Introduction to the Naïve Bayes Algorithm

The simplest solutions are usually the most powerful ones, and Naïve Bayes is a good example of that. Despite the advances in Machine Learning in the last years, it has proven to not only be simple but also fast, accurate, and reliable.

It has been successfully used for many purposes, but it works particularly well with natural language processing (NLP) problems.

Naïve Bayes is a probabilistic machine learning algorithm based on the **Bayes Theorem**.

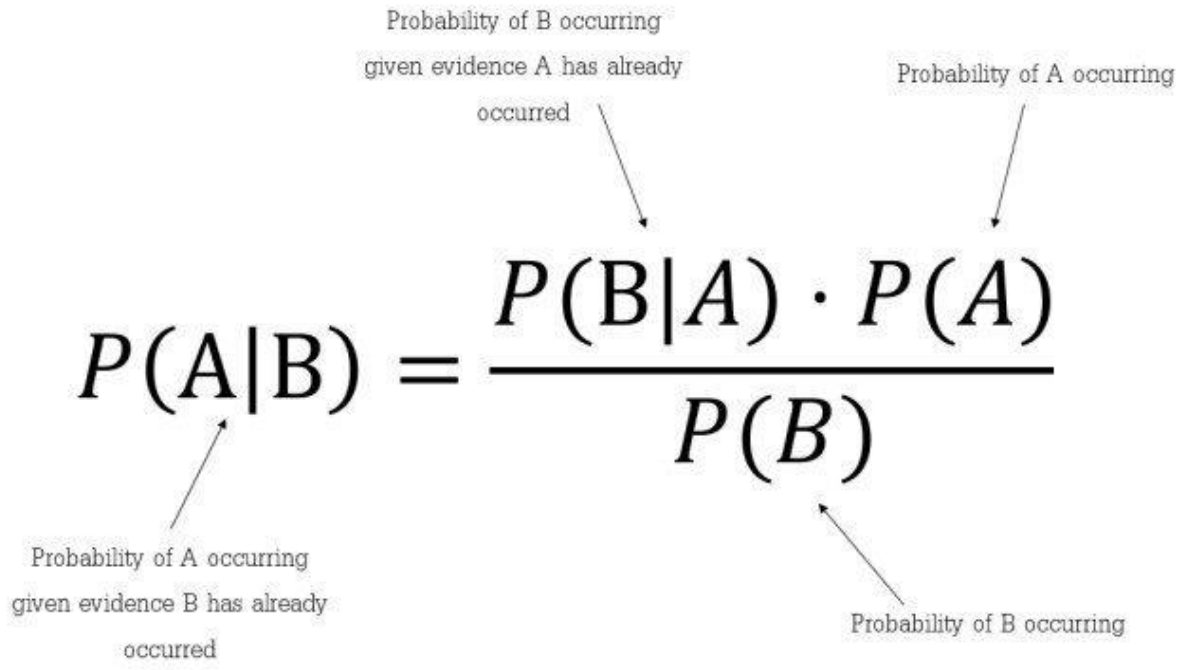
It is a classification technique based on Bayes' Theorem with an independence assumption among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Bayes Theorem

Bayes' Theorem is a simple mathematical formula used for calculating conditional probabilities.

Conditional probability is a measure of the probability of an event occurring given that another event has (by assumption, presumption, assertion, or evidence) occurred.

The formula is: —



Which tells us: how often A happens *given that B happens*, written **P(A|B)** also called posterior probability, When we know: how often B happens *given that A happens*, written **P(B|A)** and how likely A is on its own, written **P(A)** and how likely B is on its own, written **P(B).**

In simpler terms, Bayes' Theorem is a way of finding a probability when we know certain other probabilities.

How Naive Bayes Algorithm works?

Let us understand the working of the Naive Bayes Algorithm using an example. We assume a training data set of weather and the target variable 'Going shopping'. Now we will classify whether a girl will go to shopping based on weather conditions.

The given Data Set is:

Weather	Going Shopping
Sunny	No
Rainy	Yes
Overcast	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Sunny	Yes
Sunny	Yes
Rainy	No

Rainy	Yes
Overcast	Yes
Rainy	No
Overcast	Yes
Sunny	No

The following steps would be performed:

Step 1: Make Frequency Tables Using Data Sets.

Weather	Yes	No
Sunny	3	2
Overcast	4	0
Rainy	2	3

Total	**9**	**5**

Step 2: Make a likelihood table by calculating the probabilities of each weather condition and going shopping.

Weather	Yes	No	Probability
Sunny	3	2	5/14 = 0.36
Overcast	4	0	4/14 = 0.29
Rainy	2	3	5/14 = 0.36
Total	**9**	**5**	
Probability	9/14 = 0.64	5/14 = 0.36	

Step 3: Now, we need to calculate the posterior probability using the Naive Bayes equation for each class.

