# **Naive Bayes**

Naive Bayes is a simple probabilistic classifier based on Bayes theorem. It assumes that the features of a given data point are independent of each other which is often not the case in reality,

This theorem allows us to “invert” conditional probabilities as a reminder, conditional probabilities represent the probability of an event given some other event has accrued which is represented with the following formula.

Naïve Bayes classifier work differently in that they operate under of key assumptions, earning? it the title of “naive”. It assumes that the prediction in the Naïve Bayes model are conditionally independent, or unrelated to any of the other feature in the model. It also assumes that all other features in the model. It is also assumed that all features contribute equally to the outcome. While these assumptions are often violated in the real-world scenarios, it simplifies a classification problem by making it more computationally tractable.

Similar to Bayes’ theorem, it’ll use conditional and prior probabilities to calculate the posterior probabilities using the following formula.

Posterior probability = (conditional probability) \* (prior probability) / evidence (“stabilizer”)

P(h|D) = P(D|h) P(h) / P(D)

Where

* P(h): the probability of hypothesis h being true (regardless of the data), This is known as the prior probability of h.
* P(D): the probability of the data (regardless of the hypothesis). This is known as posterior probability.
* P(h|D): the probability of hypothesis h given the data D. This is known as posterior probability
* P(D|h): the probability of data d given that the hypothesis h was true. This is known as posterior probability.

The reliationship given class variable y and dependent feature vector x1 throuth xn

P(y | x1, . . . ., xn ) = P(y) P(x1, . . .,xn | y) / P (x1, . . . , xn)

Types of the Naïve Bayes Model

## **Gaussian Naïve Bayes**

It is straightforward algorithm used when the attributes are continuous. The attributes present in the data should follow the rule of Gaussian distribution. It remarkably quickens the search, and under lenient conditions, the error will be tree times greater then

## **Optimal Naïve Bayes**

Multinomial Naïve Bayes is used on documentation classification issues. The features needed for this type are frequency of the words converted from the documents. The distribution is parametrized by vectors y = (y, . . . ,yn) for each class y , where n is the number of features ( in the text classification , the size of the vocabulary) and yi is the probability of feature I appearing in a sample belonging to class y.

The parameters y is estimated by a smoothed version of maximum likelihood relative frequency counting.

Where is the number of times feature it appears in a sample of class y in the training set T and is the total count of all features for class y

## **Bernoulli Naïve Bayes**

Bernoulli Naïve Bayes is an algorithm that is useful for data has binary or Boolean attributes. The attributes will have a value yes or no, useful or not granted or rejected.

The decision rule for Bernoulli naïve Bayes is based on

P(xi | y) = P ( xi = 1|y) xi + ( 1 – P(xi = 1| y ))(1- xi)

Optimal Naïve Bayes select the class that has the greatest posterior probability of happenings. As per the name, it is optimal but it will go through all the possibilities, which is very slow and time – consuming.

**Naïve Bayes classifier calculates the probability of an event in the following steps.**

1. Calculate the prior probability for given class labels.
2. Find Likely hood probability with each attribute for each class
3. Put these values in Bayes Formula and calculate posterior probability.
4. See which class has a higher probability, given the input belongs to the higher probability class

## **Examples**

The naïve Bayes algorithm is used for various real-world problems like.

* Text classification the Naïve Bayes Algorithm is used as a probabilistic learning technique for text classification. It is one of the best-known algorithm used for document classification of one or money classes.

## **Pseudocode**

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score

# Load the Iris dataset

iris = load\_iris()

X = iris.data

y = iris.target

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Initialize the Gaussian Naive Bayes classifier

nb\_classifier = GaussianNB()

# Train the classifier

nb\_classifier.fit(X\_train, y\_train)

# Make predictions on the testing set

y\_pred = nb\_classifier.predict(X\_test)

# Calculate the accuracy of the classifier

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)