Continuous Data and Zero Frequency Problem in Naive Bayes Classifier

How to handle it mathematically and conceptually

Tarun Gupta Oct 7, 2020 · 8 min read ★

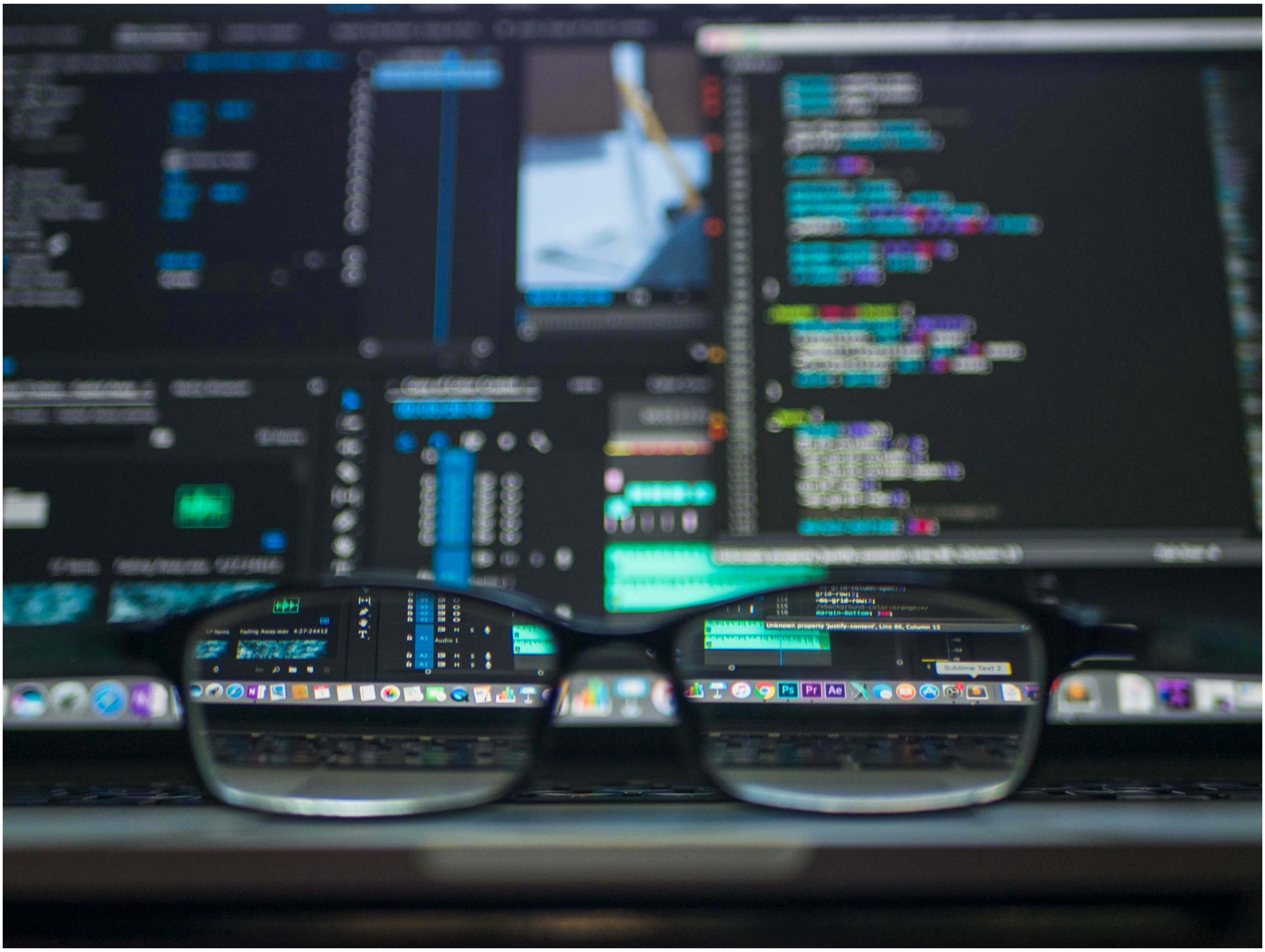


Photo by Kevin Ku on Unsplash

In the context of Supervised Learning (Classification), Naive Bayes or rather Bayesian Learning acts as a gold standard for evaluating other learning algorithms along with acting as a powerful probabilistic modelling technique. But, working with Naive Bayes comes with some challenges.

- It performs well in case of categorical data as compared to numeric data. So, how do we perform classification using Naive Bayes when the data we have is continuous in nature.

