

PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering Degree in

Computer Science and Engineering

By

UPPALA MANIKANTA SANTOSH KOUSHIK (Reg. No. 3511597)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119, TAMILNADU

JUNE 2018

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

(Established under Section 3 of UGC Act, 1956)

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119

www.sathyabamauniversity.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **UPPALA MANIKANTA SANTOSH KOUSHIK** (Reg.No.3511597) who underwent the professional training in “Spam Filtering with R” under our supervision from May 2018 to June 2018.

Internal Guide

Mrs.S.P.GODLIN JASIL, M.E.,(Ph.D)

Head of the Department

Dr. S. MURUGAN, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, Uppala Manikanta Santosh Koushik (3511597) hereby declare that the Professional Training Report on “**Spam Filtering with R** ” done by me under the guidance of **Mrs.S.P.Godlin Jasil M.E.,(Ph.D.)** at Sathyabama institute of science and technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE: 14/06/2018

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **Sathyabama** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. M. Lakshmi, M.E., Ph.D.**, Dean, School of Computing and **Dr. S. Murugan M.E., Ph.D.**, Head of the Department, Department of Computer Science and Engineering for providing necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs.S.P.GODLIN JASIL, M.E.,(Ph.D)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Computer Science and Engineering** who were helpful in many ways for the completion of the project.

Also, I thank the **Almighty** and my **Parents** for supporting me in the completion of the professional Training.

TRAINING CERTIFICATE



Roll No:NPTEL18CS28S3270094

To
SATHYABAMA UNIVERSITY
CHENNAI

613/128



No. of credits recommended by NPTEL:2

Score	Type of Certificate
>=90	Elite + Gold Medal
60-89	Elite
40-59	Successfully Completed the course
<40	No Certificate



Elite

NPTEL Online Certification

(Funded by the Ministry of HRD, Govt. of India)



This certificate is awarded to

UPPALA MANIKANTA SANTOSH KOUSHIK

for successfully completing the course

Data Science for Engineers

with a consolidated score of **66 %**

Online Assignments	17.5/25	Proctored Exam	48.75/75
--------------------	---------	----------------	----------

A. Ramesh

Prof. A. Ramesh
Chairman
Center for Continuing Education, IITM

Total number of candidates certified in this course: **769**

Feb-Mar 2018
(8 week course)

Prof. Andrew Thangaraj

Prof. Andrew Thangaraj
NPTEL Coordinator
IIT Madras



Indian Institute of Technology Madras



Roll No: NPTEL18CS28S3270094

To validate and check scores: <http://nptel.ac.in/noc>

ABSTRACT

Spam messages and spam emails are increasingly disturbing and are becoming potential threats to one's data security. Phishing, is a best example for spam messages in which the user is deceived and later looted by the intruder or the attacker. The primary purpose and objective of "Spam filter with R" is to analyse SMS and classify them under 'Spam / Ham' . i.e ; Ham is something opposite to spam which means, it is normal message and not a potential threat to the data security. Machine learning technique such as 'Naive Bayes' is employed in order to classify and later predict results based on the analysis done on the data by using the confusion matrix and other techniques. Also, R programming is a vital programming language specifically is a statistical programming language and also with the recent changes that have been made and the installation of latest packages no longer makes it a less useful software when compared to python. The main idea behind the project is to refine and detect spam SMS received by the user and predict later on by the using the former principles.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	vi
1	INTRODUCTION	1
	1.1 OUTLINE OF THE PROJECT	2
	1.2 LITERATURE REVIEW	2
2	AIM AND SCOPE OF PROJECT	3
	2.1 PROBLEM STATEMENT	3
	2.2 OBJECTIVES	3
3	ALGORITHMS AND METHODS	4
	3.1 GENERAL	5
	3.2 OVERVIEW	7
	3.3 OVERVIEW OF PLATFORM	8
	3.3.1PROGRAMMINGINTERFACE	8
4	RESULTS AND DISCUSSION	9
5	CONCLUSION	15
	5.1 FUTURE WORKS	15
6	REFERENCES	16
	APPENDICES	17
	SOURCE CODE IN R	17

TABLE OF FIGURES

Figure No.	TITLE	Page no.
1.1	Former Learning model	2
3.1	Application of Naïve Bayes	5
3.2	Dataset Overview	7
3.3	Rstudio Interface	8
4.1	RStudio application	9
4.2	Importing and Analysing	10
4.3	Data Cleansing	10
4.4	Removing irrelevant info	11
4.5	Structure after cleansing	11
4.6	Generating a WordCloud	12
4.7	Data partition process	13
4.8	Accuracy and Confusion Matrix	14

CHAPTER 1

INTRODUCTION

R is a programming language and free software environment for statistical computing and graphics that is supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. Polls, surveys of data miners, and studies of scholarly literature databases show that R's popularity has increased substantially in recent years. As of June 2018, R ranks 8th in the TIOBE index, a measure of popularity of programming languages.

R is a GNU package. The source code for the R software environment is written primarily in C, Fortran, and R. R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems. While R has a command line interface, there are several graphical front-ends, most notably RStudio and RStudio Server, which are the only GUIs developed by the R Foundation. and integrated development environments available.

