# Spam Classifier

Rishabh Chaturvedi[1], Harsh Raj[2]
[1,2] *Student, B.Tech, Computer Science, Galgotias University*

***Abstract*— SMS classification is detecting spam and ham using Naïve Bayes classifier. Naive Bayes classifier is a machine learning algorithm which uses the Bayes theorem in solving a classification problem. It's one of the simplest probabilistic models and it is used as a benchmark for comparing the performance of other classification algorithms. This algorithm is prefixed with the word 'Naive' since this algorithm strongly assumes the conditional independence between the features. This project incorporates the techniques of Naive Bayes classification to classify spam and ham messages**

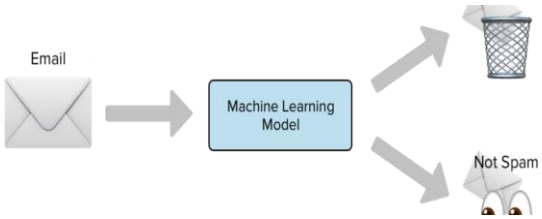
## 1. INTRODUCTION

Naive Bayes is a simple yet heavily used classification technique especially in the realm of text data. In this project we are been given a text dataset which contains SMS and its corresponding class label which is "SPAM" and "HAM". HAM is basically the messages that aren't spam ("NOT SPAM"). Principle of Naive Bayes Classifier: A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Using Bayes theorem, we can find the probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

Naive Bayes classifiers are a popular statistical technique of email filtering. They typically use bag of words features of identify spam email, an approach commonly used in text classification. Naive Bayes classifiers works by correlating the use of tokens, with spam and non-spam emails and then using Bayes theorem to calculate a probability than an email is or is not spam.




Naive Bayes spam filtering is a baseline technique for dealing with spam that can tailor itself to the email needs of individual users and give low false positive spam detection rates that are generally acceptable to users. It's is one of the oldest ways of doing spam filtering.

Naive Bayes is anon-linear model. You will see that in python when plotting the prediction boundary which will be very nice curve well separating the non-linearly distributed observations.

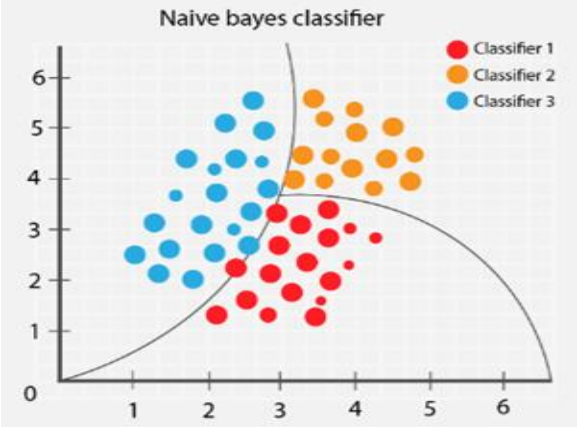
One-way spam emails/SMS are sorted is by using a Naive Bayes classifier. The Naive Bayes algorithm relies on Bayes Rule. This algorithm will classify each object by looking at all of its features individually. Bayes Rule below shows us how to calculate the posterior probability for just one feature. The posterior probability of the object is calculated for each feature and then these probabilities are multiplied together to get a final probability. This probability is calculated for the other class as well. Whichever has the greater probability that ultimately determines what class the object is in.

In machine learning, Naive Bayes classifiers are a family of simply 'probabilistic classifiers' based on applying Bayes theorem with strong independence assumptions between the features.

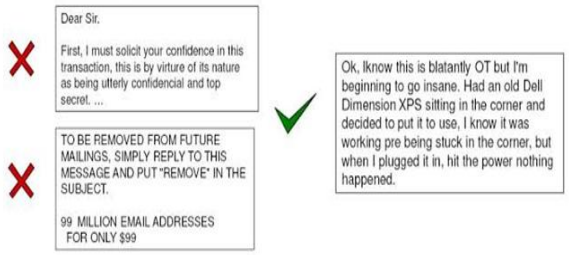
$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Using Bayesian probability terminology, the above equation can be written as,

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$




- We need to find P (message | spam) P(spam) and P (message | -spam) P (-spam)
- The message is a sequence of words (w1, w2,…)
- Bag of words representation
  - ~ The order of words in message is not important
  - ~ Each word is conditionally independent of the order given message Class (spam or not spam)




## 2. LITERATURE SURVEY

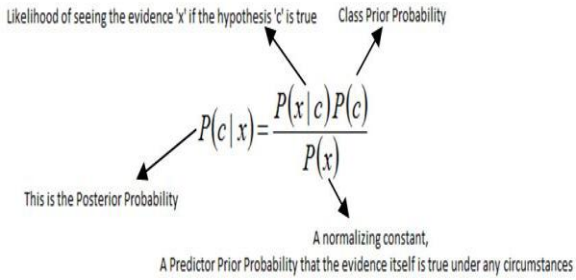
A. Survey on Naive Bayes Short Algorithm
Usage of naïve Bayes and support vector machine to classify text data given a data set of pre classified data for model training.

