NAIVE BAYES

1 QUESTION NO 1 (SPAM AND HAM MESSAGE CLASSIFICATION)

After Data import, I am curious that which kind of messages are of spam category. So, I call the library "textcat " then run the code

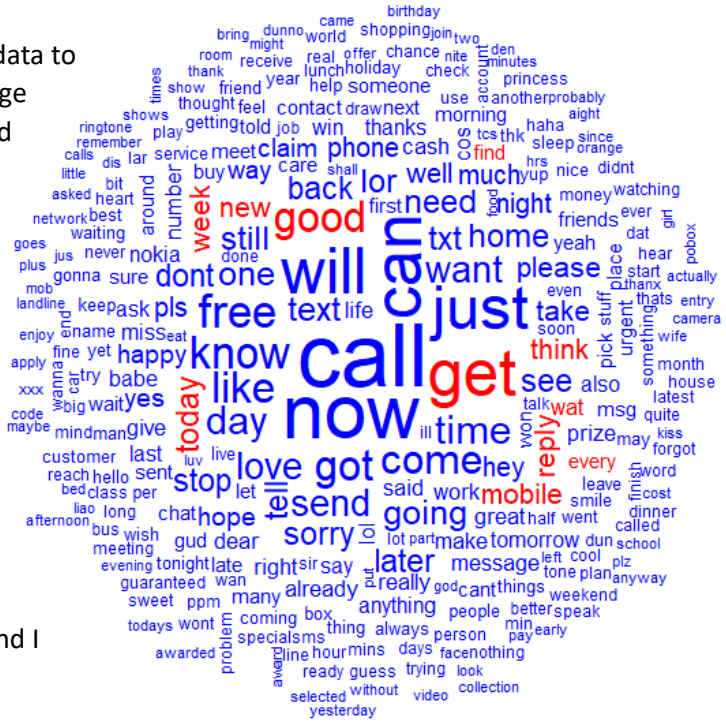
table(textcat(x = message$type),message$type)

	ham	spam
latvian	0	747
norwegian	4812	0

From here itself I am able to classify whether the message is Spam or Ham, we can say that the messages are categories as Latvian can be classified as Spam. This was a try hit method, not applicable each and every time.

But here our major focus is to go for our Naïve Bayes method,

- Converted the message to a Carpus using the library tm
- Cleaning of data as performed in Text mining i.e. data pre processing
- Convert the corpus to Document Term Matrix
- Perform train test split in the original data, corpus data and as well as the Document Term Matrix data.
- Perform WordCloud with the Original train data to see the frequent words used in spam message (in **Red** colour) as well as the frequently used words in Ham messages (in **Blue** Colour)
- Reduce the dimension of Document_Term_Matrix data, using the findFrequentTerms function.
- Finally fit our model using the naiveBayes function in library e1071 and come up with the confusion matrix



	Actual	
Predicted	ham	spam
ham	1422	37
spam	7	202

- Here my efficiency of my model is 0.9736, and I am happy with it.

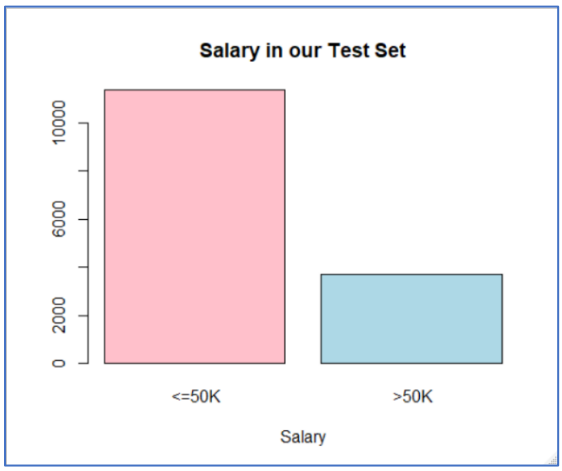
2 QUESTION NO 2 (CLASSIFICATION WITH THE SALARY DATA)

