International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

# Naive Bayes: A Text Classifier based on Machine Learning

**Ranjeet Kaur**

*Lecturer, Department of BCA, MIMIT, Malout, Punjab, India*

ABSTRACT

*It is assessed that about 80 percent of all data is unstructured, with one of the most commonly known kinds of unstructured information being content. Dissecting, interpreting, organizing, and sorting out content data is challenging and tedious in view of the untidy concept of content, so most companies struggle to eliminate an opportunity from it.This is the arrangement of role material with AI steps in. Organizations can structure business data, such as email, authoritative records, web pages, visit discussions, and online life messages, in a fast and realistic way by using content classifiers.This helps organizations to spare time when updating information on content, helping to illuminate business decisions, and robotizing business types.Most AI techniques have produced outperforming results in common language training.*

Keywords: Data Mining Tools; Navie Bayes; NLP; Classification Methods

## 1. Introduction

Text classification is the way toward appointing labels or classes to content as indicated by its substance. It's one of the principal assignments in Natural Language Processing (NLP) with wide applications, for example, opinion investigation, theme naming, spam recognition, and expectation identification.Data classification is now a common task applied in many application areas such as grouping similar functional genomes, text that demonstrate the same pattern, partitioning web pages showing the same structure, and so on [1].

Text Classification is a computerized procedure of grouping of content into predefined classifications. We can order Emails into spam or non-spam, news stories into various classes like Politics, Stock Market, Sports, and so on.

Unstructured information as content is all over the place: messages, talks, website pages, web-based life, bolster tickets, overview reactions, and that's only the tip of the iceberg. Text can be an amazingly rich wellspring of data, yet removing bits of knowledge from it very well may be hard and tedious because of its unstructured nature. Organizations are going to content arrangement for organizing content in a quick and cost-productive approach to upgrade dynamic and mechanize forms.

Text classification is the undertaking of appointing a lot of predefined classes to free-content.Text classifiers can be utilized to sort out, structure, and order essentially anything.As a model, investigate the accompanying content underneath:

*“The user interface is very direct and easy to use.”*

A classifier can take this text as an input, analyze its content, and then and automatically assign relevant tags, such as *UI* and *Easy To Use* that represent this text:

* *Corresponding author*
E-mail address: *er.ranjeetsandhu@gmail.com*

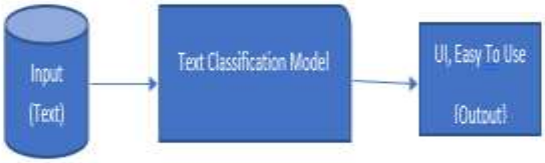


**Figure 1: A Text Classifier**

Text classification should be conceivable in two particular habits: manual and modified request. In the past, a human annotator unravels the substance of substance and orders it as requirements be. This method when in doubt can give quality results but at this point is the perfect time using and exorbitant [3]. The last applies AI, normal language getting ready, and various techniques to normally mastermind message in a snappier and all the more monetarily astute way. One reason AI is turning out to be standard is a direct result of the bunch of open source libraries accessible for designer's keen on applying it. In spite of the fact that they despite everything require AI information for building and conveying models, these libraries offer a reasonable degree of deliberation and improvement. Python, Java, and R all offer a wide determination of AI libraries that are effectively evolved and give a differing set of highlights, execution, and abilities. A few instances of text classification are:

- Understanding crowd conclusion from web- based life,
- Detection of spam and non-spam emails,
- Auto tagging of customer queries, and
- Categorization of news stories into characterized subjects.

There are many approaches to automatic text classification, which can be grouped into three different types of systems [3]:

- Rule-based systems
- Machine Learning based systems
- Hybrid systems

### 1.1 Rule-based Systems:

Rule-based methodologies order content into sorted out gatherings by utilizing a lot of high-quality semantic principles. These principles train the framework to utilize semantically applicable components of a book to recognize significant classes dependent on its substance. Each standard comprises of a precursor or design and an anticipated class [11].

### 1.2 Machine Learning Based Systems:

Rather than depending on physically made principles, content arrangement with AI figures out how to mention orders dependent on past objective facts. By utilizing pre-marked models as preparing information, an AI calculation can become familiar with the various relationship between bits of content and that a specific yield (for example labels) is normal for a specific info (for example content) [2].

### 1.3 Hybrid Systems:

Hybrids systems join a base classifier prepared with AI and a standard based framework, which is utilized to additionally improve the outcomes. These hybrid systems can be effectively calibrated by including explicit principles for those clashing labels that haven't been accurately demonstrated by the base classifier.

## 2. Machine Learning Based Text Classification Algorithms

Some of the most popular machine learning algorithms for creating text classification models include the naive bayes family of algorithms, support vector machines, and deep learning.