B. Analysis of Naive Bayes Algorithm for Email Spam Filtering across Multiple Datasets
Referred the related work section from the above research paper which gave me an idea about the formulation of Naïve Bayes classifier.

C. Research of a Spam Filtering Algorithm Based on Naive Bayes and AIS
This research paper focuses on the prioritization of preserving of non-spam mails (ham mails) using Associations rules and NB classifier.

## 3. WORKING OF PROPOSED SYSTEM

We are building a model which is basically a classifier model. It's based on Bayes theorem and is called as Naive Bayes classifier. It is a simplistic, unsophisticated, probabilistic model.
It assumes the conditional independence of each features in course of classification which accounts for its "Naive" prefix. It is based on Bayes theorem with the independent assumption between features.
Naive Bayes classifier assumes that each variable (feature) is independent of each other. It assumes there's no correlation between features.




After, extracting the features from text we can performed the Naive Bayes implementation. The algorithm was trained with the training data to learn the value of the likehood and prior probability.
In test stage, the parameters determined during the training which are used to estimate the probability of sample belonging to different class. Here class is "spam" or "ham".
By, using the above formula we can find,

**a.** P( class = spam | x ) is directly proportional to P( class = spam ).P( x1 | class = spam ).P( x2 | class = spam ) …… P( xn | class = spam ), here spam is a class and P( xi | class = spam ) are the probability learnt in the train phase.

**b.** P( class = ham | x ) is directly proportional to P( class = ham ).P( x1 | class = ham ).P( x2 | class = ham ) …… P( xn | class = ham ), here ham is a class and P( xi | class = ham ) are the probability learnt in the train phase.

Now by solving the above two point(A and B),

- If, A > B :- then the message is spam.
- If, B > A :- then the message is ham.

WHY NAIVE BAYES IS SO NAIVE?

The Naïve Bayes algorithm asumes the conditional independence between the input features which makes it so simple and naïve. Conditional independence is different from the 'NORMAL INDEPENDENCE'.

Events A, B are conditionally independent given a third event C means, suppose you already know that C has happened. Then knowing whether A happened would not convey any further information about whether B happened - any relevant information that might be conveyed by A is already known to you because you know that C happened. We also should note that no correlation does not imply conditional independence. To support my argument, I would like to give an example below.

Let's assume we flip two fair coins. Let A be the event that the first coin heads, B the event that the second coin heads, C the event that the two coins are the same (both heads or both tails). Clearly, A and B have no correlation, but they are not conditionally independent given C - if you know that C has happened, then knowing A tells you a lot about B (indeed, it would tell you that B is guaranteed).

Mathematically Event A and B are conditionally independent given an Event C if:

$P(A \mid B, C) = P(A \mid C)$ and $P(B \mid A, C) = P(B \mid C)$

OR

$P(A, B \mid C) = P(A \mid C) * P(B \mid C)$

Optimisation in Naïve Bayes :

There are two ways in which optimisation in Naïve Bayes algorthim can be done:

1. Laplace Smoothing.
2. Log Probabilities.

Use of Laplace Smoothing:

Laplace smoothing is a very important optimisation for our algorithm. There is a high chance that a word encountered in test data is absent in the train data So the likelihood probability for that data point will be missing in training stage to substitute in the naïve bayes algorithm formula. This smoothing is also known as additive smoothing. For example if a word (feature) – w* is new in train data then the probability that it belongs to say class SPAM is :

$$P(W^* \mid Class) = \frac{0 + \alpha}{N1 + \alpha.K}$$

- N1 = number of data points where class label is spam in train data.
- α= Any numerical value (typically 1). It's also referred to as the hyper parameter of this algorithm. Hyper parameter is a parameter that determines if our model is over fitted or under fitted or well fitted.
- K = distinct values w* can take.

NOTE: As alpha increases model tends to over fit. So, we need to take care that or model doesn't over fit.

Use of Log Probabilities:

If the number of features in dataset is large (which is especially large in case of text data) then its pretty evident that the probability values to be compared would be extremely small as they are result of large multiplication of very small numbers and this can land us in a big trouble. The problem is that there is some precision limit in data types in every programming language. In python the double significant digit is 16 so any number with significant digits greater than 16 will be rounded off by the compiler. This leads to data loss and our model is more prone to errors in this case. The log probability now comes into our rescue. It takes care of our significant digit problem and the probability values can then be easily and more importantly accurately calculated without any loss of data.