- If an instance in test data set has a category that was not present during training then it will assign it “Zero” probability and won’t be able to make prediction. This is known as **Zero frequency problem.** It skews the whole performance of the classification. As a Machine Learning enthusiast, everyone should know how to tackle if the situation arises.

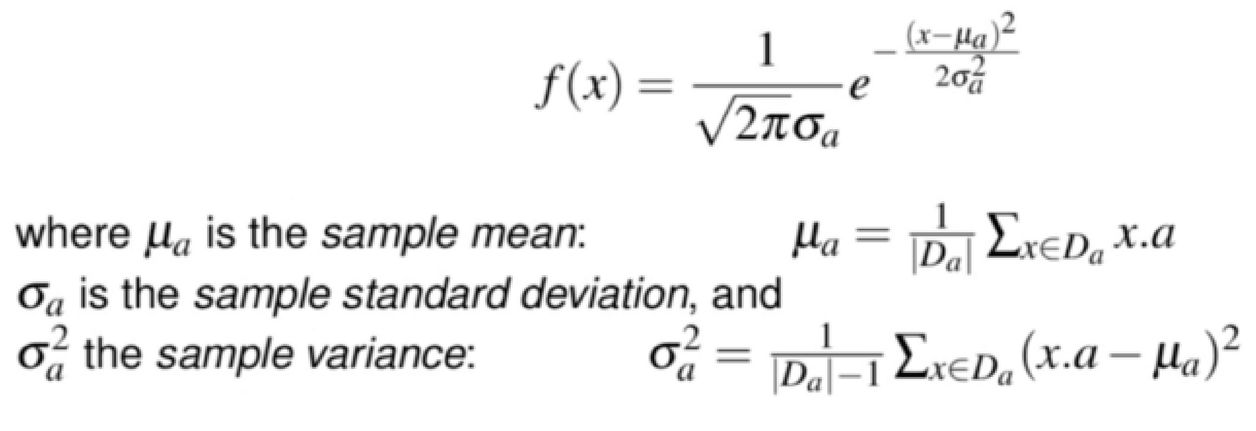
In this post, we are going to discuss the workings of Naive Bayes classifier with Numeric / Continuous Data and the Zero frequency problem, so that it can later be applied to a real world dataset.

There are two ways to estimate the class-conditional probabilities for continuous attributes in naive Bayes classifiers:

- We can discretize each continuous attribute and then replace the continuous attribute value with its corresponding discrete interval. This approach transforms the continuous attributes into ordinal attributes. The conditional probability P(X|Y=*y*), where Y is the target variable is estimated by computing the fraction of training records belonging to class *y* that falls within the corresponding interval for X.

The estimation error depends on the discretisation strategy, as well as the number of discrete intervals. If the number of intervals is too large, there are too few training records in each interval to provide a reliable estimate for P(X|Y). On the other hand, if the number of intervals is too small, then some intervals may aggregate records from different classes and we may miss the correct decision boundary. Hence, there is no rule of thumb on on the discretisation strategy.

- We can assume a certain form of probability distribution for the continuous variable and estimate the parameters of the distribution using the training data. A Gaussian distribution is usually chosen to represent the class-conditional probability for continuous attributes. The distribution is characterized by two parameters, its mean and variance.



$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_a} e^{-\frac{(x-\mu_a)^2}{2\sigma_a^2}}$$

