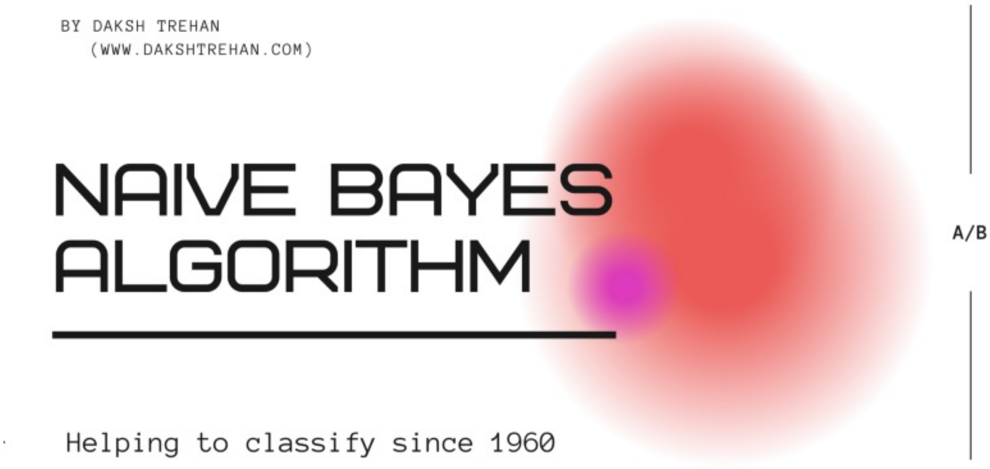
Machine Learning, Math

Naive Bayes Explained!

A quick guide to Naive Bayes, that will help you to develop a spam filtering system!



I bet most of us are familiar with the super-intelligent classification of emails as spams and non-spam. Ever wondered, what helps your email provider to classify emails so diligently into different folders and to be more precise these tasks are performed without any human intervention.

Another illustration can be regarded as; automatic classification of articles on different topics such as technology, sports, nation, and much more.

The above-mentioned techniques are use-cases of the famous Naive Bayes algorithm. The basis of this algorithm is simple: It calculates the probability of one thing based on what it knows about a related thing.

Naïve Bayes classifiers belongs to simple "probabilistic classifiers". It is *supervised learning* technique.

Naïve Bayes is comprised of two words :

Naïve: It is called Naïve because it follows conditional independence.

Bayes: It is called Bayes because it solely depends on Bayes Theorem for implementation.

Table of content :

Bayes Theorem

According to Bayes Theorem :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

“Using above formula we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the features are independent. That is presence of one particular feature does not affect the other. Hence it is called *naïve.”*

Let us take an example to understand it more clearly :

Following is the dataset determining best days to play golf.

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Our data contains 4 features wiz ***Outlook, Temperature, Humidity*** and ***Windy*** and the final output will be either ***Yes*** or ***No***.

According to the formula as discussed, the expression justifying our data would be:

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

where **y** denotes class variable, that is, if existing conditions will let us play **golf or not** and **X** denotes features that are ***Outlook, Temperature, Humidity*** and ***Windy.***

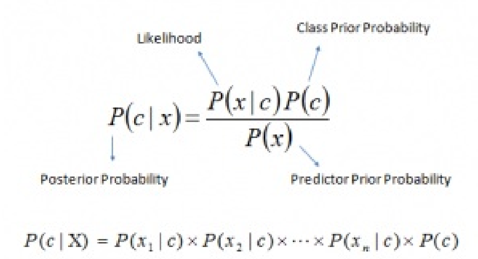
By substituting for X and expanding our expression

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

Now, you can obtain the values for each day by looking at the dataset and substitute them into the equation.

Naive Bayes Classifier

The core idea of Bayes theorem remains intact in Naive Bayes classifier.



But to develop a more efficient and robust model we need to take some assumptions that helps to reduce its complexity.

The naive Bayes algorithm does that by making an assumption of conditional independence. This helps to reduce complexity by 2n.

"The assumption of conditional independence states that, given random features X, Y and Z, we say X is conditionally independent of Y given Z, if and only if the probability distribution ruling X is independent of the value of Y given Z."

Substituting the assumptions in previous example. The features are independent of themselves, that is, if it's Rainy, it does not necessarily mean that the humidity is high. Another assumption supporting our model would be that all the features contributes equal effect on the outcome, that is, the day being humid does not have more importance in deciding to play golf or not.

> This assumption makes the Bayes algorithm, naive.

Continuing the above example :

For all features in the dataset, the Prior probability remains static. Therefore, we can use proportionality.