Electronic spamming is the use of electronic messaging systems to send an unsolicited message (spam), especially advertising, as well as sending messages repeatedly on the same site. While the most widely recognized form of spam is email spam, the term is applied to similar abuses in other media: instant messaging spam, Usenet newsgroup spam, Web search engine spam, spam in blogs, wiki spam, online classified ads spam, mobile phone messaging spam, Internet forum spam, junk fax transmissions, social spam, spam mobile apps, television advertising and file sharing spam.

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time,

1.1 OUTLINE OF THE PROJECT

The main goal of this project is to analyse the messages received by the user and then classifying them as spam or ham based on the naïve Bayesian machine learning technique . This is all done in RStudio or can also be done on Anaconda Navigator thereby installing RStudio on it again . Finally, the basic elements are -

1. Data Engineering and Analysis.
2. Data Cleansing.
3. Model learning and Prediction.
4. Confusion matrix Analysis.
5. Prediction parameters overview.

1.2 LITERATURE REVIEW

There has been a model recently published in github by Marshall.E , that claims to predict the outcomes whether the message are spam or ham using a lesser and lower classifier than naïve baye's and thus obtained results. However it's claims do not fulfil the accuracy level and the precision the model has to possess in order to be called a worthy model. But, this present model's accuracy and precision which is obtained is far more better than it, in classification and prediction respectively which makes this model better and efficient than the former one and many other projects and papers which have the same problem are present.

```
Confusion Matrix and Statistics

      Reference
Prediction ham spam
ham      130     8
spam       1    10

      Accuracy : 0.9396
      95% CI   : (0.8884, 0.972)
No Information Rate : 0.8792
P-Value [Acc > NIR] : 0.01111
```

fig:1.1: Former Learning model

CHAPTER 2

AIM AND SCOPE OF THE PROJECT

2.1 PROBLEM STATEMENT

The input data is a set of SMS messages that has been classified as either a ham or a spam message respectively in a 'csv' file. The goal of the project is to build a model to identify messages as either ham or spam and thereby predict later on by analysing the training and test data sets respectively.

2.2 OBJECTIVES

To prevent the exploitation of vulnerabilities present in a system which are often targeted by the spam message senders and that can lead to identity theft, data loss and many other ill effects by employing this spam filter.

To successfully classify the given messages into spam/ham and thereby classifying these messages as precisely as possible with a good accuracy and other parameters that influence the prediction.

To use RStudio , a tool which is used to write and execute programs written in R language to plot the required figures and insights provided by the data and also analyse using the tool's various built in packages or by installing them externally if possible.

To employ Naïve Bayes classification technique, a supervised machine learning technique which is very useful in classification especially linear classification in order to classify the messages and hence later on, predict the test set . Then, the model's confusion matrix will finally reveal the present model's performance along with the prediction parameters respectively.

CHAPTER 3

ALGORITHMS AND METHODS

The first step towards an understanding of why the study and knowledge of machine learning is so important is to define exactly what is meant by machine learning. According to a recent study, machine learning algorithms are expected to replace 25% of the jobs across the world, in the next 10 years. With the rapid growth of big data and availability of programming tools like Python and R –machine learning is gaining mainstream presence for data scientists. Machine learning applications are highly automated and self-modifying which continue to improve over time with minimal human intervention as they learn with more data. For instance, Netflix's recommendation algorithm learns more about the likes and dislikes of a viewer based on the shows every viewer watches. To address the complex nature of various real world data problems, specialized machine learning algorithms have been developed that solve these problems perfectly.

The basic techniques under machine learning are:

1. **Supervised Algorithms:** Machine learning algorithms that make predictions on given set of samples. Supervised machine learning algorithm searches for patterns within the value labels assigned to data points.
2. **Unsupervised Algorithms:** There are no labels associated with data points. These machine learning algorithms organize the data into a group of clusters to describe its structure and make complex data look simple and organized for analysis.
3. **Reinforcement Algorithms:** These algorithms choose an action, based on each data point and later learn how good the decision was. Over time, the algorithm changes its strategy to learn better and achieve the best reward.

Generally the first methods i.e. Supervised and Unsupervised algorithms are very useful and are pretty much used in day to day tasks and common activities lead from similar sources respectively.

3.1 GENERAL

It would be difficult and practically impossible to classify a web page, a document, an email or any other lengthy text notes manually. This is where Naïve Bayes Classifier machine learning algorithm comes to the rescue. A classifier is a function that allocates a population's element value from one of the available categories. For instance, Spam Filtering is a popular application of Naïve Bayes algorithm. Spam filter here, is a classifier that assigns a label “Spam” or “Not Spam” to all the emails.

Naïve Bayes Classifier is amongst the most popular learning method grouped by similarities ,that works on the popular Bayes Theorem of Probability- to build machine learning models particularly for disease prediction and document classification. It is a simple classification of words based on Bayes Probability Theorem for subjective analysis of content. Applications of Bayes theorm are mainly:



fig:3.1: Applications of Naïve Bayes

1. Sentiment Analysis : It is used primarily for Facebook to analyse status updates expressing positive or negative emotions.
2. Document Categorization : Google uses document classification to index documents and find relevancy scores i.e. the PageRank. PageRank mechanism considers the pages marked as important in the databases that were parsed and classified using a document classification technique.