where μ_a is the *sample mean*: $\mu_a = \frac{1}{|D_a|} \sum_{x \in D_a} x.a$

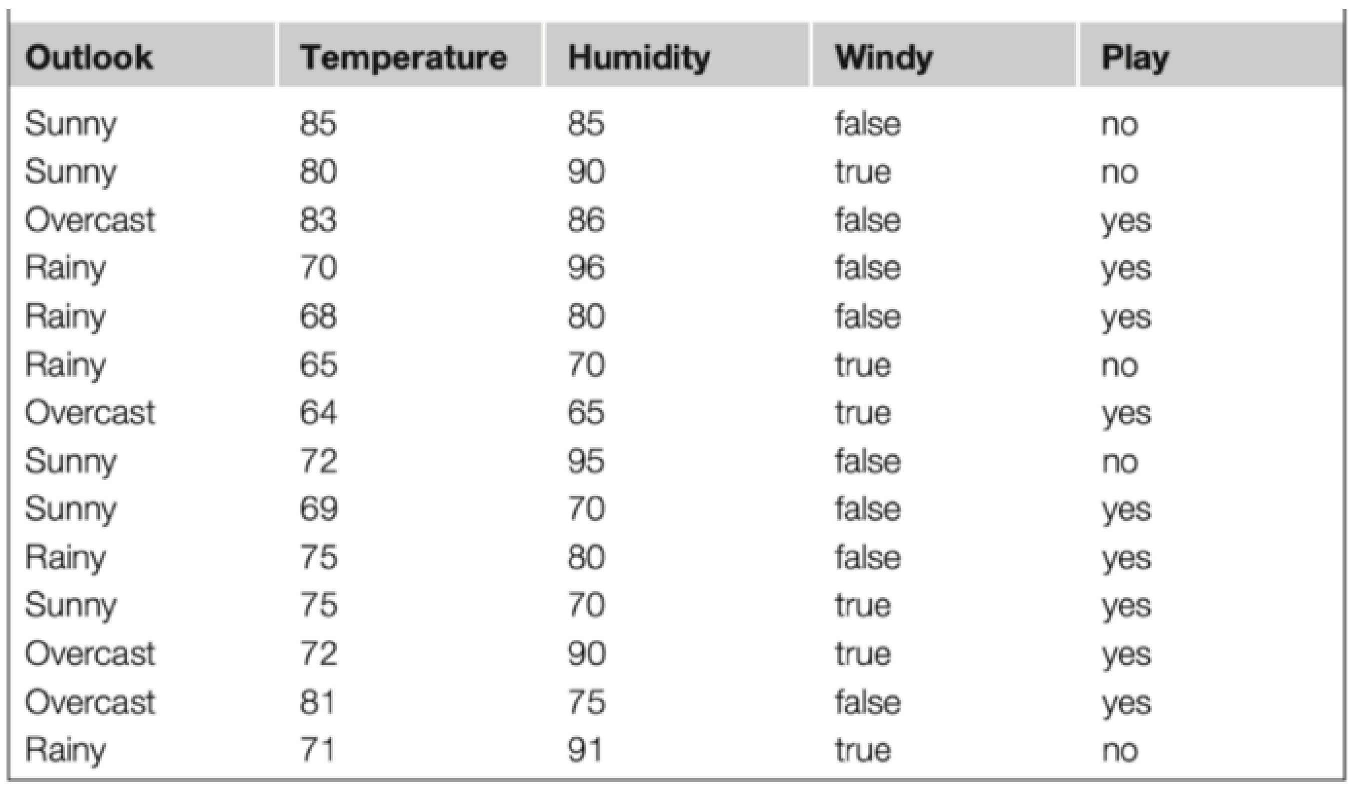
σ_a is the *sample standard deviation*, and

σ_a^2 the *sample variance*: $\sigma_a^2 = \frac{1}{|D_a|-1} \sum_{x \in D_a} (x.a - \mu_a)^2$

Image 1

Now, that we have established the foundation on how to use Gaussian distribution for continuous attributes, let’s see how it can be used a classifier in Machine Learning with an example:

Here is the dataset that we will use:



Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

Data Mining by Witten, Frank and Hall

In this particular dataset, we have a total of 5 attributes. 4 of them are independent variables (Outlook, Temperature, Humidity, Windy) and one is the dependent variable (Play) that we will predict. This is a binary classification problem because the dependent variable is of boolean nature containing either yes or no. The dataset is a mix of ordinal and numerical attributes. Temperature and Humidity being numerical. Outlook and Windy being ordinal attributes.

Since this is non-deterministic or rather probabilistic approach there is no learning for the model.