Problem instance: A girl will go shopping if the weather is overcast. Is this statement correct?

Solution:

- P(Yes|Overcast) = (P(Overcast|Yes) * P (Yes)) / P(Overcast)
- P(Overcast|Yes) = 4/9 = 0.44
- P(Yes) = 9/14 = 0.64
- P(Overcast) = 4/14 = 0.39

Now put all the calculated values in the above formula

- P(Yes|Overcast) = (0.44 * 0.64) / 0.39
- P(Yes|Overcast) = 0.722

Do the same for no

- P(No|Overcast) = (P(Overcast|No) * P (No)) / P(Overcast)
- P(Overcast|No) = 0/9 = 0
- P(No) = 5/14 = 0.36
- P(Overcast) = 4/14 = 0.39

Now put all the calculated values in the above formula

- P(No|Overcast) = (0 * 0.64) / 0.39
- P(N0|Overcast) = 0

The class having the highest probability would be the outcome of the prediction. Using the same approach, probabilities of different classes can be predicted.

Assumptions Made by Naïve Bayes

The fundamental Naïve Bayes assumption is that each feature makes an:

- independent
- equal contribution to the outcome.

Let us take an example to get some better intuition. Consider the car theft problem with attributes Color, Type, Origin, and the target, Stolen can be either Yes or No.

Naïve Bayes Example

The dataset is represented as below.

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Concerning our dataset, the concept of assumptions made by the algorithm can be understood as:

- We assume that no pair of features are dependent. For example, the color being ‘Red’ has nothing to do with the Type or the Origin of the car. Hence, the features are assumed to be **Independent**.
- Secondly, each feature is given the same influence(or importance). For example, knowing the only Color and Type alone can’t predict the outcome perfectly. So none of the

attributes are irrelevant and assumed to be contributing **Equally** to the outcome.

Note: The assumptions made by Naïve Bayes are generally not correct in real-world situations. The independence assumption is never correct but often works well in practice. **Hence the name 'Naï>ve'.**

Here in our dataset, **we need to classify whether the car is stolen, given the features of the car**. The columns represent these features and the rows represent individual entries. If we take the first row of the dataset, we can observe that the car is stolen if the Color is Red, the Type is Sports and Origin is Domestic. So we want to classify a Red Domestic SUV is getting stolen or not. Note that there is no example of a Red Domestic SUV in our data set.

According to this example, Bayes theorem can be rewritten as:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

The variable **y** is the class variable(stolen?), which represents if the car is stolen or not given the conditions. Variable **X** represents the parameters/features.

X is given as,

$$X = (x_1, x_2, x_3, ..., x_n)$$

Here `x1, x2..., xn` represent the features, i.e they can be mapped to Color, Type, and Origin. By substituting for **X** and expanding using the chain rule we get,

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Now, you can obtain the values for each by looking at the dataset and substitute them into the equation. For all entries in the dataset, the denominator does not change, it remains static. Therefore, the denominator can be removed and proportionality can be injected.

$$P(y|x_1, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

In our case, the class variable(**y**) has only two outcomes, yes or no. There could be cases where the classification could be multivariate. Therefore, we have to find the class variable(**y)** with maximum probability.

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

Using the above function, we can obtain the class, given the predictors/features.

The posterior probability **P(y|X)** can be calculated by first, creating a **Frequency Table** for each attribute against the target. Then, molding the frequency tables to **Likelihood Tables** and finally, use the Naïve Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of the prediction. Below are the Frequency and likelihood tables for all three predictors.

Frequency Table

		Stolen?	
		Yes	No
Color	Red	3	2
	Yellow	2	3

Likelihood Table

		Stolen?	
		P(Yes)	P(No)
Color	Red	3/5	2/5
	Yellow	2/5	3/5

Frequency and Likelihood tables of ‘Color’

Frequency Table

		Stolen?	
		Yes	No
Type	Sports	4	2
	SUV	1	3

Likelihood Table

		Stolen?	
		P(Yes)	P(No)
Type	Sports	4/5	2/5
	SUV	1/5	3/5

Frequency and Likelihood tables of ‘Type’

Frequency Table

		Stolen?	
		Yes	No
Origin	Domestic	2	3
	Imported	3	2

Likelihood Table

		Stolen?	
		P(Yes)	P(No)
Origin	Domestic	2/5	3/5
	Imported	3/5	2/5

Frequency and Likelihood tables of ‘Origin’

So in our example, we have 3 predictors **X**.

Color	Type	Origin	Stolen
Red	SUV	Domestic	?