3. Naïve Bayes Algorithm is also used for classifying news articles about Technology, Entertainment, Sports, Politics, etc.
4. Email Spam Filtering-Google Mail uses Naïve Bayes algorithm to classify your emails as Spam or Not Spam

Advantages of the Naïve Bayes Classifier Machine Learning Algorithm:

1. Naïve Bayes Classifier algorithm performs well when the input variables are categorical.
2. A Naïve Bayes classifier converges faster, requiring relatively little training data than other discriminative models like logistic regression, when the Naïve Bayes conditional independence assumption holds.
3. With Naïve Bayes Classifier algorithm, it is easier to predict class of the test data set. A good bet for multi class predictions as well.
4. Though it requires conditional independence assumption, Naïve Bayes Classifier has presented good performance in various application domains.

Data Science Libraries in Python to implement Naïve Bayes – Sci-Kit Learn and in our context being R the package used is ‘e1071’.

Generally the base for this theorem is Bayes theorem which states that:

$$\text{Prior} = \frac{\text{posterior} * \text{likelihood}}{\text{Evidence}}$$

Also, we can give this in terms of $P(x)$, $P(C_k/x)$ respectively too:

$$P(C_k / x) = \frac{P(C_k) P(x / C_k)}{P(x)}$$

Abstractly, naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector $x = (x_1, \dots, x_n)$ representing some n features (independent variables), it assigns to this instance probabilities respectively.

3.2 OVERVIEW

The dataset consists of approximately 510 messages which are already classified into 'ham and spam' respectively. So, we have to take the data and import it after which we have to divide it into training set and testing set, if necessary we can also include validation set also. Let's have a basic look at the data in the below figure, if we observe the ham messages, they are casual and not really a threat. But if you take the spam messages they are just fraudulent and ill solicited messages from unknown sources to grab your important information respectively from your mobile or any other personal device only.

1	type	text
2	ham	Hope you are having a good week. Just checking in
3	ham	K..give back my thanks.
4	ham	Am also doing in cbe only. But have to pay.
5	spam	complimentary 4 STAR Ibiza Holiday or £10,000 cash needs your URGENT collection. 09066364349 NOW from Landline not to lose out! Box434SK38WP150PPM18+
6	spam	okmail: Dear Dave this is your final notice to collect your 4* Tenerife Holiday or #5000 CASH award! Call 09061743806 from landline. TCs SAE Box326 CW25WX 150pp
7	ham	Aiya we discuss later lar... Pick u up at 4 is it?
8	ham	Are you this much buzy
9	ham	Please ask mummy to call father
10	spam	Marvel Mobile Play the official Ultimate Spider-man game (£4.50) on ur mobile right now. Text SPIDER to 83338 for the game & we ll send u a FREE 8Ball wallpaper
11	ham	fyi I'm at usf now, swing by the room whenever
12	ham	Sure thing big man. i have hockey elections at 6, shouldn't go on longer than an hour though
13	ham	I anything lor...
14	ham	By march ending, i should be ready. But will call you for sure. The problem is that my capital never complete. How far with you. How's work and the ladies
15	ham	Hmm well, night night
16	ham	K I'll be sure to get up before noon and see what's what
17	ham	Ha ha cool cool chikku chikku:-):-DB-)
18	ham	Darren was saying dat if u meeting da ge den we dun meet 4 dinner. Cos later u leave xy will feel awkward. Den u meet him 4 lunch lor.
19	ham	He dint tell anything. He is angry on me that why you told to abi.
20	ham	Up to u... u wan come then come lor... But i din c any stripes skirt...
21	spam	U can WIN £100 of Music Gift Vouchers every week starting NOW Txt the word DRAW to 87066 TsCs www.ldew.com SkillGame,1Winaweek, age16.150ppermessSubsc
22	ham	2mro i am not coming to gym machan. Goodnight.
23	ham	ARR birthday today:) i wish him to get more oscar.

fig:3.2: Dataset Overview

3.3 OVERVIEW OF PLATFORM

To design the learning model , we have to use RStudio which is a very statistical tools especially designed for R programming with pre installed packages that help in solving complex problems and also predicting by getting insights from the stucture of data. Data visualization is another bright feature of RStudio is to its ability to present bar graphs , plots, correlation maps, scatter plots and any other statistical plots quite beautifully than many other present day tools .