We are going to classify an instance

x = <Outlook=sunny, Temperature=66, Humidity=90, Windy=True>

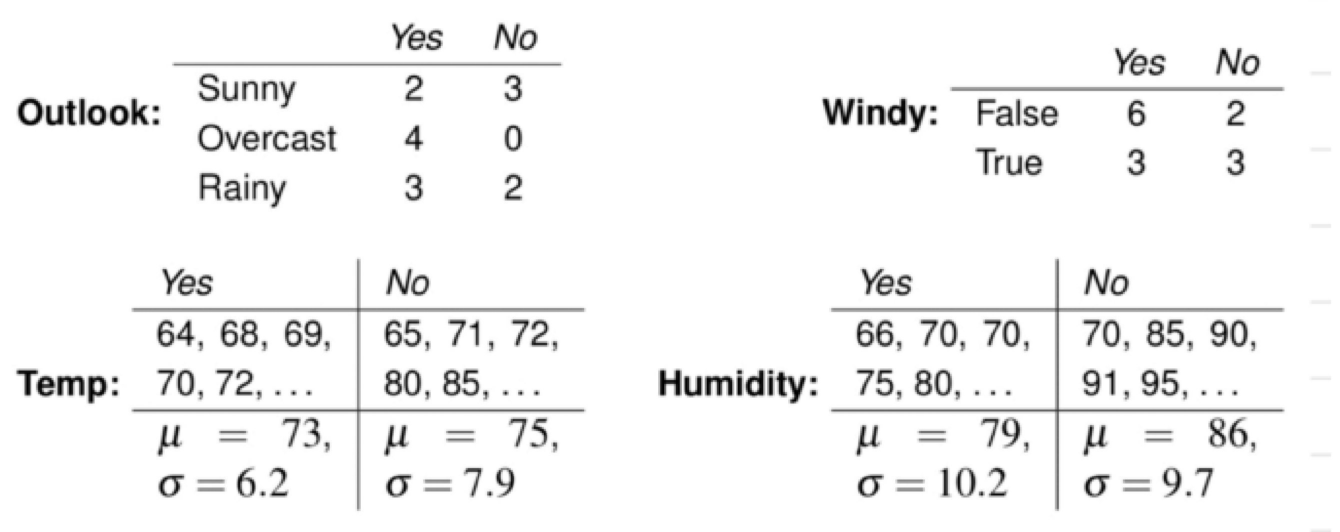
For computing this we need prior probabilities of the target variable *Play*

The total number of instance is 14 and 9 of them have *yes* as value and 5 of them has *no* as value.

p(yes) = 9/14

p(no) = 5/14

In accordance to the target variable, the distribution of independent variable can be written as:



Outlook:		*Yes*	*No*
	Sunny	2	3
	Overcast	4	0
	Rainy	3	2

Windy:		*Yes*	*No*
	False	6	2
	True	3	3

Temp:	*Yes*	*No*
	64, 68, 69, 70, 72, ...	65, 71, 72, 80, 85, ...
	$\mu = 73$, $\sigma = 6.2$	$\mu = 75$, $\sigma = 7.9$

Humidity:	*Yes*	*No*
	66, 70, 70, 75, 80, ...	70, 85, 90, 91, 95, ...
	$\mu = 79$, $\sigma = 10.2$	$\mu = 86$, $\sigma = 9.7$

Image 2

In order to classify the instance *x,* we need to calculate the maximum likelihood for both *play=yes* and *play=no* as follows:

likelihood for *play=yes*

P(*x*/yes) * P(yes) = P(sunny/yes) * P(Temperature=66/yes) * P(Humidity=90/yes) * P(True/yes) * P(yes)

likelihood for *play=no*

P(*x*/no) * P(no) = P(sunny/no) * P(Temperature=66/no) * P(Humidity=90/no) * P(True/no) * P(no)

The attributes individual probabilities are multiplied because of the naive independent assumption.

For the attributes Temperature and Humidity the probability can be computed using the Gaussian distribution formula in **Image 1** by inserting the mean and variance values for the attributes from **Image 2**.

The values needed to calculate the above equations are:

P(sunny/yes) = 2/9

P(Temperature=66/yes) = 0.034

P(Humidity=90/yes) = 0.0221

P(True/yes) = 3/9

and

P(sunny/no) = 3/5

P(Temperature=66/no) = 0.0279

P(Humidity=90/no) = 0.0381

P(True/no) = 3/5

P(*x*/yes) * P(yes) = (2/9) * 0.034 * 0.0221 * (3/9) * (9/14) = 0.000036

P(*x*/no) * P(no) = (3/5) * 0.0279* 0.0381* (3/5) * (5/14) = 0.008137

> *0.008137 > 0.000036*
>
> **Classification — NO**

Now, that we have moved handling of continuous / numeric data in Naive Bayes Classifier, let's dive into how to handle the **Zero Frequency problem.**

It occurs when any condition having **zero** probability in the whole multiplication of the likelihood makes the whole proabability **zero.** In such a case, there is something called **Laplace Estimator** is used.

$$p(x_i|y_i) = \frac{n_c + mp}{n+m}$$

Image 3

where,

nc = number of instances where xi = x and yi = y,

n = number of instances where yi = y,

p = prior estimate, example: when assuming a uniform distribution of attribute values p=1/*m*, with *m* defining the number of different (unique) attribute values.

m = number of unique values for that attribute.