### 2.1 Naive Bayes:

Naive Bayes is a group of measurable calculations. we can utilize while doing content arrangement. One of the individuals from that family is Multinomial Naive Bayes (MNB). One of its principle focal points is that you can get great outcomes when information accessible isn't a lot (two or three thousand labeled examples) and computational assets are rare. Guileless Bayes depends on Bayes' Theorem, which encourages us figure the contingent probabilities of event of two occasions dependent on the probabilities of event of every individual occasion. This implies any vector that speaks to a book should contain data about the probabilities of appearance of the expressions of the content inside the writings of a given classification with the goal that the calculation can register the probability of that content's having a place with the class [4].

### 2.2 Support Vector Machines:

Support Vector Machines (SVM) is only one out of numerous calculations we can look over while doing content grouping. Like innocent bayes, SVM needn't bother with a lot of preparing information to begin giving precise outcomes. In spite of the fact that it needs more computational assets than Naive Bayes, SVM can accomplish increasingly exact outcomes. To put it plainly, SVM deals with drawing a "line" or hyperplane that separates a space into two subspaces: one subspace that contains vectors that have a place with a gathering and another subspace that contains vectors that don't have a place with that gathering. Those vectors are portrayals of your preparation writings and a gathering is a label you have labelled your writings with.

### 2.3 Deep Learning:

Deep learning is a lot of calculations and procedures roused by how the human cerebrum functions. Content characterization has profited by the ongoing resurgence of profound learning structures because of their capability to arrive at high exactness with less need of built highlights. The two primary profound learning designs utilized in content order are Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). From one viewpoint, profound learning calculations require considerably more preparing information than conventional AI calculations, for example in any event a great many labeled models. Then again, conventional AI calculations, for example, SVM and NB arrive at a specific limit where including all the more preparing information doesn't improve their precision [13].

## 3. Naive Bayes Classifier

In machine learning, naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) unconventionality expectationsamong the features. They are among the meekest Bayesian network models. Naïve Bayes classifiers are highly mountable, requiring a number of limitsdirect in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by assessing a closed-form appearance, which takes linear time, rather than by luxurious iterative estimate as used for many other categories of classifiers.

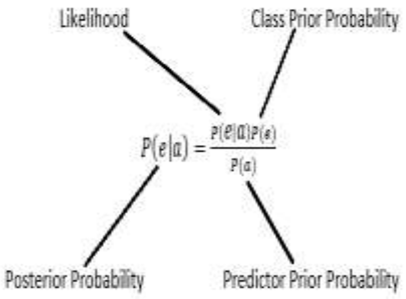
Naive Bayes is a modestprocedure for creating classifiers: models that appoint class marks to issue occurrences, denoted as vectors of feature values, where the class labels are drawn from some predetermined set. There is certifiably not a solitary calculation for preparing such classifiers, yet a group of calculations dependent on a typical guideline: all naive Bayes classifiers assume that the value of a particular feature is autonomous of the value of any other feature, given the class variable. For some types of probability models, naive Bayes classifiers can be proficient very efficiently in a supervised learning setting. In numerousreal-world applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods [5].

## 4. Naive Bayes Algorithm

It is an order method dependent on Bayes' Theorem with a hypothesis of independence among predictors. In modestrelationships, a Naive Bayes classifier undertakes that the occurrence of a precise feature in a class is unconnected to the occurrence of any other feature.For example, a shirt may be considered to be a red if it is full sleeve shirt, warm, and about 42 inches in diameter. Even if these features be contingent on each other or upon the existence of the other features, all of these possessions independently contribute to the probability that this shirt is in redcolour and that is why it is known as 'Naive' [10].

Naive Bayes model is informal to build and predominantly useful for very large data sets. Along with ease, Naive Bayes is known to outperform even extremely cultured classification methods.

Bayes theorem provides a way of calculating posterior probability $P(e|a)$ from $P(e)$, $P(a)$ and $P(a|e)$. Look at the equation below:


Likelihood
Class Prior Probability
$P(e|a) = \frac{P(e|a)P(e)}{P(a)}$
Posterior Probability
Predictor Prior Probability

$$P(e|a)=P(a_1|e)*P(a_2|e)*\ldots\ldots\ldots\ldots*P(a_n|e)*P(e)$$

**Figure 2: Bayes Theorem Calculator**

At this point,

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

### 4.1 Naive Bayes algorithm working:

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

**Step 1:**Translate 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.

**Step 3:** Now, use Naive Bayesian equation to compute the posterior probability for individually class. The class with the maximum posterior probability is the consequence of prediction [9].

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

We can explain it by means of above discussed technique of posterior probability.