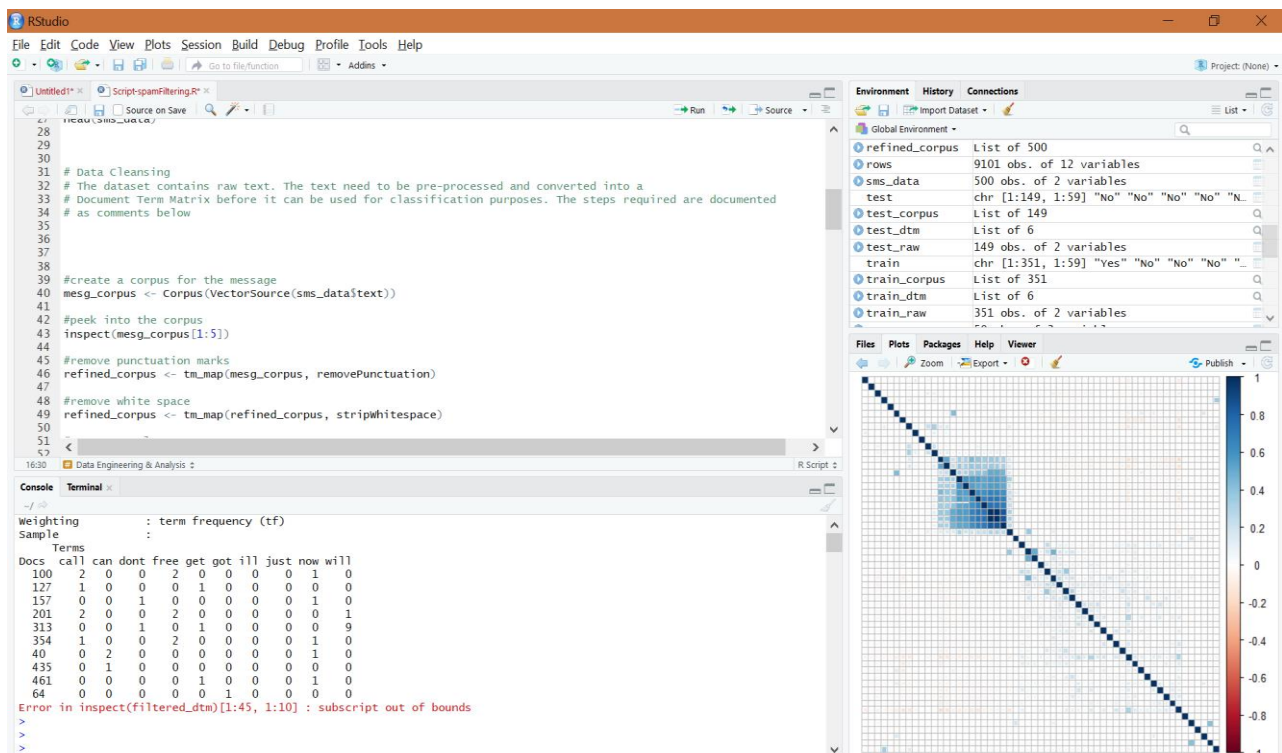


fig:3.3: RStudio Interface

3.3.1 Programming Interface:

The programming interface used here for the whole process is 'R' language which is again statistical software that was primarily built as an extension for S language and later on gained popularity as a language suitable for data analysis.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 IMPLEMENTATION AND PERFORMANCE:

This project highlights the results of the model built for classification of spam and ham messages respectively using R and naïve baye's technique. and the snapshots for each of the activities are shown along with the discussion of each activity describing it's working process. Each snapshot describes every single step of training the data set and related process involved.

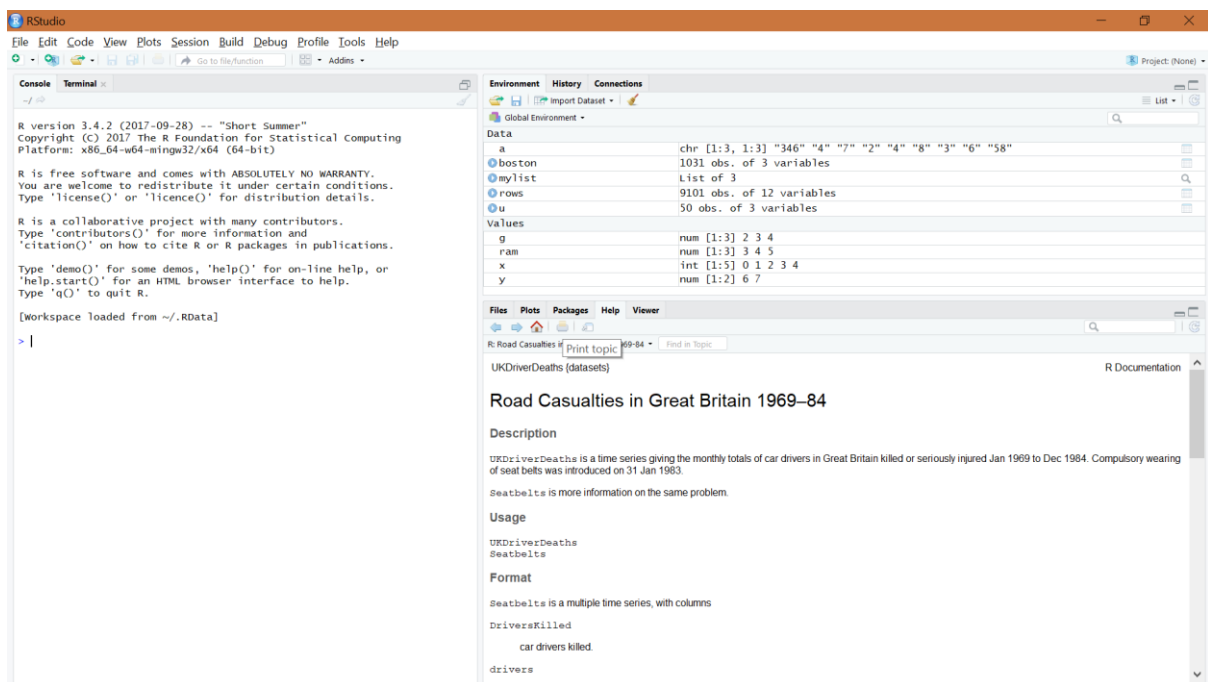


fig:4.1: RStudio Application Interface

The above figure 4.1 depicts the basic interface of the Rstudio on which we are going to further run our R code and create training sets, test sets, validation set if needed respectively. The basic requirements and interface tutorial and various tips are presented at the start of the application everytime.

