Naive Bayes Classifier in Natural Language Processing

Pranav Manjunath[1]

[1]Duke University

Abstract

Naive Bayes algorithm is a probabilistic classifier that is significantly used in Natural Language Processing, mainly in text classification. This paper will outline the mathematical concept, a written example of the Naive Bayes Algorithm in the context of text classification, as well as a Python implementation using the "20 newsgroups text" dataset of Scikit Learn.

1 Introduction

Text classification is the task of labelling a set of predefined categories to textual data. This task can be used in all different domains that includes textual information. For example, in the entertainment industry, given a particular text dialogue, one could use text classifiers to classify the dialogue by the actor. Another example of text classification would be classifying newspaper articles into its respective category (Tech, Business, Health, etc). Text classification could be used in sentiment analysis, topic labeling, spam detection, and intent detection. Once the classifier has been trained to classify text into its respective categories, it can be used to then predict in which category a new textual information belongs to. Before using an algorithm to classify the text, data pre-processing of text takes place. A few text pre-processing includes stemming, lemmatization, stop-word removal, and text enrichment/augmentation. The pre-processed text is the input to the text classifier.

There are multiple algorithms that can be used for text classifications. In this research paper, I have decided to focus on the Naive Bayes Algorithm. This paper demonstrates a live example of how Naive Bayes is used text classification as well as a Real World example using Scikit Learn Library.

2 Algorithm

2.1 Naive Bayes Algorithm

Naive Bayes classifiers represents a supervised learning method and a statistical method for classification. It is a probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumption between the features. Bayes' Theorem is stated below:

$$P(Y|X) = \frac{P(X|Y) * P(Y)}{P(X)} \tag{1}$$

Bayes theorem provides a way of calculating the posterior probability, P(Y|X), from the likelihood P(X|Y), prior P(Y), and the evidence P(X).

The Bayes Rule provides the formula for the probability of Y given X. However, when there are multiple X's and the Bayes Rule can be extended to Naive Bayes. The major assumption of Naive Bayes is that the X's are independent of each other, also known as class conditional independence.

In other words, Naive Bayes classifier assumes that the effect of the value of a predictor (X) on a class (Y) is independent of the values of other predictors. In mathematical notation,

$$X = (X1, X2, X3..Xn) \tag{2}$$

In Eq (2), X is the sentence and X1....Xn are the words in the sentence. The probability of a sentence occurring is the product of the probability of each word.

$$P(X) = P(X1) * P(X2) * ..... * P(Xn) \tag{3}$$

The Likelihood P(X|Y) is then represented as

$$P(X|Y) = P(X1|Y) * ...P(Xn|Y) \tag{4}$$

Hence, the Posterior Probability that Sentence X belongs to Class Y is

$$P(Y|X1..Xn) = \frac{P(X1|Y) * ...P(Xn|Y) * P(Y)}{P(X1..Xn)} \tag{5}$$

For all X's, the denominator of the above equation remains constant. Hence, the denominator can be removed and a proportionality can be introduced.

$$P(Y|X1..Xn) \propto P(Y)\Pi_{i=1}^{n} P(X_i|Y) \tag{6}$$

If Y, the class label has 2 or more classes, Naive Bayes can be simplified into

$$Y' = argmax_Y = P(Y)\Pi_{i=1}^{n} P(X_i|Y) \tag{7}$$

where Y' would result in the predicted class of the given text.

2.2 Laplace Smoothing

One of the cases in implementing Naive Bayes in text classification includes working with zero probability. Zero Probability is a case where a word is absent in the training dataset, and hence the likelihood of that word (P(word|Y)=0). This in turn will equate the numerator of Eq. 5 to be zero, resulting in a zero posterior probability. Hence, to deal with Zero Probability, Laplace Smoothing can be used.

Laplace Smoothing, also known as Additive Smoothing, is mathematically expressed as

$$\theta_i = \frac{X_i + \alpha}{N + \alpha d} \tag{8}$$

Where X is an observation $X = (X_1, X_2, ...X_d)$ from Multinomial distribution with N trials, parameter vector $\theta = (\theta_1, .., \theta_d)$ and $\alpha > 0$ is the smoothing parameter. $\alpha = 0$ represents no smoothing. In the context of NLP, N is the total number of words present in the given class and d is the number of unique words present in the dataset.