As per the equations discussed above, we can calculate the posterior probability P(Yes | X) as :

$$P(Yes \mid X) = P(Red \mid Yes) * P(SUV \mid Yes) * P(Domestic \mid Yes) * P(Yes)$$

$$= 3/5 * 1/5 * 2/5 * 1$$
$$= 0.048$$

and, P(No | X):

$$P(No \mid X) = P(Red \mid No) * P(SUV \mid No) * P(Domestic \mid No) * P(No)$$

$$= 2/5 * 3/5 * 3/5 * 1$$
$$= 0.144$$

Since 0.144 > 0.048, Which means given the features RED SUV and Domestic, our example gets classified as 'NO' the car is not stolen.

The Zero-Frequency Problem

One of the disadvantages of Naïve-Bayes is that if you have no

occurrences of a class label and a certain attribute value together then the frequency-based probability estimate will be zero. And this will get a zero when all the probabilities are multiplied.
An approach to overcome this 'zero-frequency problem' in a Bayesian environment is to add one to the count for every attribute value-class combination when an attribute value doesn't occur with every class value.

For example, say your training data looked like this:

	Spam = *yes*	Spam = *no*
TimeZone = US	10	5
TimeZone = EU	0	0

P(TimeZone=US|Spam=yes)=10/10=1

P(TimeZone=EU|Spam=yes)=0/10=0

Then you should add one to every value in this table when you're using it to calculate probabilities:

	Spam = yes	Spam = no
TimeZone = US	11	6
TimeZone = EU	1	1

P(TimeZone=US|Spam=yes)=11/12

P(TimeZone=EU|Spam=yes)=1/12

This is how we'll get rid of getting a zero probability.

Types of Naïve Bayes Classifiers

1. Multinomial Naïve Bayes Classifier

Feature vectors represent the frequencies with which certain events have been generated by a **multinomial distribution**. This is the event model typically used for document classification.

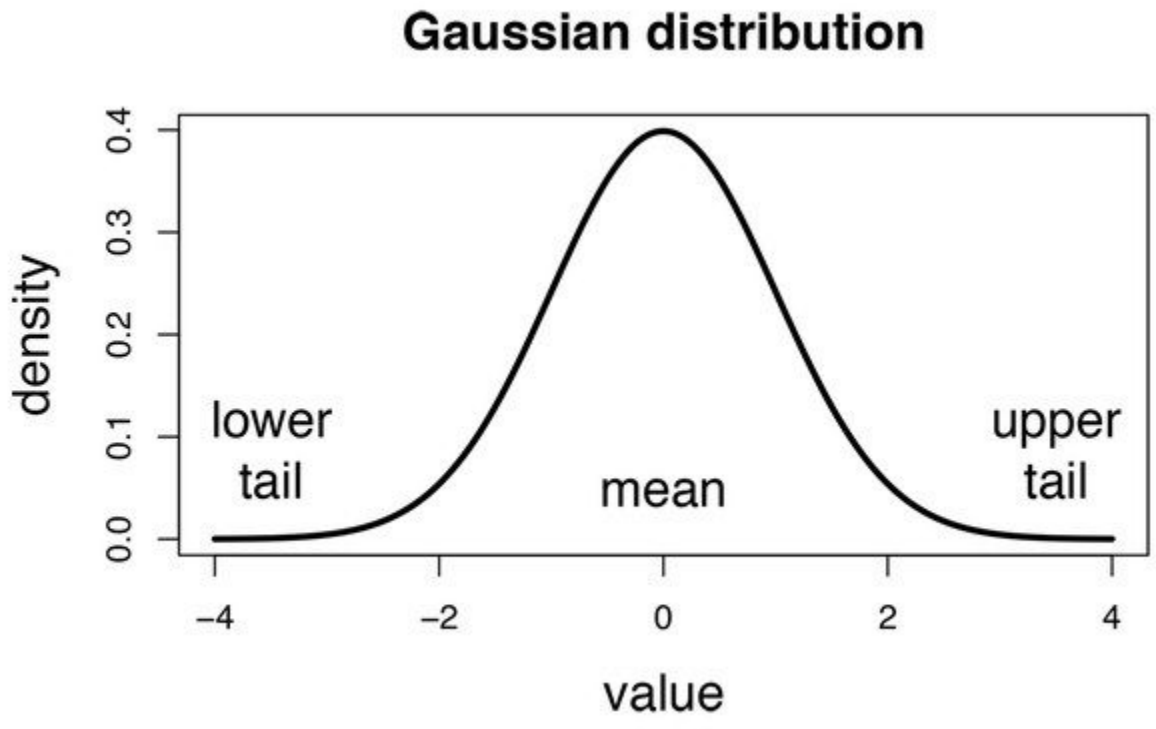
2. Bernoulli Naïve Bayes Classifier:

In the multivariate Bernoulli event model, features are independent booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks, where binary term occurrence (i.e. a word occurs in a document or not)

features are used rather than term frequencies (i.e. frequency of a word in the document).

