# ASSIGNMENT 5

## Naive Bayesian Classifier

In this assignment , the naive bayesian classifier has been used to detect whether :-

1. an email is spam or not
2. to identify a river in a given image.

Note :- We have applied Laplace smoothening to the Naive Bayesian Classifier as otherwise the probabilities for words which have not been seen before is evaluated to be 0 which in turn makes the product or probabilities of words 0 as well. In Laplace smoothening we basically add 1 to frequencies of all the words so as to avoid 0 probabilities.

### *Spam Email Dataset*

For the classification we are using Bayes' theorem, which is given by-

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

Hence for this case we can write-

***P(ham | EmailText) = (P(ham) * P(EmailText | ham)) / P(EmailText)***
***P(spam | EmailText) = (P(spam)* P(EmailText | spam)) / P(EmailText)***

We can say that if ***P(spam | EmailText) > P(ham | EmailText)*** then the Email is most likely a Spam Email.

Since both ***P(spam | EmailText)*** and ***P(ham | EmailText)*** have ***P(EmailText)*** in the denominator we can ignore this term.

We have,

***P(ham) = no of Emails belonging to category ham / total no of Emails***
***P(spam) = no of Emails belonging to category spam / total no of Emails***

***P(EmailText | spam) = P(word1 | spam) * P(word2 | spam) * .... (1)***
***P(EmailText | ham) = P(word1 | ham) * P(word2 | ham) * .... (2)***

Where
***P(word1 | spam) = count of word1 belonging to category spam / total count of words belonging to category spam.***

***P(word1 | ham) = count of word1 belonging to category ham / total count of words belonging to category ham.***

Now, if a word appears in the test case that has not appeared at all during the training cases-

***P(word1 | spam) = 0***

***P(word1 | ham) = 0***

Hence (1) and (2) would be zero and hence both probabilities would be zero.

To avoid this we take log on both sides and apply Laplace Smoothing.

Thus we calculate the logarithm of the probabilities as follows-

***log(P(ham | EmailText)) = log(P(ham)) + log(P(EmailText | ham))***

***log(P(spam | EmailText)) = log(P(spam)) + log(P(EmailText | spam))***

Where

***log(P(EmailText | ham))=log(P(word1 | ham))+log(P(word2 | ham)) ...***

***log(P(EmailText | spam))=log(P(word1|spam))+log(P(word2|spam)) ...***

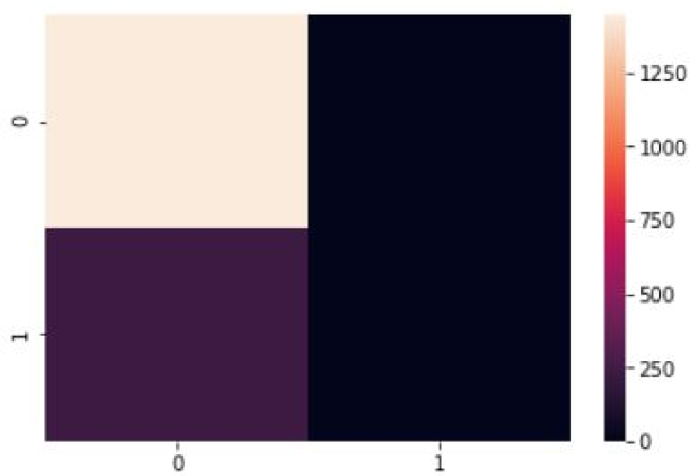
And

***P(word1 | ham) = (count of word1 belonging to category ham + 1) / (total count of words belonging to ham + no of distinct words in training data sets i.e. our database)***

***P(word1 | spam) = (count of word1 belonging to category spam + 1) / (total count of words belonging to spam + no of distinct words in training data sets i.e. our database)***

We first preprocess the dataset using to remove any special characters and convert all characters to lower case characters. We then create a dictionary of the words in spam emails as well as a dictionary of the words in the not spam email and then use bayes theorem to predict whether a given email is spam or not using these dictionaries.In a way it can be said that these dictionaries are our trained model.

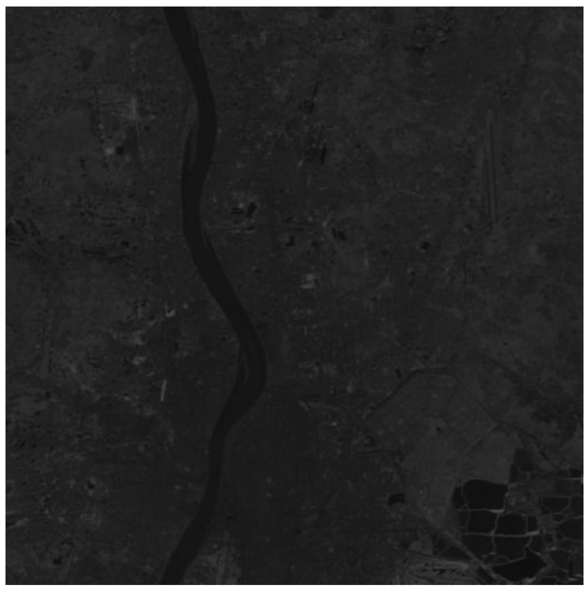
We predict the accuracy using a confusion matrix.