3 An Example

Consider the following news headings and its appropriate class/category label shown in Table 1. There are three headings that belong to the category Sports and three headings that belong to the category Real Estate.

News Headings	Class Label
Team A wins the Cricket World Cup Trophy	Sports
NBA Major Player ruled out for All Star Basketball Game	Sports
Team B have their QB plan after Player X trade	Sports
Star Player puts NJ house for sale after Team B trade	Real Estate
NBA player lists house for sale in LA	Real Estate
See Inside the house of LA Basketball Star	Real Estate

Now let us say that there is a new news heading which the model has to classify into the categories Sports or Real Estate. Using Naive Bayes Classifier, we can determine the probability that the news heading belongs to the Sports and Real Estate Categories. Let us say that the new news heading is

New Heading: "Star House For Sale After World Cup"

The aim is to identify the probability that the new heading belongs to Sports and probability that the new heading belongs to Real Estate.

3.1 Probability the heading belongs to Sports

$$P(Sports|NewHeading) = \frac{P(NewHeading|Sports) * P(Sports)}{P(NewHeading)}$$

P(Sports) would be the number of Sports Headings by the total number of headings. There are a total of 6 headings out of which 3 of them belong to the category Sports. Hence,

$$P(Sports) = \frac{3}{6}$$

As shown in Eq. 4, we can write the prior as

$$P(NewHeading|Sports) = P(Star|Sports) * P(Lists|Sports) * P(House|Sports) * P(After|Sports) * P(World|Sports) * P(Cup|Sports)$$

Each part of Equation RHS can be equated by calculating the frequency of the word in the category Sports. For example $P(Star|Sports) = 1/27$. There are total of 27 words in the Sports Category, out of which there is only one occurrence of Star. Calculating for the other words, we get

$$\begin{array}{ll} \text{P(Star}\|Sports) = 1/27 & \text{P(Lists}\|Sports) = 0/27 \\ \text{P(House}\|Sports) = 0/27 & \text{P(After}\|Sports) = 1/27 \\ \text{P(World}\|Sports) = 1/27 & \text{P(Cup}\|Sports) = 1/27 \\ \text{P(For}\|Sports) = 1/27 & \text{P(Sale}\|Sports) = 0/27 \end{array}$$

It can be seen that the problem of Zero Probability exists in this case. Hence using Laplace Smoothing with $\alpha = 1, N = 27, d = 36$. For Example, $P(Star|Sports) = (1+1)/(27+36)$. Using Laplace Smoothing to the other words, we get

$$\begin{array}{ll} \text{P(Star}\|Sports) = 2/63 & \text{P(Lists}\|Sports) = 1/63 \\ \text{P(House}\|Sports) = 1/63 & \text{P(After}\|Sports) = 2/63 \\ \text{P(World}\|Sports) = 2/63 & \text{P(Cup}\|Sports) = 2/63 \\ \text{P(For}\|Sports) = 2/63 & \text{P(Sale}\|Sports) = 1/63 \end{array}$$

$$P(NewHeading|Sports) = \frac{2}{63} * \frac{1}{63} * \frac{1}{63} * \frac{2}{63} * \frac{2}{63} * \frac{2}{63} * \frac{2}{63} * \frac{1}{63} = \frac{32}{63^8}$$

$$P(Sports|NewHeading) = P(NewHeading|Sports) * P(Sports) = \frac{32}{63^8} * \frac{3}{6} = \mathbf{\frac{16}{63^8}}$$

3.2 Probability that New Heading belongs to Real Estate

$$P(RealEstate|NewHeading) = \frac{P(NewHeading|RealEstate) * P(RealEstate)}{P(NewHeading)}$$

P(Real Estate) would be the number of Real Estate Headings by the total number of headings. There are a total of 6 headings out of which 3 of them belong to the category Real Estate. Hence,

$$P(RealEstate) = \frac{3}{6}$$

Therefore,
As shown in Eq. 4, we can write the prior as

$$P(NewHeading|RealEstate) = P(Star|RealEstate) * P(Lists|RealEstate) * P(House|Sports) * P(Cup|RealEstate) * P(After|RealEstate) * P(World|RealEstate)$$