3. Gaussian Naïve Bayes Classifier:

In Gaussian Naïve Bayes, continuous values associated with each feature are assumed to be distributed according to a **Gaussian distribution (**Normal distribution**)**. When plotted, it gives a bell-shaped curve which is symmetric about the mean of the feature values as shown below:



The likelihood of the features is assumed to be Gaussian, hence, conditional probability is given by:

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Now, what if any feature contains numerical values instead of categories i.e. Gaussian distribution.

One option is to transform the numerical values to their categorical counterparts before creating their frequency tables. The other option, as shown above, could be using the distribution of the numerical variable to have a good guess of the frequency. For example, one common method is to assume normal or gaussian distributions for numerical variables.

The probability density function for the normal distribution is defined by two parameters (mean and standard deviation).

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$ Mean

$$\sigma = \left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2\right]^{0.5}$$ Standard deviation

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$ Normal distribution

Image credit

Consider the problem of playing golf, here the only predictor is `Humidity` and `Play Golf`? is the target. Using the above formula we can calculate posterior probability if we know the mean and standard deviation.

		Humidity	*Mean*	*StDev*
Play Golf	yes	86 96 80 65 70 80 70 90 75	79.1	10.2
	no	85 90 70 95 91	86.2	9.7

$$P(\text{humidity} = 74 \mid \text{play} = \text{yes}) = \frac{1}{\sqrt{2\pi}(10.2)} e^{-\frac{(74-79.1)^2}{2(10.2)^2}} = 0.0344$$

$$P(\text{humidity} = 74 \mid \text{play} = \text{no}) = \frac{1}{\sqrt{2\pi}(9.7)} e^{-\frac{(74-86.2)^2}{2(9.7)^2}} = 0.0187$$

Case Study: Naïve Bayes Classifier From Scratch Using Python

An existing problem for any major website today is how to handle virulent and divisive content. Quora wants to tackle this problem to keep its platform a place where users can feel safe sharing their knowledge with the world.

Quora is a platform that empowers people to learn from each other. On Quora, people can ask questions and connect with others who contribute unique insights and quality answers. A key challenge is to weed out insincere questions — those founded upon false premises, or that intend to make a statement rather than look for helpful answers. The goal is to develop a Naïve Bayes classification model that identifies and flags insincere questions.

The dataset can be downloaded from here. Once you have downloaded the train and test data, load it and check.

```
import numpy as np
import pandas as pd
import os
train = pd.read_csv('./drive/My Drive/train.csv')
print(train.head())test = pd.read_csv('./drive/My Drive/test.csv')
```

	qid	question_text	target
188974	24f478324bc328608830	Which hairstyle suits a thin and parrot nosy 1...	0
969067	bdddffacb23411200857	As a Brit that admires American conservatism, ...	1
355688	45b8639338af0d29358d	What are the best ways to use Slack in board w...	0
1121566	dbc628b2821848b7edd4	Is there a way to make the name you go by, rat...	0
819998	a0ac336114d699926985	Is eating pork harmful?	0

Training dataset

Let us see how are sincere questions look like.

```
Taking a look at Sincere Questions
784979     As an icu nurse or a cardiac care unit nurse, ...
1235061    If Beef (Cow meat) is banned in Karnataka then...
850370     Which course is suitable to become a professio...
480408     What does three parallel lines mean in proposi...
500525                       Why do I always mix up my words?
Name: question_text, dtype: object
```

Sincere questions

we see how are insincere questions look like.

```
Taking a look at Insincere Questions
544746    What happened to Kim Jong-un that he decides t...
336329    Quora has an answer for anything I can imagine...
681144    Explaining why colonization contributed to ret...
250442    How powerful is the Jewish community in media ...
852460     Why does Trump always sit like he's on a toilet?
Name: question_text, dtype: object
```

Insincere questions

Text Preprocessing

The next step is to preprocess text before splitting the dataset into a train and test set. The preprocessing steps involve: Removing Numbers,

Removing Punctuations in a string, Removing Stop Words, Stemming of Words and Lemmatization of Words.

Constructing a Naive Bayes Classifier

Combine all the preprocessing techniques and create a dictionary of words and each word's count in training data.

1. Calculate probability for each word in a text and filter the words which have a probability less than threshold probability. Words with probability less than threshold probability are irrelevant.
2. Then for each word in the dictionary, create a probability of that word being in insincere questions and its probability insincere questions. Then finding the conditional probability to use in naive Bayes classifier.
3. Prediction using conditional probabilities.