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

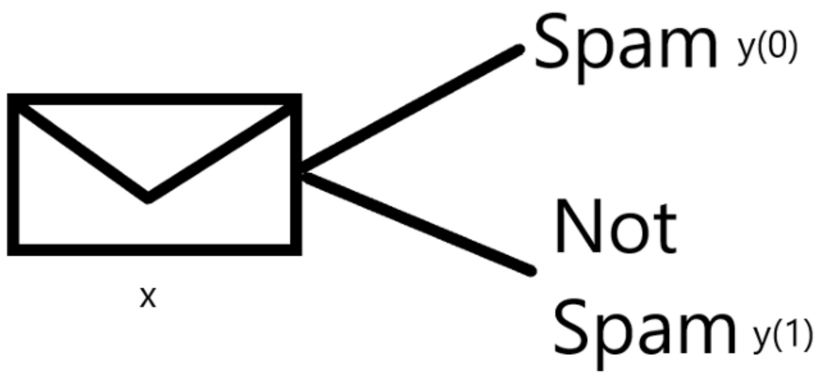
In our case, the class variable(**y**) has only binary outcomes, that is, either the day would be suitable for playing golf or it won't be.

Following the basic rule of probability, the sum of all probabilities will be one.

Suppose, P(playing golf)=0.6 then P(not playing golf) would be 0.4, therefore, to find if we can play golf on a particular day with some environmental conditions, we are going to take maximum of both values.

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

Spam detection using Naive Bayes classifier



Again, we have a class variable (y) and input features i.e. x.

Following Baye's classifier

$$P(y=1|x) = P(x|y=1)*P(y=1)/P(x)$$

$$P(y=0|x) = P(x|y=0)*P(y=0)/P(x)$$

$$P(y)=argmax(P(y|x))$$

where :-

P(y|1) : the mail is not spam ;

P(y=0) : the mail is spam ;

P(x|y=1) : We are given x and mail is not spam;

P(x|y=0) : We are given x and mail is spam.

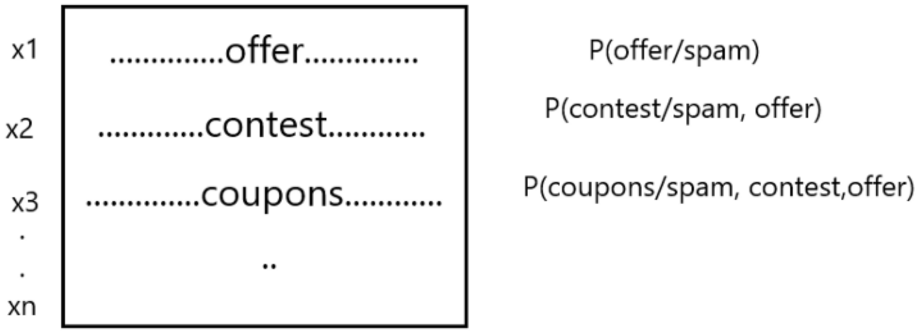
$$P(Y=1) = \text{count of all non spam mail / total mails}$$

$$P(y=0) = \text{count of all spam mail / total mails}$$

Getting probability for both cases

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

Calculating probability using bayes theorem



But according to Naive Bayes classifier, the probability of xi doesn't depends upon any feature and this is the reason it doesn't, else it will make our model really sensitive.

$$P(\text{contest}) = P(\text{contest/spam})$$
$$P(\text{coupons}) = P(\text{coupons/spam})$$

After applying conditional independence

But, what if given feature is not present in testing data ?

We all know, the idea of mass sending mass email is to promote their brands or to fascinate customers for their lucrative and jaw dropping offers.

But thanks to Naive Bayes for saving us from unwanted emails but sometimes these spammers use unique keywords, so that they can skip going to spam folder.

For illustration, take the above example our dataset contains words : offer, contest and coupons.

Now anything else coming as feature will yield zero probability, because anything else don't exist in our dataset and its count will be **0**.

But this is wrong, how can we have **0** probability; that is the loop hole to this algorithm.

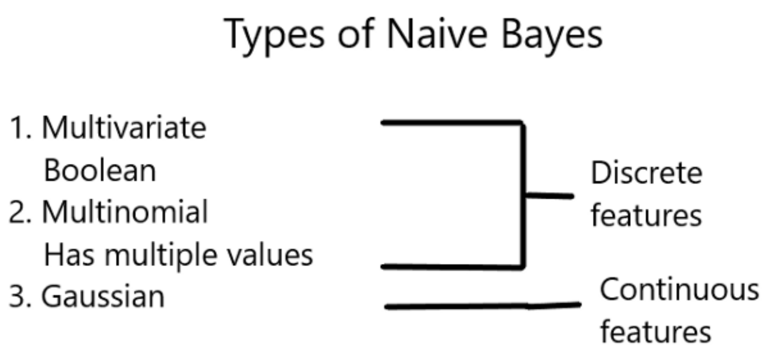
But to avoid this drawback, we use a technique known as ***Laplace smoothing.***

$$P(x|y=c) = [count(x,y=c)+1] / [count(w,y=c)+1]$$

where w= vocabulary/dataset

c=[0,1]

Types of Naive Bayes Classifier



Multinomial Naive Bayes:

This is mostly used variation for Naive Bayes and is used to classify documents/articles in different realms.

The features used by the classifier are the frequency of the words present in the document.

It contains discrete features but as output can have multiple different values.

Frequency of features are used here.

frequency = tf(t,d)
#number of times 't' appeared in document 'd'

normalized term frequency = tf(t,d)/nd
#nd = number of documents

P(x|y=c) = tf(x,d)/nd

Multivariate Bernoulli Naive Bayes:

This variations don't use frequency.

