

SPAM FILTRATION PROJECT USING NLP

Project report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Ankita Chauhan

161254

Under the supervision of

Yugal Kumar

to



Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Wahnaghat,
Solan-173234, Himachal Pradesh

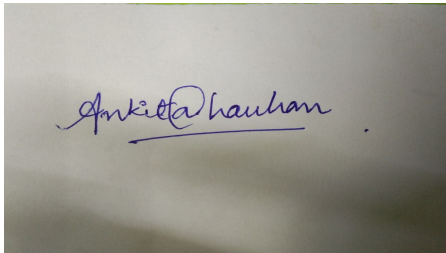
Certificate

Candidate's Declaration

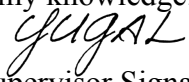
I hereby declare that the work presented in this report entitled “SPAM FITERATION USING NLP.....” in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from Feb 2020 to June 2020 under the supervision of Dr. Yugal Kumar professor of Computer Science Department.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Ankita Chauhan (161254).

A photograph of a handwritten signature in blue ink that reads "Ankita Chauhan". The signature is written on a light-colored surface and is underlined.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

 15/07/2020
(Supervisor Signature)

Dr .Yugal Kumar
Computer Science Department

Dated:

ACKNOWLEDGEMENT

It is my privilege to thank all the people who contributed to make this internship period a great and unforgettable experience of my life.

First, I would like to thank Jaypee University Of Information Technology , Solan for providing me the golden opportunity of gaining practical industry knowledge through internship training at Infosys Ltd and I made a

I take this opportunity to express my sincere gratitude and deep regards to my mentor,

Mr. Yugal Kumar for his exemplary guidance, monitoring and constant encouragement in completing Training Program within a given stipulated time..

This internship program provided me many opportunities to work in different technologies and exposed me to interaction with different sets of people. I was able to utilize and enhance different skills in me like decision-making, teamwork and coordination.

TABLE OF CONTENT

sno	topic	Page no
1	Problem Statement	22
2	Objective	22
3	Methodology	23
4	Literature Survey	25
5	System Development	37
6	Performance Analysis	41
7	Plagiarism Report	50

LIST OF ABBREVIATIONS

Abbreviation	Full Form
NLP	Natural Language Processing
NLTK	Natural Language Tool Kit
Numpy	Numerical Python
Pandas	Python Data Analysis Library
$P(x)$	Probability of x
TIF	Term Inverses Frequency
IDF	Inverse Document Frequency

LIST OF FIGURES

- Image 1: Step 1 showed with output
- Image 2: Figure of step 2 with output
- Image 3: Figure of step 3 with output
- Image 4: Test and train the dataset
- Image 5: Figure shows the word cloud of spam category
- Image 6: Figure of word cloud of ham category
- Image 7: the fig shows the processing of data
- Image 8: Count of spam and ham in dataset
- Image 9: Applying classification on dataset
- Image 10 : Finding out precision , accuracy ,f - score of the dataset
- Image 11: Finally checking the trained test by testing it with input

LIST OF TABLES

Table 1 : Normalization table

Table 2 - Example of noise removal

Table 3 :Stemming of words

CHAPTER 1 - INTRODUCTION

1.1 Introduction

In this world of full of data , getting e - mails and messages every second the problem arises as the time to check our mails and message inbox is limited and the messages we get can be useful or completely fraud . This problem is arising in the whole world with all of us . So its time we come up with solutions that rectify this problem and help people differentiate between what is spam and what is really important . But this needs to be done with utmost care and intelligent algorithms .

Technology can be useful here like in this problem use of Machine Learning and NLP that is Natural Language Processing can help is to analyze the data and apply algorithms and concepts by training and testing the data . So that we can finally classify as spam and not spam mails and text.

Companies like Google has also used this technology to work on text like mails and automatically classify .

1.2 Problem Statement

Receiving hundreds of mails in a day and don't have time to read all ,but staying updated with the important messages and mails is extremely important . So lets solve the problem of spam mails so that you don't waste time and life goes much smoother .

In the project we are solving the problem of classifying spam and not spam mails in our inboxes.

1.3 Objective

The objective of our project is to study and implement how can we solve the problem of spam classification . The project involves the study of Natural Language Processing (NLP) ,Natural Language Tool Kit (NLTK) ,different algorithms and process.

Finding out which algorithm is suitable and can be used for universal process so that it yields results which are accurate and successfully helps people to find spam mail or genuine mails or messages .