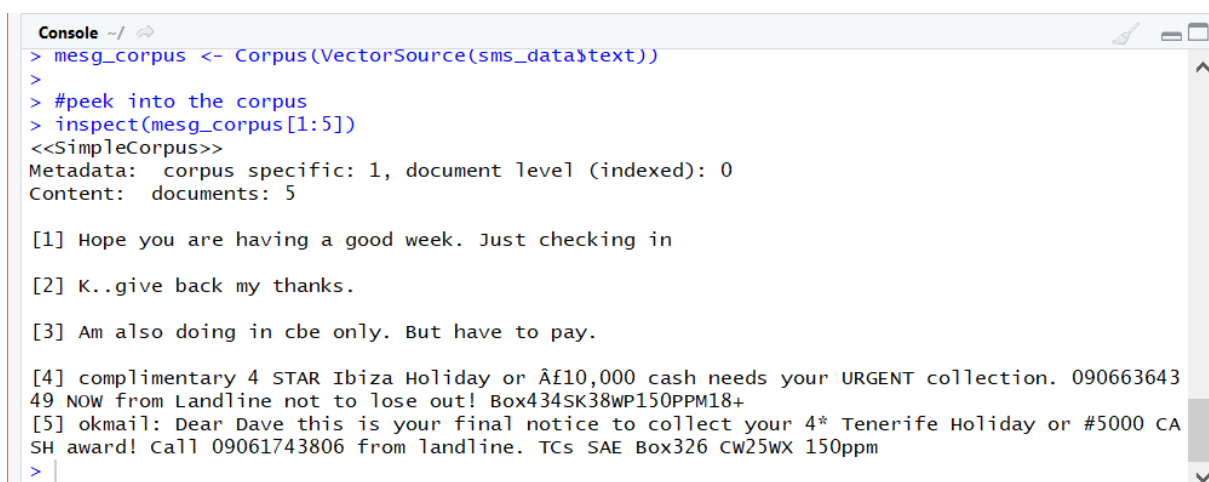
```

sms_data <- read.csv("Data/sms_spam_short.csv", stringsAsFactors=FALSE)
sms_data$type <- as.factor(sms_data$type)
str(sms_data)
summary(sms_data)
head(sms_data)

```

fig:4.2: Importing and Analyzing the dataset

The above figure 4.2 suggests the importing of data set from a csv file downloaded online and is saved in your respective repository in your system. The str function actually shows no of entries and the factor levels which is 2 : 'ham' and 'spam' respectively. The head function is used to display the first few observations.



```

Console ~/
> mesg_corpus <- Corpus(VectorSource(sms_data$text))
>
> #peek into the corpus
> inspect(mesg_corpus[1:5])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 5

[1] Hope you are having a good week. Just checking in
[2] K..give back my thanks.
[3] Am also doing in cbe only. But have to pay.
[4] complimentary 4 STAR Ibiza Holiday or £10,000 cash needs your URGENT collection. 090663643
49 NOW from Landline not to lose out! Box434SK38WP150PPM18+
[5] okmail: Dear Dave this is your final notice to collect your 4* Tenerife Holiday or #5000 CA
SH award! Call 09061743806 from landline. TCS SAE Box326 CW25WX 150ppm
>

