6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R

Home

Sunil Ray — September 11, 2017

Beginner Data Science Machine Learning Maths Probability Python R Technique

Note: This article was originally published on Sep 13th, 2015 and updated on Sept 11th, 2017

Overview

- Understand one of the most popular and simple machine learning classification algorithms, the Naive Bayes algorithm
- It is based on the Bayes Theorem for calculating probabilities and conditional probabilities
- Learn how to implement the Naive Bayes Classifier in R and Python

Introduction

Here's a situation you've got into in your data science project:

You are working on a classification problem and have generated your set of hypothesis, 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 quite a few 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 this algorithm, so that next time when you come across large data sets, you can bring this algorithm 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.

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

- Naive Bayes 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

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

Table of Contents

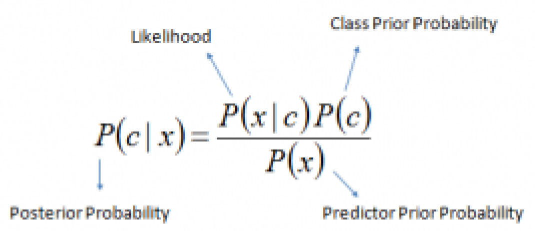
What is Naive Bayes algorithm?

It is a classification technique based on Bayes' Theorem with an assumption of independence 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.

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'.

Naive Bayes 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 calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



$$P(c|x)=\frac{P(x|c)P(c)}{P(x)}$$

$$P(c|\mathbf{X}) = P(x_1|c)\times P(x_2|c)\times\cdots\times P(x_n|c)\times P(c)$$

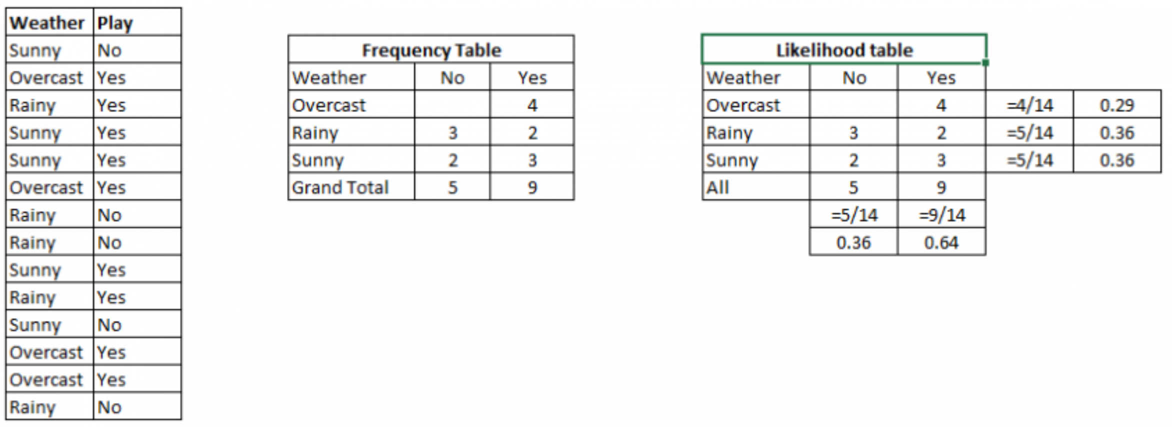
Above,

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

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.

Step 1: Convert the data set into a frequency table

Step 2: 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		

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

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

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

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

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.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

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, a Naive Bayes classifier performs better compare to other models like logistic regression and you need 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 Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

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 Bayes 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