1.4 Methodology

The project SPAM FILTRATION USING NPL is divided stepwise into parts :

1. Study of Natural Language Processing (NLP) and NLTK
2. Studying algorithms based on NLP like porter stemmer , Snowball stemmer
3. Finding out the right algorithm
4. Finding out relevant data set
5. Cleaning data set
6. Training the data
7. Applying algorithm
8. Testing the data
9. Finding out the precision and accuracy
10. Displaying the results

1.5 Organization

The project is stepwise organized into 5 Chapters . Each chapter very well explains the project and attention is paid to study every minute detail about the project.

CHAPTER 1 - Introduction:

The Introduction tells us the objective ,steps,problem statement and methodology of our project briefly.

CHAPTER 2 - Literature Survey

Literature survey includes the theoretical knowledge gathered by reading the research paper and studying the algorithms

CHAPTER 3 - System Development

System Development tells the system requirements, that is libraries , language used to code to perform the objective of our project.

CHAPTER 4 - Performance Analysis

Now we will analyze the performance of our algorithm testing it on data set we have chosen .

CHAPTER 5 -Conclusions

Finally we will conclude wether we could justify the problem statement coming with the solution of the problem.

References

Appendices

CHAPTER 2 - LITERATURE SURVEY

NLP (Natural Language Processing)

Natural Language Processing is the platform for human interaction of natural human languages and computing devices .NLP is one of the major aspect of computational linguistics, and also is useful for artificial intelligence and computer science.

It deals with :

- Signs
- Menus
- E mail
- Sms
- Web pages

NLTK (Natural :NLTK is one of the best platform for building Python programs to work with databases that is human language data. NLTK has very user friendly interfaces including lot of datasets called corpuses and a number of text processing libraries for classification, tagging, stemming,parsing, and

semantic reasoning, tokenization, wrappers for working on raw data to classify and find certain patterns .

NLTK has a very good documentation which very well explains the libraries

topics like computational linguistics and API documentation. It is suitable for linguists ,engineers ,students ,researchers educators and other people in industry.

NLTK is available for Mac OS X, Windows, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK is an amazing library to play with natural language. The latest version is NLTK 3.5.

Tokenization

Tokenization is one of the initial steps in the NLP process, it is one of the important step which splits longer strings of text into smaller pieces also called **tokens**. Larger chunks of text are broken also called tokenized into sentences further sentences can be broken also called tokenized into words, etc. After the text has been tokenized completely then only algorithm is applied to classify ,visualize , find patterns to do analysis.

Ex of sentence tokenization :

```
from nltk.tokenize import sent_tokenize
```

```
text = "Hey. Welcome to India."
```

```
sent_tokenize(text)
```

```
Output: ['Hey' , 'Welcome to India']
```

Ex of word tokenization:

```
from nltk.tokenize import word_tokenize text =  
"Hello everyone. Welcome to  
GeeksforGeeks."word_tokenize(text)
```

```
Output: ['Hello' , 'everyone' , '.', 'Welcome' ,  
'to' , 'GeeksforGeeks' , '.']
```

Ex of punkttokenizer:

```
from nltk.tokenize import PunktWordTokenizer  
tokenizer = PunktWordTokenizer()  
  
tokenizer.tokenize("Hey! Let's see how it's  
working.")
```

```
Output : ['Hey!', 'Let', "'s", 'see', 'how', 'it', "'s",  
'working', '.']
```

Normalization

Text analysis can be done accurately if only if text is normalized . Normalization refers to a series of related code to convert all the data we are dealing with in uppercase or lower case . This process also includes removing punctuation ,converting numbers to word format , expanding contractions .

Normalization allows treating all the words under same conditions and treating them uniformly.

Normalization is extremely useful for analyzing the social media comments as the people there usually use slangs ,short forms ,wrong spellings and abbreviation which needs to be correctly manipulated before analyzing .

Table 1 : Normalization table

Raw	Normalized
2morrow	tomorrow
2mrrw	tomorrow
Tomrw	tomorrow
b4	before
Otw	On the way
Ttyl	Talk to you later
:)	smile

Noise Removal

Noise removal is one of the most important text processing steps. It includes removing characters and digits . Also pieces of text that can interfere with your data analysis. This also is highly domain dependent.

For example in Tweets of Twitter application : Noise can produce inconsistent ,results as people include unnecessary punctuations . Only hashtags in a tweet signify some important concepts that charecterize a tweet.

Table 2 - Example of noise removal

Rawword	noise removal
..trouble..	trouble
trouble<	trouble
trouble!	trouble
<a>trouble	trouble
1.trouble	trouble

STEMMING

Stemming is the process of eliminating affixes like suffixes, prefixes, infixes, circumfixes from a word in order to obtain a word stem.