Each part of above equation in the RHS can be equated by calculating the frequency of the word in the category Real Estate. For example P(Star| Real Estate) = 1/31. There are total of 31 words in the Real Estate Category, out of which there is only one occurrence of Star. Calculating for the other words, we get

$$\begin{array}{ll} \text{P(Star}\|RealEstate) = 1/27 & \text{P(Lists}\|RealEstate) = 1/27 \\ \text{P(House}\|RealEstate) = 3/27 & \text{P(After}\|RealEstate) = 1/27 \\ \text{P(World}\|RealEstate) = 1/27 & \text{P(Cup}\|RealEstate) = 1/27 \\ \text{P(For}\|RealEstate) = 2/27 & \text{P(Sale}\|RealEstate) = 2/27 \end{array}$$

It can be seen that the problem of Zero Probability does not exists in this case. Hence, Laplace Smoothing does not need to be used in this case.

$$P(NewHeading|RealEstate) = \frac{1}{27} * \frac{1}{27} * \frac{3}{27} * \frac{1}{27} * \frac{1}{27} * \frac{1}{27} * \frac{2}{27} * \frac{2}{27} = \frac{12}{27^8}$$

$$P(RealEstate|NewHeading) = P(NewHeading|RealEstate) * P(RealEstate) = \frac{12}{27^8} * \frac{3}{6} = \mathbf{\frac{6}{27^8}}$$

3.3 Classification of New Heading

Once the Posterior Probability is calculated for both categories, the category with the highest posterior probability would be label for the New Heading (Mathematically expressed in Eq 7). In this case, we can see that, P(New Heading|Real Estate)>P(New Heading|Sports)

$$\frac{6}{27^8} > \frac{16}{63^8}$$

Hence, the New Heading = "Star House For Sale After World Cup" would be classified in the Real Estate Category by the Naive Bayes Classifier.

4 Python Implementation

In Python, Naive Bayes can be implemented using the MultinomialNB package from the Scikit Learn library. The "20 newsgroups text" dataset in the Scikit Learn Library consists of 18846 newsgroups posts on 20 topics and is split into a 40% testing split.

For this example, text pre-processing has not been done. The News Articles are first converted into the Term Frequency - Inverse Document Frequency Matrix (TF-IDF). The TF-IDF matrix indicates the importance of a word among multiple documents. Upon conversion to a TF-IDF matrix, each document is converted into a $1xn$ matrix where n is the number of unique words in all the documents. The values for each word in the TFIDF matrix represents the importance of each word. This matrix is then sent as input to the MultinomialNB model. The MultinomialNB model is trained using the 60% training data (11314 posts). For comparison, I have also trained a baseline K-Nearest Neighbors Algorithm and Multinomial Logistic Regression Model to the training dataset.

4.1 Results

The bar plot (Figure 1) below describes the testing metrics of the three models.

