International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

Finding the original writer of an anonymous text using Naïve Bayes Classifier

Noorul Amin, S V Athawale

Student, Dept. of Computer Engineering, AISSMS COE, Maharashtra, India
Professor, Dept. of Computer Engineering, AISSMS COE, Maharashtra, India

---

Abstract - *Many a time we come across some texts which are written by some anonymous writers. These texts can either be in the form of books, emails, messages, or blogs. We know that every writer has a unique style of writing. One simple way to find this uniqueness is to list the words the writer has used and then find the frequency of each word. Each writer will have a different set of frequencies for different words. If we match it with frequencies of the word of the known writer then we can predict the original writer.*

***Key Words*: anonymous writer, frequency, writer's prediction**

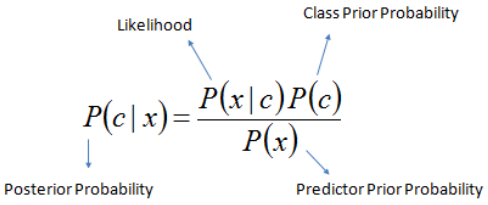
1. INTRODUCTION

Writer identification has become very difficult in this digital world. People are making more use of the digital medium to write. Hence it becomes more difficult to detect the writer of the literary text if the name of the original writer is not specified. If the text would have handwritten it would have been easy to find the original writer. Many research has been done on handwritten text's writer identification [1] but, printed text's writer identification requires a different approach. Hence, we would employ some machine learning algorithms to solve this problem. We would use the Naive Bayes algorithm which is a supervised machine learning algorithm. Naïve Bayes is suitable for text based classification. Hence, we will employ it to solve our current problem.

1.1 Naïve Bayes

Naive Bayes belongs to a family of supervised learning machine algorithms based on Bayes' theorem with the strong assumption of conditional independence between every pair of features.

- P(c|x) = posterior probability of class given predictor.
- P(c) = prior probability of class.
- P(x|c) =probability of predictor given class.
- P(x) = prior probability of predictor.



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

Naïve Bayes is a linear classifier that is very efficient. Naive Bayes is easy to build and works well with large data set. Making the assumption that each feature is independent may feel unrealistic in practice but still result calculated on the basis of this assumption can outperform the more powerful algorithms like SVM and Decision Tree. Naïve Bayes classifier is generally applied for text based operations like fake news detection [3], email spam detection [4] and authorship attribution [2] and in forensics.

1.2 Solution

Every document or text can be represented as a bag of words, which is a set of unordered words used in the sentences, ignoring its position in the sentence but keeping its frequency in the text.

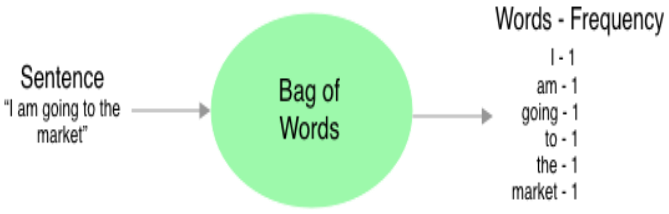


Fig -1: 'Bag of Words' Model

In this way, we will get the word frequency table. Then we will convert the frequency table to the likelihood table, which will give the probability of the occurrence of a word in the text. From this posterior probability is calculated using the Naive Bayesian equation for each class. The class with the highest posterior probability will be the result of the prediction.

1.3. Pros and Cons of using Naive Bayes

Pros

- Easy and fast to predict the class of the dataset.
- Requires less training data.
- Have a very high accuracy rate.

Cons

- If categorical variable features a class, which was not observed in the training data set, then the model will assign a 0 probability and will not be able to make a prediction.

2. Naïve Bayes compared to SVM and Decision Tree Algorithms

Clearly, Naïve Bayes does the job of classification very well. But what is its performance as compared to other supervised algorithms? When working on Enron datasets, containing 500000 emails, I measured the performance of these three algorithms in terms of training time, prediction time and accuracy.