So, if a uniform distribution is assumed the formula in **Image 3** modifies to the following:

$$p(x_i|y_i) = \frac{n_c + 1}{n+m}$$

Image 4

The explanation for the formula in **Image 3** can be a bit difficult to wrap your head around when it is unseen. Let's understand it better with the help of an example:

We are going to classify an instance using the same dataset and distribution in **Image 1 and Image 2.**

x = <Outlook=overcast, Temperature=66, Humidity=90, Windy=True>

For computing this we need prior probabilities of the target variable *Play*

The total number of instance is 14 and 9 of them have *yes* as value and 5 of them has *no* as value.

p(yes) = 9/14

p(no) = 5/14

In accordance to the target variable, the distribution of independent variable can be written as:

In order to classify the instance *x,* we need to calculate the maximum likelihood for both *play=yes* and *play=no* as follows:

likelihood for *play=yes*

P(*x*/yes) * P(yes) = P(overcast/yes) * P(Temperature=66/yes) * P(Humidity=90/yes) * P(True/yes) * P(yes)

likelihood for *play=no*

P(*x*/no) * P(no) = P(overcast/no) * P(Temperature=66/no) * P(Humidity=90/no) * P(True/no) * P(no)

The new values needed to calculate the above equations are:

P(overcast/yes) = 4/9

and

P(overcast/no) = 0/5 = 0

Rest of the values needed to calculate the likelihood are taken from the previous example itself.

P(*x*/yes) * P(yes) = (2/9) * 0.034 * 0.0221 * (3/9) * (9/14) = 0.000036

P(*x*/no) * P(no) = 0 * 0.0279* 0.0381* (3/5) * (5/14) = 0

> *0.000036 > 0*
>
> **Classification — YES**

Here, it can be seen that one conditional probability ***P(overcast/no)*** was the driving factor in classification. Now, let’s see how can we employ the formula for **Laplace Estimator** from **Image 4** under the uniform distribution assumption.

For **Outlook = overcast,** the new probability becomes

P(overcast/yes) = (4 + 3 * (1/3)) / (9 + 3)= 5/12

where,

nc = 4, since 4 instances where Outlook = overcast & play = yes,

n = 9, since total instances where play = yes, m = 3, since the attribute Outlook has 3 unique values (sunny, overcast, rainy),

p = 1/*m* = 1/3, since the uniform distribution is assumed

Similarly,

P(overcast/no) = (0 + 3 * (1/3)) / (5 + 3)= 1/8

where,

nc = 0, since 0 instances where Outlook = overcast & play = no,

n = 5, since total instances where play = no,

m = 3, since the attribute Outlook has 3 unique values (sunny, overcast, rainy),

p = 1/*m* = 1/3, since the uniform distribution is assumed

> Note: While applying Laplace Estimator, ensure that you apply it to all the ordinal attributes. You can't just apply is to the attribute where the Zero frequency problem is occurring.

Since, the other ordinal attribute in our instance to classify is the attribute Windy, we need to apply Laplace Estimator there as well. After applying the modified probabilities are:

For **Windy = True,** the new probability becomes

P(True/yes) = (3 + 2 * (1/2)) / (9 + 2) = 4/11

where,

nc = 3, since 3 instances where Windy = True & play = yes,

n = 9, since total instances where play = yes,

m = 2, since the attribute Windy has 2 unique values (True, False),

p = 1/*m* = 1/2, since the uniform distribution is assumed

Similarly,

P(True/no) = (3 + 2* (1/2)) / (5 + 2)= 4/7

where,

nc = 3, since 3 instances where Windy = True & play = no,

n = 5, since total instances where play = no,

m = 2, since the attribute Windy has 2 unique values (True, False),

p = 1/*m* = 1/2, since the uniform distribution is assumed

P(*x*/yes) * P(yes) = (5/12) * 0.034 * 0.0221 * (4/11) * (9/14) = 0.0000731

P(*x*/no) * P(no) = 1/8 * 0.0279* 0.0381* (4/7) * (5/14) = 0.0000271

> *0.0000731 > 0.0000271*

> **Classification — YES**

Even though the classification did not change but now we have a better scientific reasoning behind our conclusion.

If you enjoy reading stories like these and want to support me as a writer, consider signing up to become a Medium member. It’s $5 a month, giving you unlimited access to stories on Medium. If you sign up using my link, I’ll earn a small commission at no extra cost to you.

Join Medium with my referral link - Tarun Gupta

As a Medium member, a portion of your membership fee goes to writers you read, and you get full access to every story…

tarun-gupta.medium.com

I am giving away a free eBook on Consistency. Get your free eBook here.

Thank you for reading. I hope anyone reading this got handling of Continuous Data and Zero Frequency Problem in Naive Bayes Classifier cleared up. Share if you feel like it can help others. You can read more of my posts here:

Quick Links for Lists of Tagged Stories — Thank You for Visiting

I also have a publication with the goal of fast paced publishing. Read to become a writer.

tarun-gupta.medium.com

Data Science Machine Learning Computer Science Towards Data Science Classification