Lets see the structure of our data:

```
'data.frame':    30161 obs. of  14 variables:
 $ age          : int  39 50 38 53 28 37 49 52 31 42 ...
 $ workclass    : Factor w/ 7 levels " Federal-gov",..: 6 5 3 3 3 3 5 3 3 ...
 $ education    : Factor w/ 16 levels " 10th"," 11th",..: 10 10 12 2 10 13 7 12 13 10 ...
 $ educationno  : int  13 13 9 7 13 14 5 9 14 13 ...
 $ maritalstatus: Factor w/ 7 levels " Divorced"," Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
 $ occupation   : Factor w/ 14 levels " Adm-clerical",..: 1 4 6 6 10 4 8 4 10 4 ...
 $ relationship : Factor w/ 6 levels " Husband"," Not-in-family",..: 2 1 2 1 6 6 2 1 2 1 ...
 $ race         : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5 5 ...
 $ sex          : Factor w/ 2 levels " Female"," Male": 2 2 2 2 1 1 1 2 1 2 ...
 $ capitalgain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
 $ capitalloss  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ hoursperweek : int  40 13 40 40 40 40 16 45 50 40 ...
 $ native       : Factor w/ 40 levels " Cambodia"," Canada",..: 38 38 38 38 5 38 22 38 38 38 ...
 $ Salary       : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 2 2 2 ...
```

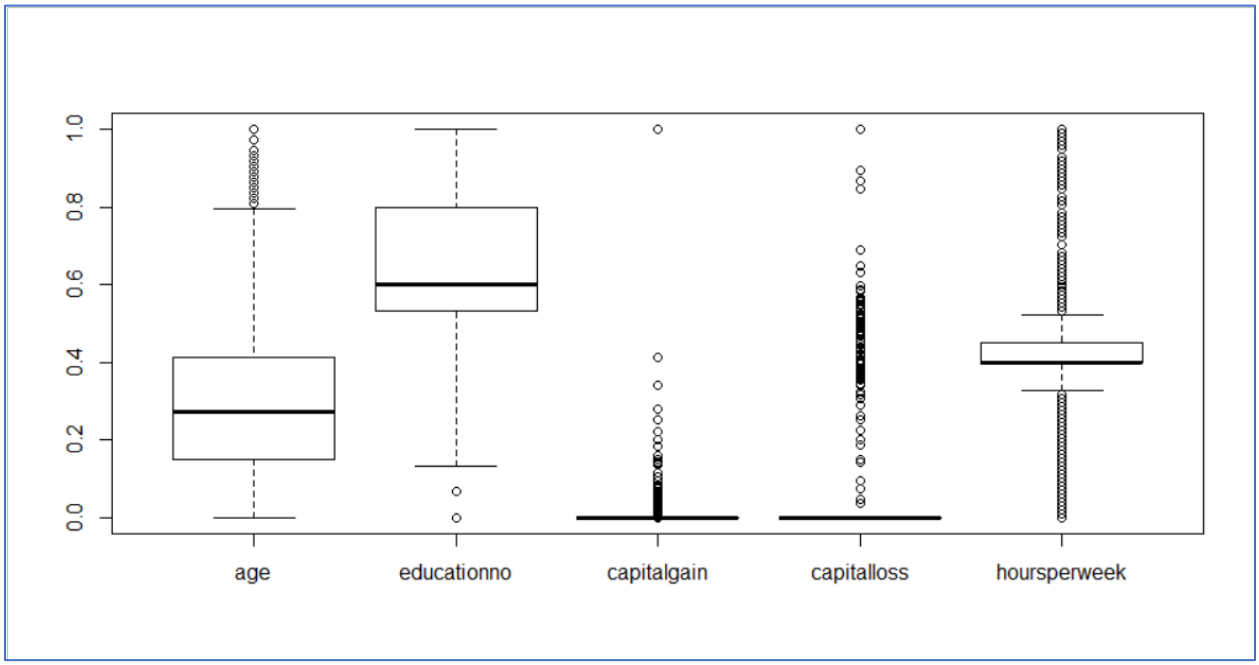
Here my data contains 9 factor columns and 5 numeric variables. So, here I am going to pass my function normalized dummy to normalize the whole data as well as create dummy variables for all the factor data.

Let's have a look on our Salary (categorical) variable:



Although it's imbalanced. I may balance the data if I find something specious in my results.

2.1 BOXPLOT OF NUMERICAL VARIABLES IN TEST DATA SET AFTER THE NORMALIZATION :



Here we can see lots of outlier in my data, so in such scanerio I may not consider to remove them as I may face loss of lots of informations.

So I consider to move for my model fitting with the normalised dummy data.

2.2 MODEL 1 WITHOUT LAPLACE SMOOTHING:

Summary of my model is

	Length	Class	Mode
apriori	2	table	numeric
tables	102	-none-	list
levels	2	-none-	character
isnumeric	102	-none-	logical
call	3	-none-	call

Here I got my efficiency as 0.78373.
With the confusion matrix as given below

	Predicted	
Actual	<=50K	>50K
<=50K	10753	607
>50K	2650	1050

With Laplace smoothing also I come up with the same result
So here my conclusion is with my efficiency as 0.78373