It again contains discrete features but the parameters that we use to predict the class variable take up only values yes or no.

D1 = ["Like I like to swim"]
D2 = ["I love coding"

vocab = [I, like, coding, swim, to]
x0, x1, x2, x3, x4

Now according to vocab, we will build feature vector both both D1 & D2.

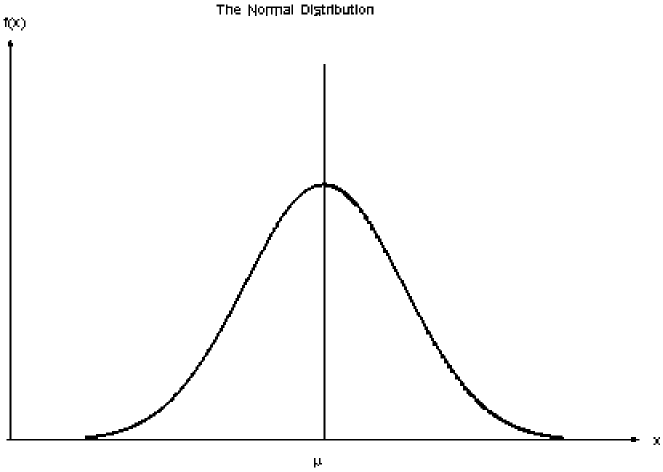
D1 = [1 1 0 1 1]
D2 = [1 1 1 0 0]

P(x|y=c) = conditional probability of generating sentence in class c.

Gaussian Naive Bayes:

The features here are discrete and not continuous, we assume that these values are sampled from a gaussian distribution.

It depends on the mean.



The formula for Gaussian Naive Bayes is :

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

Code for Naive Bayes using Sci-kit Learn

Multinomial Naive Bayes

```
from sklearn.naive_bayes import MultinomialNB
mnb=MultinomialNB()
#x_vec is feature vector; y is class variable
#xt_vec is testing data
mnb.fit(x_vec,y)                        #Training
mnb.predict(xt_vec)                     #Prediction
mnb.predict_proba(xt_vec)               #Getting prior probability
```

Multinomial Naive Bayes hosted with ❤ by **GitHub** **view raw**

Multivariate Bernoulli Naive Bayes

```
from sklearn.naive_bayes import BernoulliNB
bnb=BernoulliNB()
#x_vec is feature vector; y is class variable
#xt_vec is testing data
bnb.fit(x_vec,y)                        #Training
bnb.predict(xt_vec)                     #Prediction
bnb.predict_proba(xt_vec)               #Getting prior probability
```

Multivariate Bernoulli Naive Bayes hosted with ❤ by **GitHub** **view raw**

To create a movie review sentiment analysis using Naive Bayes follow :

dakshtrehan/Movie-Review-Classifier
*You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or...*github.com

---

Pros and Cons of Naive Bayes

Pros:

- It is easy and fast.
- Naive Bayes classifier performs better compare to other models.
- It perform well in case of categorical input.

Cons:

- The probability outputs from predict_proba are not always accurate.
- Assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Conclusion

Hopefully, this article have helped you to understand the everything about Naive Bayes and its use cases.

As always, thank you so much for reading, and please share this article if you found it useful! :)

Feel free to connect:

> *Join me at www.dakshtrehan.com*
>
> *LinkedIN ~ https://www.linkedin.com/in/dakshtrehan/*
>
> *Instagram ~ https://www.instagram.com/_daksh_trehan_/*
>
> *Github ~ https://github.com/dakshtrehan*

Check my other articles:-

The inescapable AI algorithm: TikTok
*Describing a progressive recommendation system used by TikTok to keep its users hooked!*towardsdatascience.com
Why are YOU responsible for George Floyd's murder & Delhi Communal Riots!!
*A ML enthusiast's approach to change the world.*medium.com
Detecting COVID-19 using Deep Learning
*A practical approach to help medical practitioners helping us in the battle against COVID-19*towardsdatascience.com
Activation Functions Explained
*Step, Sigmoid, Hyperbolic Tangent, Softmax, ReLU, Leaky ReLU Explained*medium.com
Parameters Optimization Explained
*A brief yet descriptive guide to Gradient Descent, ADAM, ADAGRAD, RMSProp*towardsdatascience.com
Gradient Descent Explained
*A comprehensive guide to Gradient Descent*towardsdatascience.com
Logistic Regression Explained
*Explaining Logistic Regression as easy as it could be.*towardsdatascience.com
Linear Regression Explained
*Explaining Linear Regression as easy as it could be.*medium.com
Determining perfect fit for your ML model.
*Teaching Overfitting vs Underfitting vs Perfect fit in easiest way.*medium.com
Relating Machine Learning Techniques to Real-Life.
*Explaining types of ML model as easy as it could be.*levelup.gitconnected.com
Serving Data Science to a Rookie
*So, last week my team head asked me to interview some of the possible interns for the team, for the role of data…*medium.com

By Daksh Trehan on June 25, 2020.

[Canonical link]

Exported from [Medium] on July 15, 2020.