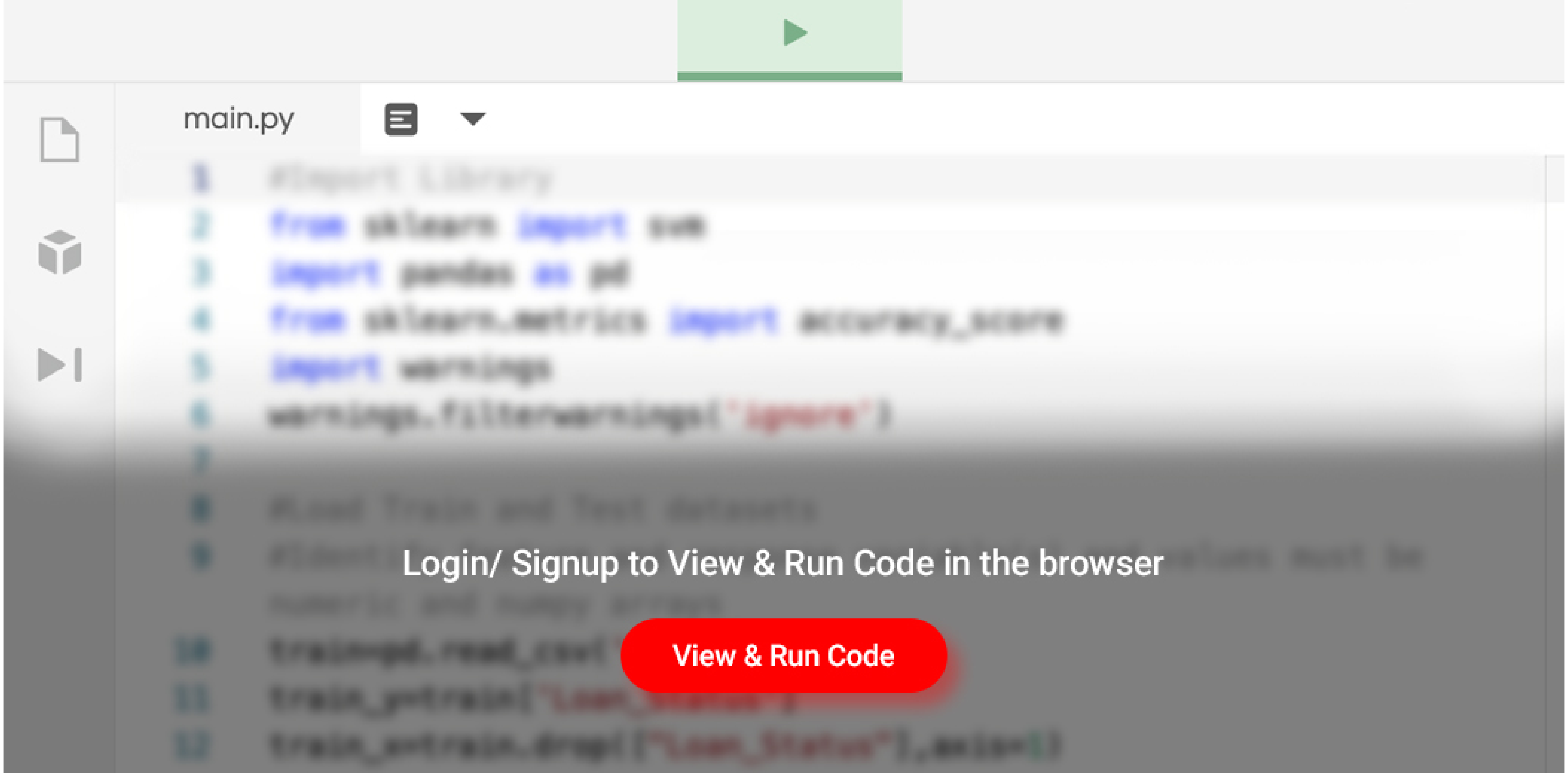
How to build a basic model using Naive Bayes in Python and R?

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

- **Gaussian:** It is used in classification and it assumes that features follow a normal distribution.
- **Multinomial:** 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".
- **Bernoulli:** The binomial model is useful if your feature vectors are binary (i.e. zeros and ones). One application would be text classification with 'bag of words' model where the 1s & 0s are "word occurs in the document" and "word does not occur in the document" respectively.

Python Code:

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



```
#Import Library
from sklearn import svm
import pandas as pd
from sklearn.metrics import accuracy_score
import warnings
warnings.filterwarnings('ignore')

#Load Train and Test datasets
#Ident[illegible] values must be numeric and numpy arrays
train=pd.read_csv([illegible]
train_y=train['Loan_Status']
train_x=train.drop(["Loan_Status"],axis=1)
```