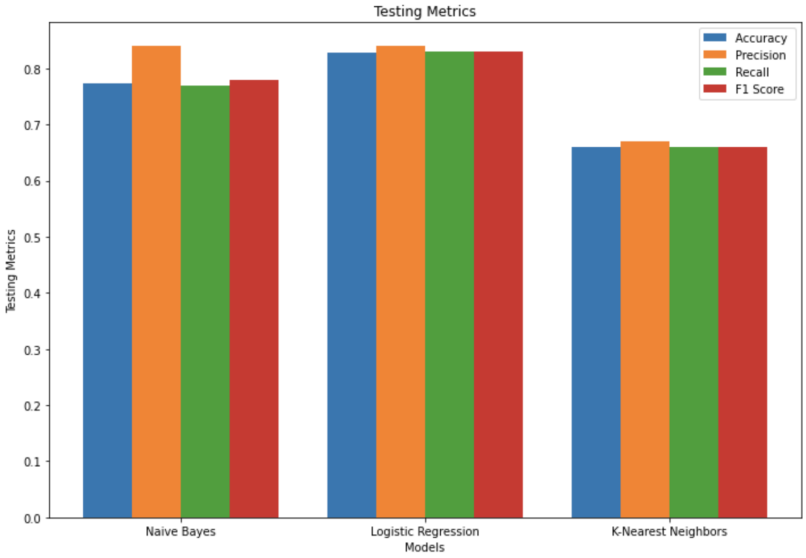

Figure 1: Model Testing Metrics

On average, it can be seen that the Logistic Regression Model performs better than the Naive Bayes which performs better than the K-Nearest Neighbors model. Interestingly, Naive Bayes Algorithm has a higher precision when compared to the other models. In the real world, Naive Bayes Algorithm performs well and is used extensively to solve real world problems.

Figure 2 graphically represents the training and testing time taken by the three models.

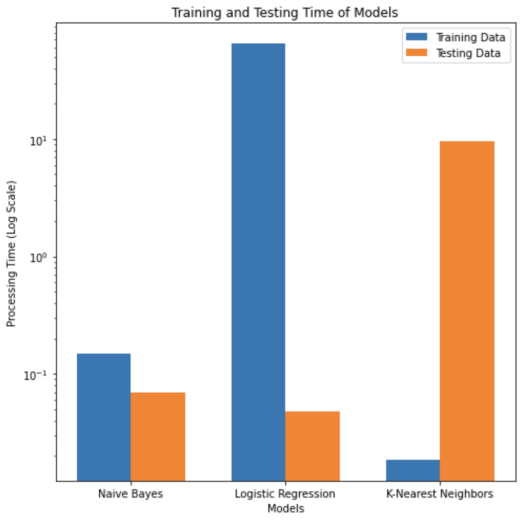

Figure 2: Model Processing Time

When looking at the total processing time (training + testing), it can be seen that Naive Bayes takes the least time when compared to the other two models. While there is a large difference between the training and testing time with the Logistic Regression and K-Nearest Neighbor model, the Naive Bayes Model interestingly does not have the large difference.

5 Advantages and Disadvantages of Naive Bayes Algorithm

5.1 Advantages

1. Naive Bayes requires a small amount of training data to estimate the test data. Hence, the training period is less.
2. Naive Bayes is also easy to implement.
3. Naive Bayes is suitable for solving multi-class prediction problems
4. Naive Bayes is better suited for categorical input variables than numerical variables.

5.2 Disadvantages

1. Naive Bayes assumes that all predictors (or features) are independent.
2. The problem of 'Zero Probability' occurs where it assigns zero probability to a categorical variable where the category in the test data set is not present in the training dataset.

6 Conclusion

Naive Bayes Algorithm is extensively used in Natural Language Processing to help classify text into its respective class labels. This algorithm follows Bayes Theorem, where the aim is to identify the posterior probability of a text belonging to a class P(Y|X) when given the prior probability of the class P(Y) and the likelihood P(X|Y). The special case of Naive Bayes is when there are multiple values of X and they are all independent of each other. This paper looks at an example where the goal is to classify a News Heading "Star House For Sale After World Cup" into either the category of Sports or Real Estate. Using Naive Bayes, the algorithm classified this News Heading into the category Real Estate. Further, the paper demonstrates a real world example of Naive Bayes Algorithm using Python and comparing the results with the Logistic Regression and K-Nearest Neighbors models. In terms of average testing metrics, while the Naive Bayes (Accuracy = 0.77) performed better than the K-Nearest Neighbors (Accuracy = 0.66), the Logistic Regression model performed the best with an accuracy of 0.84 (84%). One method to achieve better model performance would be to work extensively on the text pre-processing before inputting the text into the model.

References

[1] Yuguang Huang, Lei Li. *Naive Bayes Classification Algorithm Based on Small Sample Set.* Beijing University of Posts and Telecommunications, Beijing, China, 2011.

[2] Huma Parveen, Prof. Shikha Pandey. *Sentiment Analysis on Twitter Data-set Using Naive Bayes Algorithm.* Rungta College of Engineering and Technology Bhilai, India, 2016.

[3] Haiyi Zhang, Di Li, *Naïve Bayes Text Classifier.* IEEE International Conference on Granular Computing, 2007.