.2: Gaussian Probability Density Function.

This function is really useful for Naive Bayes. We can assume that the input variables are each drawn from a Gaussian distribution. By calculating the mean and standard deviation of each input variable from the training data, we can use the Gaussian PDF to estimate the likelihood of each value for each class. We will see how this is used to calculate conditional probabilities in the next section.

21.3 Learn a Gaussian Naive Bayes Model

There are two types of probabilities that we need to summarize from our training data for the naive Bayes model:

- Class Probabilities.
- Conditional Probabilities.

21.3.1 Class Probabilities

The dataset is a two class problem and we already know the probability of each class because we contrived the dataset. Nevertheless, we can calculate the class probabilities for classes 0 and

1 as follows:

$$P(Y = 1) = \frac{\text{count}(Y = 1)}{\text{count}(Y = 0) + \text{count}(Y = 1)}$$

$$P(Y = 0) = \frac{\text{count}(Y = 0)}{\text{count}(Y = 0) + \text{count}(Y = 1)}$$
(21.2)

or

$$P(Y = 1) = \frac{5}{5 + 5}$$

$$P(Y = 0) = \frac{5}{5 + 5}$$
(21.3)

This works out to be a probability of 0.5 (50%) for any given data instance belonging to class 0 or class 1.

21.3.2 Conditional Probabilities

The conditional probabilities are the probabilities of each input value given each class value. The conditional probabilities that need to be collected from the training data are as follows:

$$P(X1|Y = 0)$$

$$P(X1|Y = 1)$$

$$P(X2|Y = 0)$$

$$P(X2|Y = 1)$$
(21.4)

The $X1$ and $X2$ input variables are real values. As such we will model them as having being drawn from a Gaussian distribution. This will allow us to estimate the probability of a given value using the Gaussian PDF described above. The Gaussian PDF requires two parameters in addition to the value for which the probability is being estimated: the mean and the standard deviation. Therefore we must estimate the mean and the standard deviation for each group of conditional probabilities that we require. We can estimate these directly from the dataset. The results are summarized below.

	P(X1 Y=0)	P(X1 Y=1)	P(X2 Y=0)	P(X2 Y=1)
Mean	2.742014401	7.614652372	3.005468669	2.991467979
Stdev	0.926568329	1.234432155	1.107329589	1.454193138

Listing 21.3: Summary of population statistics by class.

We now have enough information to make predictions for the training data or even a new dataset.

21.4 Make Prediction with Gaussian Naive Bayes

We can make predictions using Bayes Theorem, introduced and explained in a previous chapter. We don't need a probability to predict the most likely class for a new data instance. We only need the numerator and the class that gives the largest response is the predicted response.

$$MAP(h) = \max(P(d|h) \times P(h))$$
(21.5)

Let's take the first record from our dataset and use our learned model to predict which class we think it belongs. Instance: $X1 = 3.393533211$, $X2 = 2.331273381$, $Y = 0$. We can plug the probabilities for our model in for both classes and calculate the response. Starting with the response for the output class 0. We multiply the conditional probabilities together and multiply it by the probability of any instance belonging to the class.

$$\begin{aligned} \text{class 0} &= P(\text{pdf}(X1)|\text{class} = 0) \times P(\text{pdf}(X2)|\text{class} = 0) \times P(\text{class} = 0) \\ \text{class 0} &= 0.336255919 \times 0.299321284 \times 0.5 \\ \text{class 0} &= 0.050324277 \end{aligned} \tag{21.6}$$

We can perform the same calculation for class 1:

$$\begin{aligned} \text{class 1} &= P(\text{pdf}(X1)|\text{class} = 1) \times P(\text{pdf}(X2)|\text{class} = 1) \times P(\text{class} = 1) \\ \text{class 1} &= 0.000934051 \times 0.247475201 \times 0.5 \\ \text{class 1} &= 0.000115577 \end{aligned} \tag{21.7}$$

We can see that 0.050324277 is greater than 0.000115577, therefore we predict the class as 0 for this instance, which is correct. Repeating this process for all instances in the dataset we get the following outcomes for class 0 and class 1. By selecting the class with the highest output we make accurate predictions for all instances in the training dataset.

X1	X2	Output Y=0	Output Y=1	Prediction
3.393533211	2.331273381	0.050324277	0.000115577	0
3.110073483	1.781539638	0.038912299	4.02571E-05	0
1.343808831	3.368360954	0.023541958	1.06771E-07	0
3.582294042	4.67917911	0.016404354	0.000108931	0
2.280362439	2.866990263	0.067972749	3.89292E-06	0
7.423436942	4.696522875	6.91958E-08	0.022027299	1
5.745051997	3.533989803	0.000362402	0.013133374	1
9.172168622	2.511101045	2.4463E-12	0.018937265	1
7.792783481	3.424088941	2.54861E-08	0.04197207	1
7.939820817	0.791637231	1.5426E-09	0.013636753	1

Listing 21.4: Predictions using Gaussian Naive Bayes.

The prediction accuracy is 100%, as was expected given the clear separation of the classes.

21.5 Summary

In this chapter you discovered how to implement the Gaussian Naive Bayes classifier from scratch. You learned about:

- The Gaussian Probability Density Function for estimating the probability of any given real value.
- How to estimate the probabilities required by the Naive Bayes model from a training dataset.
- How to use the learned Naive Bayes model to make predictions.

You now know how to implement Gaussian Naive Bayes from scratch for real-valued data. In the next chapter you will discover the k -Nearest Neighbors algorithm for classification and regression.