The accuracy obtained is 86.48% for the given test and train dataset.

## *River Dataset*

One of the sample images used for training is as follows-



We then obtain a set of coordinates that the river is present at and that no river is present at respectively and store the intensity at these points along with the number of points of that intensity.

***P(River | Intensity) = (P(River) * P(Intensity | River)) / P(Intensity)***

***P(NotRiver|Intensity)=P(NotRiver)*P(Intensity|NotRiver)/P(Intensity)***

We can ignore the denominator as it is the same in both terms.

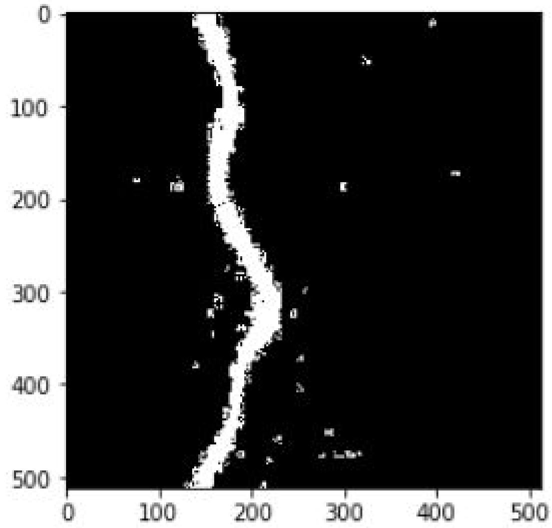
***P(River) = ½***
***P(NotRiver) = ½***

***P(Intensity | River) = (count of points of that intensity belonging to category River + 1) / (total count of points belonging to River + no of distinct points in training data sets i.e. our database)***

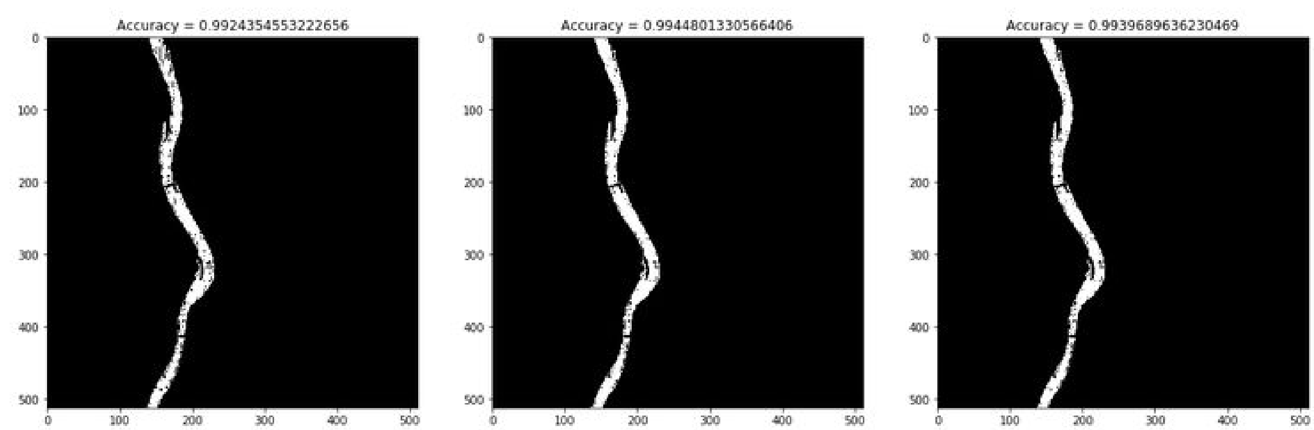
***P(Intensity | NotRiver) = (count of points of that intensity belonging to category NotRiver + 1) / (total count of points belonging to NotRiver + no of distinct points in training data sets i.e. our database)***

We first obtain the coordinates of 50 river points and 100 background points from one of the images. We then apply a mask to the input images and prepare a dictionary for the river and the non-river points using the intensities of the pixels corresponding to the obtained coordinates. Following this, we read a test image as an input and use the bayes theorem on each pixel to compare the probabilities of a point being a river or a non-river points. If it is a river point we make the intensity 0 in the output image and if it is a non-river point we make the intensity 1.

The output obtained on using a naive bayesian classifier in a masked input image is :-



If however, we do not wish to mask the input image, we have to use the Bayes Decision Rule instead of the Naive Bayesian Classifier to obtain a correct output. On using the Bayes Decision rule to predict probabilities on non-masked input images, the output looks like :-



While the Naive Bayesian classifier gives an accuracy of 88.75%(on a masked input image), the Bayes Decision Rule gives a higher accuracy of around 99.42%.