Naive Bayes Classifier Explained: Applications and Practice Problems of Naive Bayes Classifier

Sunil Ray — Published On September 11, 2017 and Last Modified On May 11th, 2023
Beginner Data Science Machine Learning Maths Probability Python R Technique

Introduction

Let's start with a practical example of using the Naive Bayes Algorithm.

Assume this is a situation you've got into in your data science project:

You are working on a classification problem and have generated your set of hypotheses, created features, and discussed the importance of variables. Within an hour, stakeholders want to see the first cut of the model.

What will you do? You have hundreds of thousands of data points and several variables in your training data set. In such a situation, if I were in your place, I would have used '**Naive Bayes**,' which can be extremely fast relative to other classification algorithms. It works on Bayes' theorem of probability to predict the class of unknown data sets.

In this article, I'll explain the basics of Naive Bayes (NB) in machine learning (ML), so that the next time you come across large data sets, you can bring this classification algorithm in ML (which is a part of AI) to action. In addition, if you are a newbie in Python or R, you should not be overwhelmed by the presence of available codes in this article.

Learning Objectives

- Understand the definition and working of the Naive Bayes algorithm.
- Get to know the various applications, pros, and cons of the classifier.
- Learn how to implement the NB Classifier or bayesian classification in R and Python with a sample project.

If you prefer to learn the Naive Bayes' theorem from the basics concepts to the implementation in a structured manner, you can enroll in this free course:

- Naive Bayes Course from Scratch

Are you a beginner in Machine Learning? Do you want to master the machine learning algorithms like Naive Bayes? Here is a comprehensive course covering the machine learning and deep learning algorithms in detail –

- Certified AI & ML Blackbelt+ Program

Table of contents

Naive Bayes Classifier Explained: Applications and Practice Problems of Naive Bayes Classifier

What Is the Naive Bayes Algorithm?

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.

The Naïve Bayes classifier is a popular supervised machine learning algorithm used for classification tasks such as text classification. It belongs to the family of generative learning algorithms, which means that it models the distribution of inputs for a given class or category. This approach is based on the assumption that the features of the input data are conditionally independent given the class, allowing the algorithm to make predictions quickly and accurately.

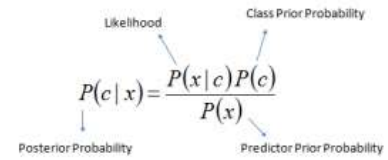
In statistics, naive Bayes classifiers are considered as simple probabilistic classifiers that apply Bayes' theorem. This theorem is based on the probability of a hypothesis, given the data and some prior knowledge. The naive Bayes classifier assumes that all features in the input data are independent of each other, which is often not true in real-world scenarios. However, despite this simplifying assumption, the naive Bayes classifier is widely used because of its efficiency and good performance in many real-world applications.

Moreover, it is worth noting that naive Bayes classifiers are among the simplest Bayesian network models, yet they can achieve high accuracy levels when coupled with kernel density estimation. This technique involves using a kernel function to estimate the probability density function of the input data, allowing the classifier to improve its performance in complex scenarios where the data distribution is not well-defined. As a result, the naive Bayes classifier is a powerful tool in machine learning, particularly in text classification, spam filtering, and sentiment analysis, among others.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

An NB model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of computing posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Above,

- $P(c|x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).

Naive Bayes Classifier Explained: Applications and Practice Problems of Naive Bayes Classifier

Sample Project to Apply Naive Bayes

Problem Statement

HR analytics is revolutionizing the way human resources departments operate, leading to higher efficiency and better results overall. Human resources have been using analytics for years.

However, the collection, processing, and analysis of data have been largely manual, and given the nature of human resources dynamics and HR KPIs, the approach has been constraining HR. Therefore, it is surprising that HR departments woke up to the utility of machine learning so late in the game. Here is an opportunity to try predictive analytics in identifying the employees most likely to get promoted.

Practice Now

How Do Naive Bayes Algorithms Work?

Time needed: 1 minute.

Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

1. **Convert the data set into a frequency table**

   In this first step data set is converted into a frequency table

2. **Create Likelihood table by finding the probabilities**

   Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

3. **Use Naive Bayesian equation to calculate the posterior probability**

   Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of the prediction.

Problem: Players will play if the weather is sunny. Is this statement correct?

We can solve it using the above-discussed method of posterior probability.

P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)

Here P(Sunny | Yes) * P(Yes) is in the numerator, and P (Sunny) is in the denominator.

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability

What Are the Pros and Cons of Naive Bayes?

Pros:

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, the classifier performs better compared to other machine learning models like logistic regression or decision tree, and requires less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as "Zero Frequency". To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- On the other side, Naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.
- Another limitation of this algorithm is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Applications of Naive Bayes Algorithms

- **Real-time Prediction:** Naive Bayesian classifier is an eager learning classifier and it is super fast. Thus, it could be used for making predictions in real time.
- **Multi-class Prediction:** This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayesian classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- **Recommendation System:** Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.

How to Build a Basic Model Using Naive Bayes in Python & R?

Again, scikit learn (python library) will help here to build a Naive Bayes model in Python. There are five types of NB models under the scikit-learn library:

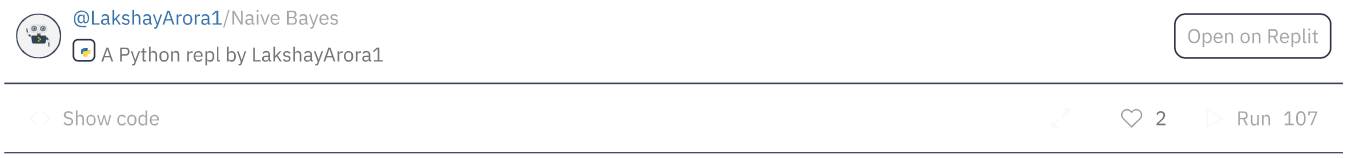
- **Gaussian Naive Bayes:** gaussiannb is used in classification tasks and it assumes that feature values follow a gaussian distribution.
- **Multinomial Naive Bayes:** It is used for discrete counts. For example, let's say, we have a text classification problem. Here we can consider Bernoulli trials which is one step further and instead of "word occurring in the document", we have "count how often word occurs in the document", you can think of it as "number of times outcome number x_i is observed over the n trials".

the model weights. So, this is suitable for imbalanced data sets and often outperforms the MNB on text classification tasks.

- **Categorical Naive Bayes:** Categorical Naive Bayes is useful if the features are categorically distributed. We have to encode the categorical variable in the numeric format using the ordinal encoder for using this algorithm.

Python Code:

Try out the below code in the coding window and check your results on the fly!



R Code:

```
require(e1071) #Holds the Naive Bayes Classifier
Train <- read.csv(file.choose())
Test <- read.csv(file.choose())

#Make sure the target variable is of a two-class classification problem only

levels(Train$Item_Fat_Content)

model <- naiveBayes(Item_Fat_Content~., data = Train)
class(model)
pred <- predict(model,Test)
table(pred)
```

Above, we looked at the basic NB Model. You can improve the power of this basic model by tuning parameters and handling assumptions intelligently. Let's look at the methods to improve the performance of this model. I recommend you go through this document for more details on Text classification using Naive Bayes.

Tips to Improve the Power of the NB Model

Here are some tips for improving power of Naive Bayes Model:

- If continuous features do not have normal distribution, we should use transformation or different methods to convert it in normal distribution.

Naive Bayes Classifier Explained: Applications and Practice Problems of Naive Bayes Classifier

- Naive Bayes classifiers has limited options for parameter tuning like alpha=1 for smoothing, fit_prior=[True|False] to learn class prior probabilities or not and some other options (look at detail here). I would recommend to focus on your pre-processing of data and the feature selection.
- You might think to apply some *classifier combination technique like* ensembling, bagging and boosting but these methods would not help. Actually, "ensembling, boosting, bagging" won't help since their purpose is to reduce variance. Naive Bayes has no variance to minimize.