```

fig:4.3:Data cleansing and inspecting

The above figure 4.3 depicts the image of the console when the corpus i.e the collection of docs or messages is inspected after being created using the "corpus" function from scratch which is available in the 'tm' library. The inspect function is used to display specified number of statements in which we have given first 5 only and are displayed above and all of this is before cleaning.

```

#remove punctuation marks
refined_corpus <- tm_map(mesg_corpus, removePunctuation)

#remove white space
refined_corpus <- tm_map(refined_corpus, stripwhitespace)

#convert to lower case
refined_corpus <- tm_map(refined_corpus, content_transformer(tolower))

#remove numbers in text
refined_corpus <- tm_map(refined_corpus, removeNumbers)|

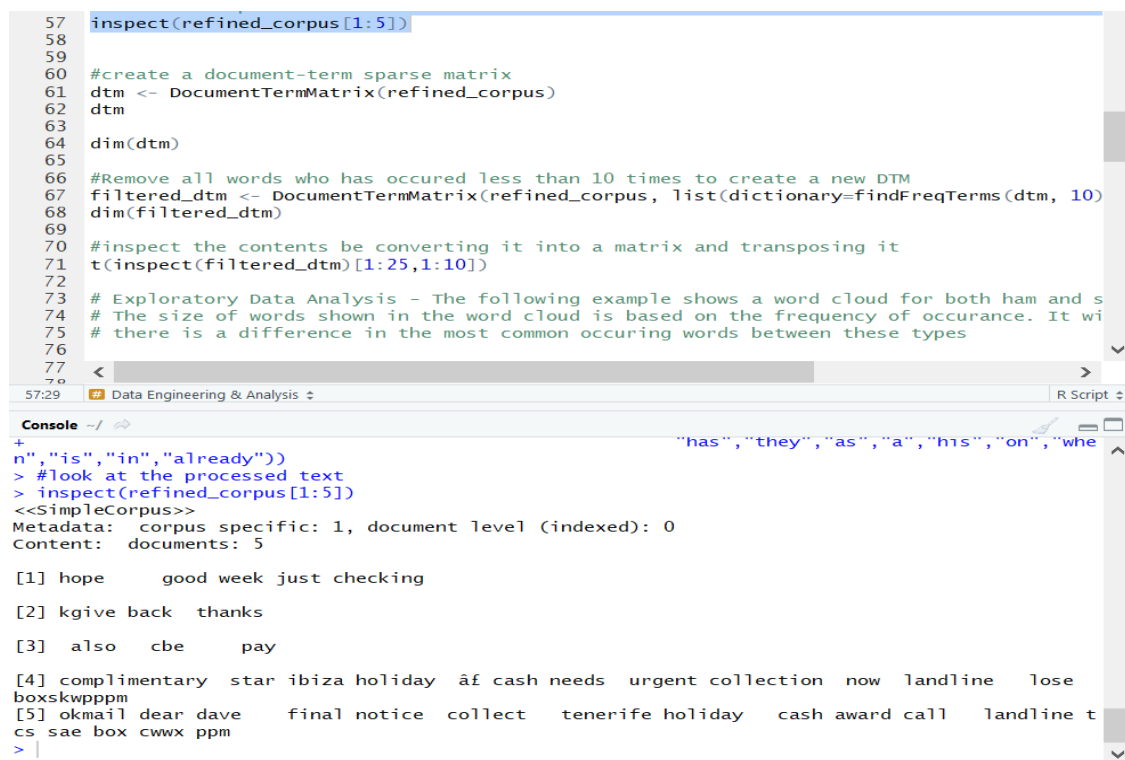
#remove stop words
refined_corpus <- tm_map(refined_corpus, removeWords, stopwords())

#remove specific words
refined_corpus <- tm_map(refined_corpus, removeWords, c("else","the","are","for",
"has","they","as","a","his","on",

```

fig:4.4:Removing irrelevant content from the message.

The above figure 4.4 depicts the steps and code involved in changing the messages in the raw or existing form to a form which is useful for natural language processing respectively. Remove stop words, punctuation marks, white spaces, other words and numbers using the 'tm_map' function respectively available in the 'tm' package.



```

57 inspect(refined_corpus[1:5])
58
59
60 #create a document-term sparse matrix
61 dtm <- DocumentTermMatrix(refined_corpus)
62 dtm
63
64 dim(dtm)
65
66 #Remove all words who has occurred less than 10 times to create a new DTM
67 filtered_dtm <- DocumentTermMatrix(refined_corpus, list(dictionary=findFreqTerms(dtm, 10)
68 dim(filtered_dtm)
69
70 #inspect the contents by converting it into a matrix and transposing it
71 t(inspect(filtered_dtm)[1:25,1:10])
72
73 # Exploratory Data Analysis - The following example shows a word cloud for both ham and s
74 # The size of words shown in the word cloud is based on the frequency of occurrence. It wi
75 # there is a difference in the most common occurring words between these types
76
77
78
79
80

```

57:29 Data Engineering & Analysis R Script

```

+
n","is","in","already"))
> #look at the processed text
> inspect(refined_corpus[1:5])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 5

[1] hope      good week just checking

[2] kgive back  thanks

[3] also      cbe      pay

[4] complimentary star ibiza holiday af cash needs urgent collection now landline lose
boxskwpppm

[5] okmail dear dave final notice collect tenerife holiday cash award call landline t
cs sae box cwxw ppm
> |

```

fig:4.5: Structure after Data cleansing process

The above figure 4.5 depicts the image of the console after executing the inspect command after the data cleansing process that is the removal of the various irrelevant words. Also, this also includes the creation of Document Sparse matrix which can be primarily used for classification respectively. 'dim' function gives the dimensions of the entity that is its rows and observations respectively. As we can see, all the irrelevant punctuations, numbers are removed successfully.

```
library(wordcloud)

pal <- brewer.pal(9,"Dark2")

wordcloud(refined_corpus[sms_data$type=="ham"], min.freq=5,
random.order=FALSE, colors=pal)
```