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

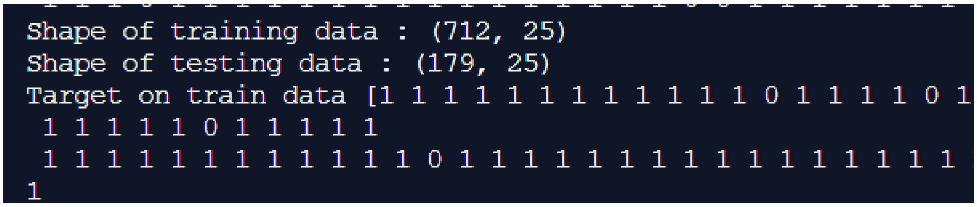
Here we have P (Sunny |Yes) = 3/9 = 0.33,

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.

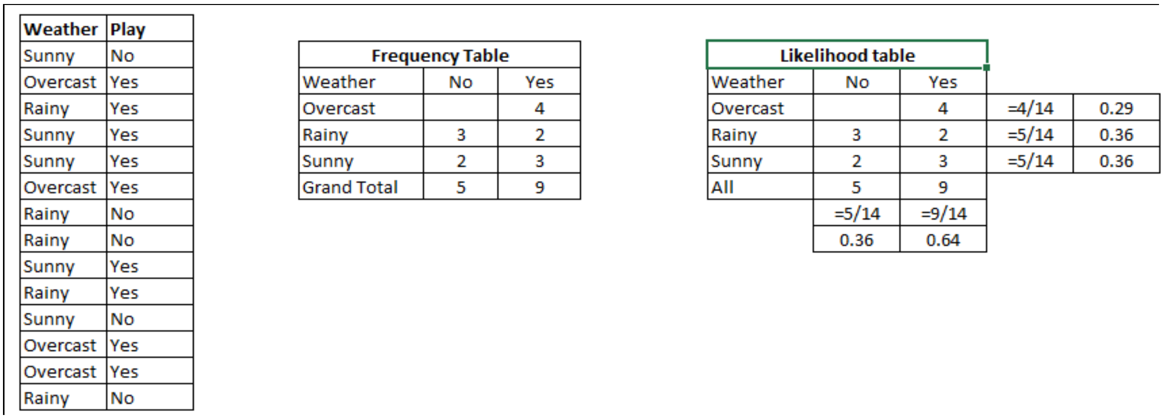
In table 1, the calculations for players are done with the help of naïve bayes equation and the predefined parameters. In figure 3 and figure 4, the outcome of resultant data available in table1 is converted into programming language "Python" and result are produced. It shows the trained data set values and the test data set values along with their accuracy. This predicts that the accuracy of trained data set is more as compared with test data set as it depends upon the selected parameters used by the player at the time of playing.



```
Shape of training data : (712, 25)
Shape of testing data : (179, 25)
Target on train data [1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1
 1 1 1 1 1 0 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1
```

**Figure 3: Training Data set values [12]**

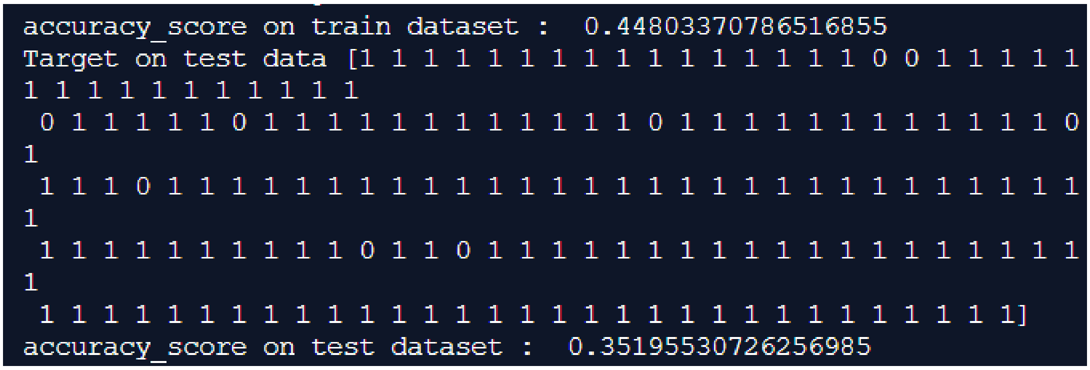
**Table 1: Naive Bayes Algorithm Calculation**



| 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 | | |



```
accuracy_score on train dataset :  0.44803370786516855
Target on test data [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0
1
 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1
 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
accuracy_score on test dataset :  0.35195530726256985
```

Figure 4: Accuracy score of Train dataset and Test dataset

### 4.2 Applications of Naive Bayes Algorithms [6]:

#### 4.2.1 Real time Prediction:

Naive Bayes is an excited learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.