## 4. RESULT

The model is successfully implemented and tested on the test data. The test data is obtained by randomly splitting the original data into 80:20 ratio. 80 percent of the randomly sampled data comprises of the train data which is used to train the model to learn the essential parameters to make the prediction in the test stage. The parameters that are learnt involve the likelihood and the prior probabilities. The model's performance on test data is approximately 93 percent which is quite a decent model given the simplicity of Naïve Bayes. So, to put it in simple words the model manages to accurately classify new messages into "spam" or "ham" 93 out of 100 times.

## 5.CONCLUSION

The Naive Bayes Classifier is successfully built using the python open sourced library Scikit learn. Bayes theorem despite being a very old theorem happens to be the most simple, elegant and very useful theorem. I managed to create the model with a highly significant accuracy on a given unseen data point (future message). The best thing about Naïve Bayes is that feature importance can be directly obtained from the model that is selecting those words in a message which are more helpful towards the classification. Feature interpretability is also a bonus for us here in this model. Also, the model can be extended to a multi class classification say we want to classify the ham messages into three categories: FRIENDS, FAMILY AND SCHOOL then we can extend the Naive Bayes binary class classifier to multi class classifier but again we have to do data labeling for our model training. Also, Naive Bayes if used with log probabilities can be efficiently used with large dimensional data. This model is extensively used for Categorical data and is run time complexity and runtime space is also low. These positive points give our model a higher usability rate and is often used as a benchmark for other classification model. No matter how sophisticated other Machine learning algorithms are, Naïve Bayes performs fairly well. It's simple, intuitive and highly interpretable.

## 6. FUTURE SCOPE

1. Integration with some more powerful machine learning algorithm will improve our model's performance.
2. Full Automation of the model.
3. Optimization of the Algorithm to solve a wide range of real-world problem scenario.
4. Training with more data will also account for the increase in our model's performance.

## REFERENCES

[1] http://www.ijaerd.com/papers/finished_papers/Short%20Survey%20on%20Naive%20Bayes%20Algorithm-IJAERDV04I1140826.pdf
[2] http://ieeexplore.ieee.org/document/5709036/
[3] https://www.google.com/imgres?imgurl=https%3A%2F%2Fmiro.medium.com%2Ffit%2Fc%2F1838%2F551%2F0*545fgyQm2_5dja6T.png&imgrefurl=https%3A%2F%2Fmedium.com%2F%40nonthakon&tbnid=KnBPNXaLuNGqHM&vet=12ahUKEwiAt-7O44bpAhUE5zgGHYKwCakQMygAegQIARAX..i&docid=YFqixcNl0uDSGM&w=1838&h=551&q=spam%20email%20machine%20learning&ved=2ahUKEwiAt-7O44bpAhUE5zgGHYKwCakQMygAegQIARAX
[4] https://www.google.com/imgres?imgurl=https%3A%2F%2Fmiro.medium.com%2Fmax%2F1280%2F1*7lg_uLm8_1fYGjxPbTrQFQ.png&imgrefurl=https%3A%2F%2Fbecominghuman.ai%2Fnaive-bayes-theorem-d8854a41ea08&tbnid=fRVvwDYihPWhAM&vet=12ahUKEwiz3Oj_4obpAhUZHXIKHfiWDbEQMygEegUIARCHAg..i&docid=mAsy-dxL1myh1M&w=640&h=206&q=naive%20bayes%20formula&ved=2ahUKEwiz3Oj_4obpAhUZHXIKHfiWDbEQMygEegUIARCHAg
[5] https://www.google.com/imgres?imgurl=https%3A%2F%2Fmiro.medium.com%2Fmax%2F6190%2F1*39U1Ln3tSdFqsfQy6ndxOA.png&imgrefurl=https%3A%2F%2Ftowardsdatascience.com%2Fintroduction-to-na%25C3%25AFve-bayes-classifier-fa59e3e24aaf&tbnid=zb_lFeZe0yYc0M&vet=12ahUKEwjdu4js5IbpAhW48DgGHZowAPUQMygAegUIARCLAg..i&docid=GufP6WjAQq7-YM&w=3095&h=1549&q=naive%20bayes%20classifier&ved=2ahUKEwjdu4js5IbpAhW48DgGHZowAPUQMygAegUIARCLAg
[6] https://datavedas.com/wp-content/uploads/2018/01/image003-11.jpg