Conclusion

In this article, we looked at one of the supervised machine learning algorithms, "Naive Bayes" mainly used for classification. Congrats, if you've thoroughly & understood this article, you've already taken your first step toward mastering this algorithm. From here, all you need is practice.

Further, I would suggest you focus more on data pre-processing and feature selection before applying the algorithm. In a future post, I will discuss about text and document classification using naive bayes in more detail.

Did you find this article helpful? Please share your opinions/thoughts in the comments section below.

Key Takeaways

- The Naive Bayes algorithm is one of the most popular and simple machine learning classification algorithms.
- It is based on the Bayes' Theorem for calculating probabilities and conditional probabilities.
- You can use it for real-time and multi-class predictions, text classifications, spam filtering, sentiment analysis, and a lot more.

Frequently Asked Questions

Q1. What is naive in Naive Bayes classifier?

A. Naive Bayes classifier assumes features are independent of each other. Since that is rarely possible in real-life data, the classifier is called naive.

Q2. What are the 3 different Naive Bayes classifiers?

A. Out of the 5 different Naive Bayes classifiers under sklearn.naive_bayes, the 3 most widely used ones are Gaussian, Multinomial, and Bernoulli.

Q3. What is the difference between Naive Bayes and Maximum Entropy?

A. In Naive Bayes classifier each feature is used independently, while in Maximum Entropy classifier, the model uses search-based optimization to find weights for features. Reference:http://sentiment.christopherpotts.net/classifiers.html

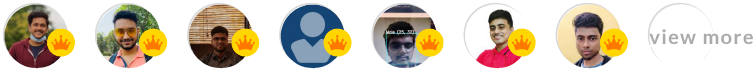
You can use the following free resource to learn more

- Machine Learning Certification Course for Beginners

bayes theorem classification conditional probability data science live coding machine learning Naive Bayes naive bayes classifier naive bayes in R probability python

Naive Bayes Classifier Explained: Applications and Practice Problems of Naive Bayes Classifier

Our Top Authors



Download

Analytics Vidhya App for the Latest blog/Article

Previous Post

4 Essential Tools any Data Scientist can use to improve their productivity

Next Post

Python vs. R vs. SAS – which tool should I learn for Data Science?

44 thoughts on "Naive Bayes Classifier Explained: Applications and Practice Problems of Naive Bayes Classifier"

Arun CR says:
September 14, 2015 at 5:36 am

Hi Sunil , From the weather and play table which is table [1] we know that frequency of sunny is 5 and play when sunny is 3 no play when suny is 2 so probability(play/sunny) is 3/5 = 0.6 Why do we need conditional probabilty to solve this? Is there problems that can be solved only using conditional probability. can you suggest such examples. Thanks, Arun

Reply

Arun CR says:
September 14, 2015 at 5:42 am

Great article and provides nice information.

Reply

Nishi Singh says:
December 12, 2015 at 10:35 am

Amazing content and useful information

Reply

matt says:
January 24, 2016 at 4:42 am

Arun, An example of a problem that requires the use of conditional probability is the Monty Hall problem. https://en.wikipedia.org/wiki/Monty_Hall_problem Conditional probability is used to solve this particular problem because the solution depends on Bayes' Theorem. Which was described earlier in the article.

Reply

Leena says:

March 07, 2016 at 2:44 pm

very useful article.

Reply

RAJKUMAR says:
March 16, 2016 at 1:30 pm

Very nice....but...if u dont mind...can you please give me that code in JAVA ...