#### 4.2.2 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.

#### 4.2.3 Text classification/ Spam Filtering/ Sentiment Analysis:

Naive Bayes classifiers mostly used in text classification, have higher achievement rate as compared to other algorithms. As a consequence, it is extensively used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)

#### 4.2.4 Recommendation System:

Naive Bayes Classifier and Collaborative Filtering together shapes a Recommendation System that uses machine learning and data mining techniques to filter hiddenmaterial and predict whether a user would like a given resource or not.

## 5. Pros and Cons of Naive Bayes

Following are the main advantages of naïve bayes algorithm:

- It is easy and fast to predict class of test data set. It also performs 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 performs well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed [7].

Here, these are the disadvantages of naïve bayes approach:

- 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 predictive probability 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 [8].

## 6. Conclusion

This paper presented an efficient technique for text classification. In this article, we looked at one of the supervised machine learning algorithms “Naive Bayes” mainly used for classification. We presented the text classifier types and explain the machine learning based text classifier i.e naïve by taking an example.

## 7. Future Scope

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.

## References

[1] Chai, K.; H. T. Hn, H. L. Chieu; “Bayesian Online Classifiers for Text Classification and Filtering”, Proceedings of the 25th annual international ACM SIGIR conference on Research and Development in Information Retrieval, August 2002, pp 97-104.

[2] Elkan C. Magical Thinking in Data Mining: Lessons From CoIL Challenge 2000, Department of Computer Science and Engineering, University of California, San Diego, USA, 2001.

[3] Chowdhury Mofizur Rahman, Ferdous Ahmed Sohel, Parvez Naushad, Kamruzzaman S. M, “Text Classification Using the Concept of Association Rule of Data Mining,” In Proceedings of International Conference on Information Technology, Kathmandu, Nepal, pp 234-241, May 23-26, 2003.

[4] Vishal Gupta and Gurpreet S. Lehal, “A Survey of Text Mining Techniques and Applications”, Journal of Emerging Technologies in Web Intelligence, Vol. 1, No. 1, 2009, pp. 60-76

[5] Monika D Khatri, S.S Dhande, “Implementation of Text Mining with auxiliary Information using classification”, International Journal for Technological Research in Engineering, Vol. 2, Issue 10, 2015, pp. 2387-2393.

[6] Xu, B.; Guo, X.; Ye, Y.; Cheng, J. An Improved Random Forest Classifier for Text Categorization. JCP **2012**, 7, 2913–2920.

[7] Gupta, V.; Lehal, G.S. A survey of text mining techniques and applications. J. Emerg. Technol. Web Intell. **2009**, 1, 60–76.

[8] Pahwa, B.; Taruna, S.; Kasliwal, N. Sentiment Analysis-Strategy for Text Pre-Processing. Int. J. Comput. Appl. **2018**, 180, 15–18.

[9] Mawardi, V.C.; Susanto, N.; Naga, D.S. Spelling Correction for Text Documents in Bahasa Indonesia Using Finite State Automata and Levinshtein Distance Method. EDP Sci. **2018**, 164.

[10] Spirovski, K.; Stevanoska, E.; Kulakov, A.; Popeska, Z.; Velinov, G. Comparison of different model’s performances in task of document classification. In

Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, Novi Sad, Serbia, 25–27 June 2018; p. 10.

[11] Singh, K.; Kaur, R.; Kumar, D. Comment Volume Prediction Using Neural Networks and Decision Trees. In Proceedings of the 2015 17th UKSIM'15 UKSIM-AMSS International Conference on Modelling and Simulation, IEEE Computer Society, Washington, DC, USA, 25–27 March 2015; pp. 15–20.

[12] T. Singh, M. Kumari, T. L. Pal and A. Chauhan, "Current Trends in Text Mining for Social Media", Int. J. Grid Distrib. Comput., vol. 10, no. 6, (2017), pp. 11-28.

[13] Patel, Ankit Dilip, and Yogesh Kumar Sharma. "Web Page Classification on News Feeds Using Hybrid Technique for Extraction." Information and Communication Technology for Intelligent Systems. Springer, Singapore, 2019. 399-405.

[14] Hadni, M., Lachkar, A., & Ouatik, S.A. A new and efficient stemming technique for arabic text categorization. IEEE International Conference on Multimedia Computing and Systems (ICMCS), 2012.

[15] Dr. Yogesh Kumar Sharma, Ranjeet Kaur, "Stock market prediction analysis: a review", International Journal of Scientific Research and Engineering Development, Volume 2 Issue 6, Nov-Dec 2019.