Algorithms	Accuracy	Training Time	Prediction Time
Naïve Bayes	92.0364050057%	0.2450 s	0.021 s
Linear SVM	95.9613196815%	23.484 s	2.091 s
RBF SVM	97.6109215017%	12.982 s	0.935 s
Decision Tree	96.7007963595%	4.715 s	0.003 s

Table -1: Performance of different algorithms

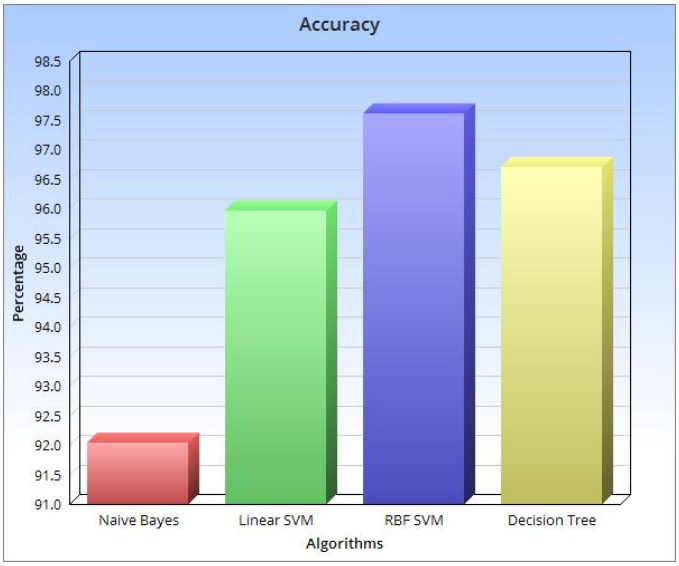


Chart -1: Accuracy of different algorithms

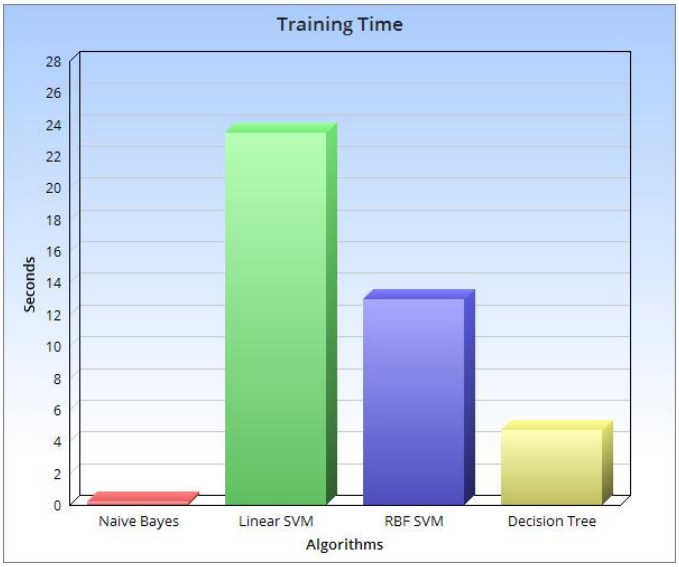


Chart -2: Training time of different algorithms

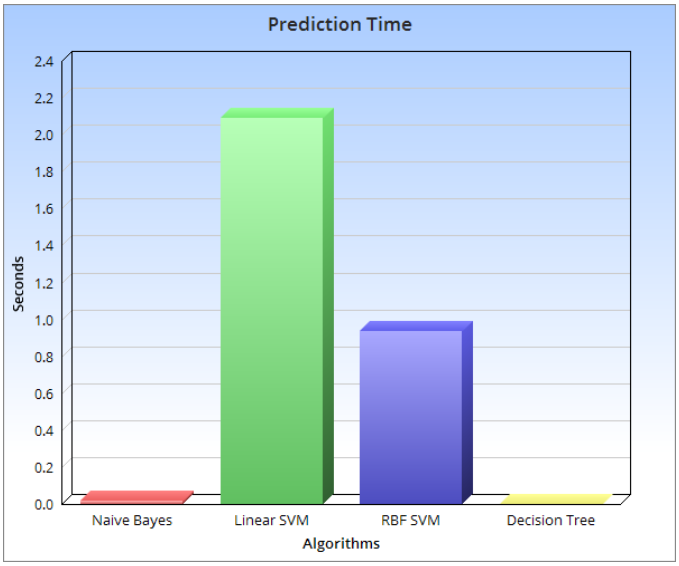


Chart -3: Prediction time of different algorithms

Since performance of any of these algorithms is

- $\propto$ Accuracy
- $\propto$ 1/(Training Time + Prediction Time)

Hence, Performance of Algorithm,

ρ = (K*Accuracy)/(Training Time + Prediction Time)

If K = 1, then

ρ = (Accuracy)/(Training Time + Prediction Time)

Here, ρ is the performance factor.