Reply

Erdem Karakoylu says:
March 17, 2016 at 2:26 pm

It's a trivial example for illustration. The "Likelihood table" (a confusing misnomer, I think) is in fact a probability table that has the JOINT weather and play outcome probabilities in the center, and the MARGINAL probabilities of one variable (from integrating out the other variable from the joint) on the side and bottom. Say, weather type = w and play outcome = p. P(w,p) is the joint probabilities and P(p) and P(w) are the marginals. Bayes rule described above by Sunil stems from: P(w,p) = P(w|p) * P(p) = P(p|w) * P(w). From the center cells we have P(w,p) and from the side/bottom we get P(p) and P(w). Depending on what you need to calculate, it follows that: (1): P(w|p) = P(w,p) / P(p) and (2:)P(p|w) = P(w,p) / P(w), which is what you did with P(sunny,yes) = 3/14 and P(w) = 5/14, yielding (3/14) (14/5), with the 14's cancelling out. The main Bayes take away is that often, one of the two quantities above, P(w|p) or P(p|w) is much harder to get at than the other. So if you're a practitioner you'll come to see this as one of two mathematical miracles regarding this topic, the other being the applicability of Markov Chain Monte Carlo in circumventing some nasty integrals Bayes might throw at you. But I digress. Best, Erdem.

Reply

---

jitesh says:
April 05, 2016 at 3:20 am

is it possible to classify new tuple in orange data mining tool??

Reply

SPGupta says:
April 11, 2016 at 4:43 pm

good.

Reply

---

devenir riche au Maroc says:
April 14, 2016 at 5:13 pm

I am really impressed together with your writing skills as wwell as with the format to your weblog. Is that this a paid theme or did you customiz it yourself? Anyway stay up the excellent quality writing, it's uncommon tto see a great weblog like this one these days..

Reply

รับซื้อ Patek Philippe says:
April 22, 2016 at 3:52 am

You should be a part of a contest for one of the most useful websites online. I will highly recommend this blog!

Reply

TingTing says:
April 22, 2016 at 3:18 pm

Thanks for tips to improve the performance of models, that 's really precious experience.

Reply

Leanna Partridge says:
May 11, 2016 at 8:18 am

Nice piece - Just to add my thoughts , people require a CA OCF-1 , We used a sample document here http://goo.gl/ibPgs2

Reply

transformation or different methods to convert it in normal distribution.' Can you provide an example or a link to the techniques? Thank you, MB

Reply

bd tv says:
June 06, 2016 at 2:45 am

Can I simply just say what a comfort too find someone who actualy knows what they're talking about over the internet. You certainly know how to bring an issue to light and make it important. A lot more people ought to read this and understand this side of your story. I can't believe you aren't more popular because you certainly have the gift.

Reply

nir says:
July 13, 2016 at 6:57 pm

Great article! Thanks. Are there any similar articles for other classification algorithms specially target towards textual features and mix of textual/numeric features?

Reply

John Siano says:
July 21, 2016 at 3:27 am

I had the same question. Have you found an answer? Thanks!

Reply

Nick says:
August 19, 2016 at 6:29 am

great article with basic clarity.....nice one

Reply

Catherine says:
August 29, 2016 at 5:18 pm

This article is extremely clear and well laid-out. Thank you!

Reply

pangavhane nitin says:
August 31, 2016 at 8:41 am

ty

Reply

alfiya says:
August 31, 2016 at 9:26 am

Explanation given in simple word. Well explained! Loved this article.

Reply

Chris Rucker says:
September 01, 2016 at 7:02 pm

The 'y' should be capitalized in your code - great article though.

Reply

Akash Swamy says:
September 06, 2016 at 7:24 am

This is the best explanation of NB so far simple and short :)

Reply

so : model.fit(x, Y) Thanks!

Reply

Adnan says:
September 20, 2016 at 5:12 am

Is this dataset related to weather? I am confused as a newbie. Can you please guide?

Reply

Ismael Ezequiel says:
October 15, 2016 at 12:12 am

import pandas as pd person = pd.read_csv('example.csv') mask = np.random.rand(len(sales)) < 0.8 train = sales[mask] test = sales[~mask]

Reply

bh says:
October 19, 2016 at 5:40 pm

best artical that help me to understand this concept

Reply

Richard says:
October 25, 2016 at 7:52 pm

Am new to machine learning and this article was handy to me in understanding naive bayes especially the data on weather and play in the table. Thanks for sharing keep up

Reply

Lisa says:
November 10, 2016 at 8:29 am

Thanks to you I can totally understand NB classifier.