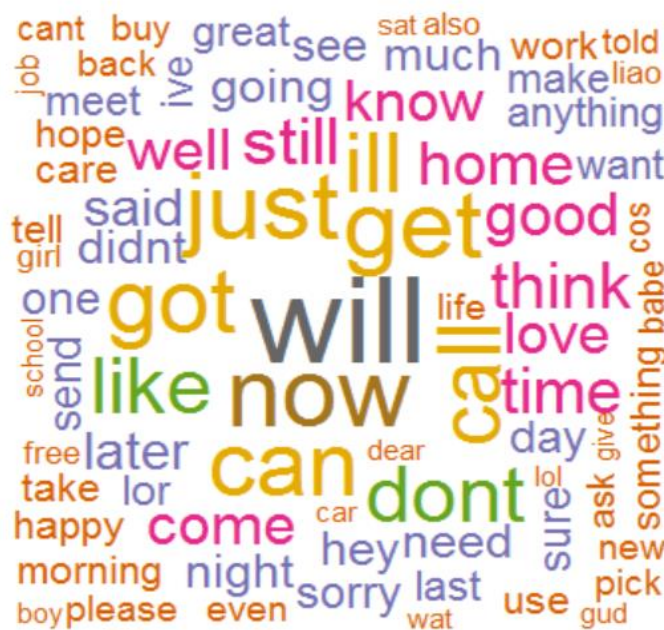


fig:4.6: Generating a wordcloud

The above figure 4.6 depicts a word cloud which is basically highlighting the more frequently analysed words respectively and the largest one is the one which was used most number of time. This function is available in 'wordcloud' package respectively and can be imported if not installed.

```

##### Modeling & Prediction #####
library(caret)

inTrain <- createDataPartition(y=sms_data$type ,p=0.7,list=FALSE)

#Splitting the raw data

train_raw <- sms_data[inTrain,]
test_raw <- sms_data[-inTrain,]

#splitting the corpus
train_corpus <- refined_corpus[inTrain]
test_corpus <- refined_corpus[-inTrain]

#splitting the dtm
train_dtm <- filtered_dtm[inTrain,]
test_dtm <- filtered_dtm[-inTrain,]

# Instead of using the counts of words within document, we will replace them with indicat
# Yes indicates if the word occurred in the document and No indicate it does not. This pro
# Numeric data into factor data
conv_counts <- function(x) {
  x <- ifelse(x > 0, 1, 0)
  x <- factor(x, levels = c(0, 1), labels = c("No", "Yes"))
}
train <- apply(train_dtm, MARGIN = 2, conv_counts)
test <- apply(test_dtm, MARGIN = 2, conv_counts)

#convert to a data frame and add the target variable
df_train <- as.data.frame(train)
df_test <- as.data.frame(test)
df_train$type <- train_raw$type
df_test$type <- test_raw$type
df_train[1:10,1:10]

#Model Building Build model based on the training data
library(e1071)

#Leave out the last column (target)
modFit <- naiveBayes(df_train[,-60], df_train$type)
modFit

```

fig:4.7: The Data partition: Training and Testing process

The above picture depicts the final process involved in our model which is learning and predicting thereby being able to predict with a good accuracy. Data is partitioned into training set and test set. Mostly 70% for training set and the remaining for the test set respectively. After that classification process begins and 'yes' appears if the word has occurred in the document and vice versa if not occurred in the document. The library 'e1071' has the naïve bayes classifiers and has to be imported and then we implement the naïve bayes classification algorithm over each word and find the conditional probability of it being 'ham' and 'spam' respectively.


```

> ##### Testing #####
##
> # Now let us predict the class for each sample in the test data. Then compare the prediction
with the actual
> # value of the class..
>
> predictions <- predict(modFit, df_test)
> confusionMatrix(predictions, df_test$type)
Confusion Matrix and Statistics

          Reference
Prediction ham spam
   ham   128     5
   spam    3    13

      Accuracy : 0.9463
      95% CI   : (0.8969, 0.9765)
 No Information Rate : 0.8792
 P-Value [Acc > NIR] : 0.004759

      Kappa : 0.7345
McNemar's Test P-Value : 0.723674

      Sensitivity : 0.9771
      Specificity : 0.7222
   Pos Pred Value : 0.9624
   Neg Pred Value : 0.8125
    Prevalence : 0.8792
   Detection Rate : 0.8591
 Detection Prevalence : 0.8926
  Balanced Accuracy : 0.8497

      'Positive' Class : ham

```

fig:4.8: The Data Prediction: Accuracy and Confusion matrix

The above figure 4.8 depicts the prediction process took place in the code for spam filtering respectively. The 'predict' function serves primarily for this purpose alone. Accuracy rate is the general parameter taken into consideration to declare a project worthy or unworthy. The confusion matrix is made up of True negatives, True positives, Actual positives and Negatives respectively. We got an accuracy of 94.5 % which is arguably very good accuracy rate. If you see the confusion matrix, one can understand from its structure that, ham was misclassified as spam only 3 times where as spam was misclassified as ham 5 times which is really not a very big issue to be concerned with respectively.

CHAPTER 5

CONCLUSION

This project will increase a person's interest towards data analysis and the power of data which could be used further in every field other than computer science like business, agriculture, educational sector, medicine sector(pharmacy) and many more sectors. By using machine learning to predict the outcomes in the future, one may sometimes or majority of times fail due to less organizational intent and more pressure. Data scientists need a pressure free environment in order to work out well and also , more importantly give their best and also will increase a person's enthusiasm towards design and analysis of new machine learning algorithms. Not only, these kinds of projects will do the above mentioned things, but also will be a stepping stone for many other budding young programmers and aspiring data science enthusiasts all over to do many more activities.

As a computer scientist, it is important to understand all of these types of M.L algorithms so that one can use them properly. Furthermore, there is a need to understand the details of each technique involved so that it will be possible to predict if there are special cases in which the software won't work quickly, or if it will produce unacceptable results like other errors.