R Code:

```
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 Naive Bayes model, you can improve the power of this basic model by tuning parameters and handle assumption intelligently. Let's look at the methods to improve the performance of Naive Bayes Model. I'd recommend you to go through this document for more details on Text classification using Naive Bayes.

Tips to improve the power of Naive Bayes 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.
- If test data set has zero frequency issue, apply smoothing techniques "Laplace Correction" to predict the class of test data set.
- Remove correlated features, as the highly correlated features are voted twice in the model and it can lead to over inflating importance.
- 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.

End Notes

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

Further, I would suggest you to focus more on data pre-processing and feature selection prior to applying Naive Bayes algorithm.0 In 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.

You can use the following free resource to learn- Naive Bayes-

- Machine Learning Certification Course for Beginners

Learn, engage, compete, and get hired!

Performing Sentiment Analysis With Naive Bayes Classifier!
Naive Bayes Algorithm: A Complete guide for Data Science Enthusiasts
Introduction To Naive Bayes Algorithm

6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R

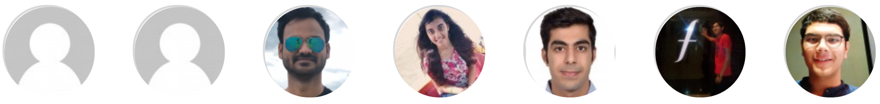
bayes theorem classification conditional probability data science live coding machine learning Naive Bayes naive bayes classifier naive bayes in R probability python

About the Author

Sunil Ray

I am a Business Analytics and Intelligence professional with deep experience in the Indian Insurance industry. I have worked for various multi-national Insurance companies in last 7 years.

Our Top Authors



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?

42 thoughts on "6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R"

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

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:

February 08, 2016 at 5:59 am

I'm new to machine learning and Python.Could you please help to read data from CSV and to separate the same data set to training and test data

Reply

Sushma honnidige 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

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

Miguel Batista says:
May 11, 2016 at 9:54 am

Hi, I have a question regarding this statement: 'If continuous features do not have normal distribution, we should use 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 aЬout over the internet. You certainly know how to bring an issue to light and make it impοrtant. 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

John says:

September 09, 2016 at 5:20 pm

Great article! Really enjoyed it. Just wanted to point out a small error in the Python code. Should be a capital "Y" in the predict like 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 20, 2017 at 10:50 am

Superb information in just one blog.Enjoyed the simplicity.Thanks for the effort.

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

Leave a Reply

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

Comment

Name*

Email*

Website

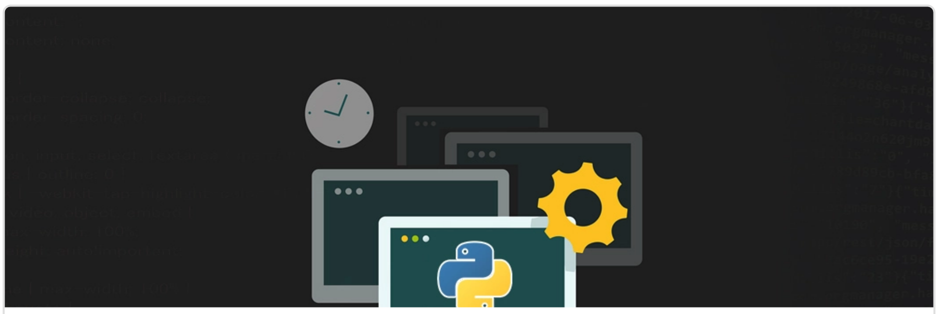
□ Notify me of follow-up comments by email.

□ Notify me of new posts by email.

Top Resources

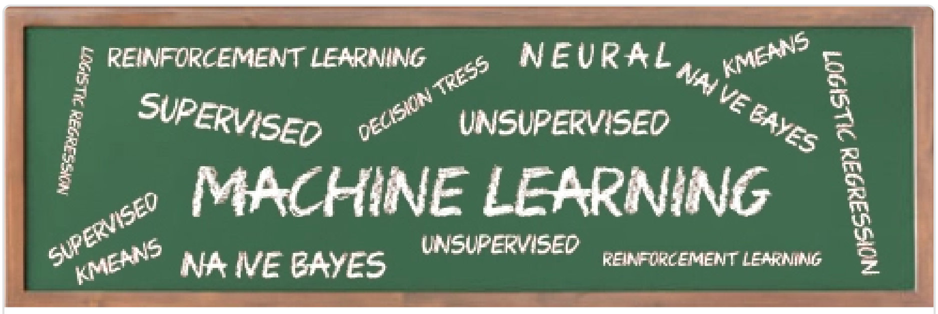
A Comprehensive Guide to PySpark RDD Operations

Rahul Shah - OCT 09, 2021



Python Tutorial: Working with CSV file for Data Science

Harika Bonthu - AUG 21, 2021



Commonly used Machine Learning Algorithms (with Python and R Codes)

Sunil Ray - SEP 09, 2017

3 Interesting Python Projects With Code for Beginners!

Gaurav Sharma - JUL 18, 2021

Download App

Analytics Vidhya

About Us

Our Team

Careers

Contact us

Companies

Post Jobs

Trainings

Hiring Hackathons

Advertising

Data Scientists

Blog

Hackathon

Discussions

Apply Jobs

Visit us

Privacy Policy Terms of Use Refund Policy