Hence greater the value of ρ better is the algorithm.

If we plot the ρ of different algorithms then as expected, ρ for Naïve Bayes is highest.

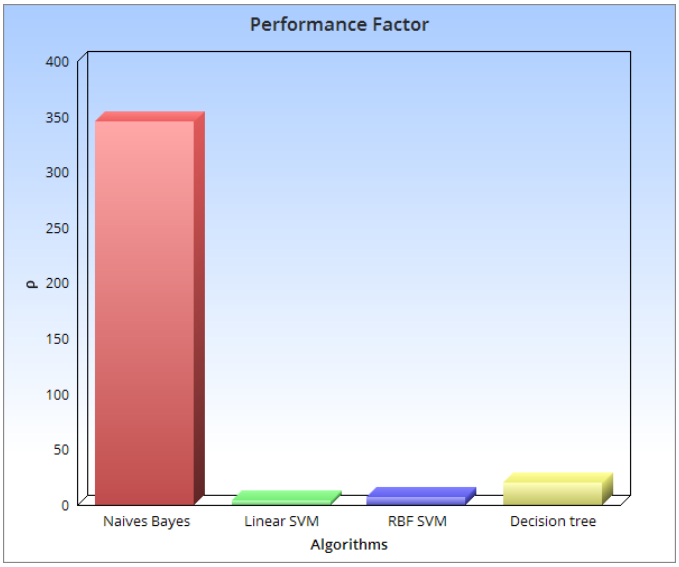


Chart -3: Performance Factor of different algorithms

From all these charts we can infer that Naive Bayes may not give the best accuracy, but it gives the lowest combined time of training time and prediction time. Accuracy for this algorithm greatly depends on the size of training data sets. With the increase in data set size, the accuracy of this algorithm increases such that after some point it will get past these algorithms in both accuracy and time scores.

Hence, for very large data sets it is the best algorithm to use considering both time and accuracy.

3. CONCLUSIONS

Hence, we can predict the original writer of an anonymous text using Naive Bayes algorithm with high accuracy given that we have the set of texts of that writer for reference.

ACKNOWLEDGEMENT

I would like to thank Prof. S. V. Athawale Sir, under whose guidance I was able to complete this journal. I would also like to thank my group mates for their moral support.

REFERENCES

[1] Stefan Fiel and Robert Sablatnig," Writer Retrieval and Writer Identification using Local Features", 2012 10th IAPR International Workshop on Document Analysis Systems, Year: 2012, Pages: 145 – 149.

[2] Fatma Howedi and Masnizah Mohd, "Text Classification for Authorship Attribution Using Naive Bayes Classifier with Limited Training Data", Computer Engineering and Intelligent Systems ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)Vol.5, No.4, 2014.

[3] Mykhailo Granik and Volodymyr Mesyura, "Fake news detection using naive Bayes classifier", 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Year: 2017, Pages: 900 - 903

[4] Anirudh Harisinghaney, Aman Dixit, Saurabh Gupta and Anuja Arora, "Text and image based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm", 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), Year:2014, Pages:153 – 155.

[5] 6 Easy Steps to Learn Naive Bayes Algorithm (with codes in Python and R.). Retrieved 26 March, 2019, from https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ .

[6] Naive Bayes and Text Classification. Retrieved 26 March, 2019, from https://sebastianraschka.com/Articles/2014_naive_bayes_1.html .

[7] A practical explanation of a Naive Bayes classifier. Retrieved 26 March, 2019, from https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/ .

[8] Gareth James; Daniela Witten; Trevor Hastie Robert Tibshirani, "An Introduction to Statistical Learning with Applications in R", 2013.

[9] Andreas C. Mueller and Sarah Guido, "Introduction to Machine Learning with Python", O'reilly, 2016.

[10] Allen B. Downey, "Think Bayes Bayesian Statistics Made Simple", O'reilly, 2012.