5.1 FUTURE WORKS

Of course, there are often times when this project may run across a problem that has not been previously studied. In these cases, one has to come up with a new algorithm, or apply an old technique in a new way. The more one knows about M.L algorithms in this case, the better, are the chances of finding a good way to solve the problem. In many cases, a new problem can be reduced to an old problem without too much effort, but one will need to have a fundamental understanding of the old problem in order to do this. This has to be done in order to get a higher accuracy than the previous model made.

CHAPTER 6

REFERENCES

1. <http://www.codecademy.co.in>
2. <http://www.datacamp.org/Analysis/MachineLearning>
3. http://www.heyitskoushik.wordpress.com/Data_Science_Revolution
4. <http://www.programmingsimplified.com/R/>
5. <http://www.stackoverflow.com>
6. http://www.wikipedia.co.in/Machine_Learning
7. <http://www.dataquest.com>
8. <http://github.com>

APPENDICES

CODING IN R:

```
# Problem Statement
# The input data is a set of SMS messages that has been classified as either ham or
spam. The goal of the
# exercise is to build a model to identify messages as either ham or spam.
```

```
sms_data <- read.csv("sms_spam_short.csv", stringsAsFactors=FALSE)
```

```
sms_data$type <- as.factor(sms_data$type)
```

```
str(sms_data)
```

```
summary(sms_data)
```

```
head(sms_data)
```

```
# Data Cleansing
```

```
# The dataset contains raw text. The text need to be pre-processed and converted
into a
```

```
# Document Term Matrix before it can be used for classification purposes. The steps
required are documented
```

```
# as comments below
```

```
install.packages("tm")
```

```
library(tm)
```

```
#create a corpus for the message
```

```
mesg_corpus <- Corpus(VectorSource(sms_data$text))
```

```
#peek into the corpus
```

```
inspect(mesg_corpus[1:5])
```

```
#remove punctuation marks
```

```
refined_corpus <- tm_map(mesg_corpus, removePunctuation)
```

```
#remove white space
```

```
refined_corpus <- tm_map(refined_corpus, stripWhitespace)
```

```

#convert to lower case
refined_corpus <- tm_map(refined_corpus, content_transformer(tolower))

#remove numbers in text
refined_corpus <- tm_map(refined_corpus, removeNumbers)

#remove stop words
refined_corpus <- tm_map(refined_corpus, removeWords, stopwords())

#remove specific words
refined_corpus <- tm_map(refined_corpus, removeWords, c("else","the","are","for",
"has","they","as","a","his","on","when","is","in","already"))
#look at the processed text
inspect(refined_corpus[1:5])

#create a document-term sparse matrix
dtm <- DocumentTermMatrix(refined_corpus)
dtm

dim(dtm)

#Remove all words who has occurred less than 10 times to create a new DTM
filtered_dtm <- DocumentTermMatrix(refined_corpus,
list(dictionary=findFreqTerms(dtm, 10)))
dim(filtered_dtm)

#inspect the contents by converting it into a matrix and transposing it
t(inspect(filtered_dtm)[1:25,1:10])

# Exploratory Data Analysis - The following example shows a word cloud for both
ham and spam message.
# The size of words shown in the word cloud is based on the frequency of
occurrence. It will clearly show that
# there is a difference in the most common occurring words between these types

install.packages("wordcloud")
library(wordcloud)

pal <- brewer.pal(9,"Dark2")

wordcloud(refined_corpus[sms_data$type=="ham"], min.freq=5,
          random.order=FALSE, colors=pal)

```

```
wordcloud(refined_corpus[sms_data$type=="spam"], min.freq=2,  
          random.order=FALSE, colors=pal)
```

```
library(caret)
```

```
inTrain <- createDataPartition(y=sms_data$type ,p=0.7,list=FALSE)
```

```
#Splitting the raw data
```

```
train_raw <- sms_data[inTrain,]  
test_raw <- sms_data[-inTrain,]
```

```
#splitting the corpus
```

```
train_corpus <- refined_corpus[inTrain]  
test_corpus <- refined_corpus[-inTrain]
```

```
#splitting the dtm
```

```
train_dtm <- filtered_dtm[inTrain,]  
test_dtm <- filtered_dtm[-inTrain,]
```

```
# Instead of using the counts of words within document, we will replace them with  
indicators "Yes" or "No".
```

```
# Yes indicates if the word occurred in the document and No indicate it does not. This  
procedure converts
```

```
# Numeric data into factor data
```

```
conv_counts <- function(x) {  
  x <- ifelse(x > 0, 1, 0)  
  x <- factor(x, levels = c(0, 1), labels = c("No", "Yes"))  
}
```

```
train <- apply(train_dtm, MARGIN = 2, conv_counts)
```

```
test <- apply(test_dtm, MARGIN = 2, conv_counts)
```

```
#convert to a data frame and add the target variable
```

```
df_train <- as.data.frame(train)
```

```
df_test <- as.data.frame(test)
```

```
df_train$type <- train_raw$type
```

```
df_test$type <- test_raw$type
```

```
df_train[1:10,1:10]
```

```
#Model Building Build model based on the training data  
library(e1071)
```

```
#Leave out the last column (target)  
modFit <- naiveBayes(df_train[,-60], df_train$type)  
modFit
```

```
# Now let us predict the class for each sample in the test data. Then compare the  
prediction with the actual  
# value of the class..
```

```
predictions <- predict(modFit, df_test)  
confusionMatrix(predictions, df_test$type)
```