Reply

T B says:
July 03, 2017 at 12:17 am

Really nice article, very use-full for concept building.

Reply

AKshay says:
July 04, 2017 at 8:40 pm

I didn't understand the 3rd step. Highest probability out of which probability values? >> Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability. Higher than what?

Reply

DN says:
July 30, 2017 at 8:29 am

Concept explained well... nice Article

Reply

Rajeshwari says:
August 31, 2017 at 2:36 pm

thanks nice artical that help me to understand this concept

Reply

amit Kumar yadav says:
September 18, 2017 at 12:49 am

Good article and I am waiting for text and documents classification using naive base algorithm.

Reply

Aishwarya says:
November 26, 2017 at 10:12 pm

Good start point for beginners

Reply

Abdul Samad says:
April 12, 2018 at 10:19 am

Weldone sanil I have a question regarding naive bayes,currently i am working on a project that is detect depression through naive bayes algorithm so plz suggest few links regarding my projects.i shall be gratefull to you. Thanku so much

Reply

Aishwarya Singh says:
April 16, 2018 at 4:37 pm

Hi Abdul, Refer this link.

Reply

Tongesai Maune says:
May 04, 2018 at 7:21 pm

I am not understanding the x and the y variables. Can someone help me

Reply

Aishwarya Singh says:
May 06, 2018 at 1:24 pm

Hi Tongesai, X here represents the dependent variable and y is used for target variable (which is to be predicted).

Reply

Artificial Neural Networks with Edge-Based Architecture – The Informaticists says:
August 25, 2020 at 1:24 pm

[...] [4] Ray, Sunil. (2020, April 01). Learn Naive Bayes Algorithm: Naive Bayes Classifier Examples. Retrieved August 05, 2020, from https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ [...]

Reply

Jayer says:
January 23, 2023 at 8:43 pm

Hello, how do i check which variables are significant using the naive Bayes algorithm in R

Reply

Sisay says:
January 26, 2023 at 7:35 pm

It is nice, please make it open the code

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Naive Bayes Classifier Explained: Applications and Practice Problems of Naive Bayes Classifier

Notify me of follow-up comments by email.

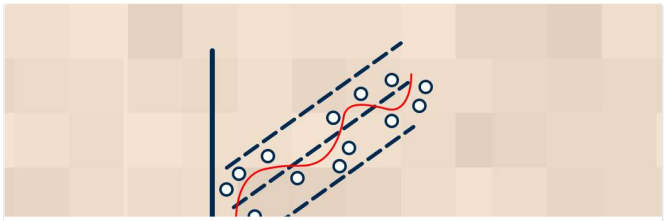
Notify me of new posts by email.

Submit

Top Resources

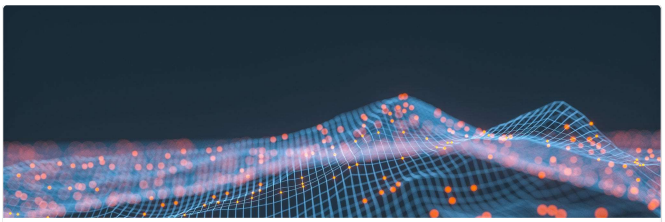
Understand Random Forest Algorithms With Examples (Updated 2023)

Sruthi E R - JUN 17, 2021



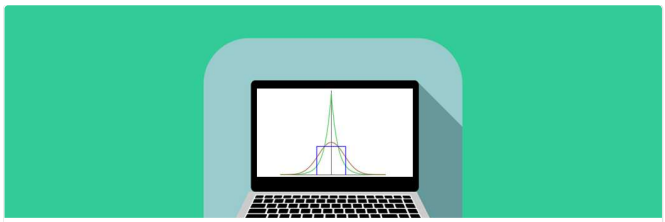
Everything you need to Know about Linear Regression!

KAVITA MALI - OCT 04, 2021



Decision Tree Algorithm – A Complete Guide

Anshul Saini - AUG 29, 2021



Skewness and Kurtosis: Quick Guide (Updated 2023)

Suvarna Gawali - MAY 02, 2021