Stem which is also called root is the part of any word to which you add changing or deriving affixes such as -ed,-ize, -s,-de,mis . So stemming a word or sentence may result in words that are not actual words. The actual process of Stemming is created by removing the suffixes or prefixes used within a word to take out the root word

Table 3 :Stemming of words

words	Stemmed words
Running	run
ran	run
connections	connect
connected	connect
connecting	connect
connection	connect

Table : Stemming of words

PORTER STEMMER ALGORITHM

- PorterStemmer is of the oldest stemming algorithm originally developed in 1979. While LancasterStemmer was developed in 1990 and it uses a more aggressive approach than Porter Stemming Algorithm.
- PorterStemmer uses Suffix Stripping to produce stems.
- PorterStemmer algorithm follow a set of 05 rules for different cases that are applied in phases (step by step) to stem the words.
- It does not have a lookup table but applies algorithm to generate stem.
- It has rules whether if it should strip a suffix or not.

Graph 1: Flowchart of porter stemmer :

Step 1 deals with plurals and past participles.

Step 1 a:

- SSES -> SS (Example : caresses -> caress)
- IES -> I (Example : ponies -> poni ; ties -> ti)
- SS -> SS (Example : caress -> caress)
- S -> (Example : cats -> cat)



Step 1b:

- (m>0) EED -> EE (Example : feed -> feed ; agreed -> agree)
- (v) ED -> (Example : plastered -> plaster ; bled -> bled)
- (v) ING -> (Example : motoring -> motor ; sing -> sing)
- S -> (Example : cats -> cat)



Step 2:

- (m>0) ATIONAL -> ATE (Example : relational -> relate
- (m>0) TIONAL -> TION (Example : conditional -> condition ;
rational -> rational)
- (m>0) ENCI -> ENCE (Example : valenci -> valence)
- (m>0) ANCI -> ANCE (Example : hesitanci -> hesitance)
- (m>0) IZER -> IZE (Example : digitizer -> digitize)
- (m>0) ABLI -> ABLE (Example : conformabli -> conformable)
- (m>0) ALLI -> AL (Example : radicalli -> radical)

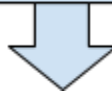


Step 3:

- (m>0) ICATE -> IC (Example : triplicate -> triplic)
- (m>0) ATIVE -> (Example : formative -> form)
- (m>0) ALIZE -> AL (Example : formalize -> formal)
- (m>0) ICITI -> IC (Example : electriciti -> electric))
- (m>0) ICAL -> IC (Example : electrical -> electric)



- (m>1) AL -> (Example : revival -> reviv)
- (m>1) ANCE -> (Example : allowance -> allow)
- (m>1) ENCE -> (Example : inference -> infer)
- (m>1) ER -> (Example : airliner -> airlin)
- (m>1) IC -> (Example : gyroscopic -> gyroscop)
- (m>1) ABLE -> (Example : adjustable -> adjust)
- (m>1) IBLE -> (Example : defensible -> defens)
- (m>1) ANT -> (Example : irritant -> irrit)



Step 5:

m > 1 and *_d and *_L) -> single letter
(Example : controll -> control ; roll -> roll)

Lemmatization

Lemmatization is generally confused with stemming .Although it produces sometimes the same result sometimes but its way too different. In lemmatization

We are actually able to capture the canonical forms that is the actual root word which is present in this language.

For example, stemming the word "better" would fail to return its citation form (another word for lemma); however, lemmatization would result in the following:

1. better -> good
2. Corpora -> corpus

It should be easy to see why the implementation of a stemmer would be the less difficult feat of the two.

Applications of lemmatization:

- Search Engine :Used in comprehensive retrieval systems like search engines.
- Compact indexing :Used in compact indexing

Difference between Lemmatization and Stemming :

- Although both Stemming and Lemmatization generate the root form of the passed words. The main difference is that lemma is an actual language word whereas stem might not be an actual word.
- If we compare the speed and accuracy then Stemming follows an algorithm with steps to perform on the words which makes it much faster than lemmatization . Whereas, in lemmatization, we use corpus for example corpus like WordNet corpus and a corpus for stop words as well to produce lemma which makes it slower than stemming. There is also option of part of speech tagging .

Corpus: A corpus is defined as defined as a collection of text documents. It is a bunch of text files in a directory. It is a latin word which means body often alongside many other directories of text files.

Corpora can also consists themed texts like historical ,biblical . It is a important part for NLP Processes over language as its the body to analyze.

Stop Words

Stop Words are those words which contribute very little almost negligible. That is removing those words does not change the meaning of the sentence. That is they are the extra words in a sentence not helping in analysis of the sentence. For example- “the”, “and”, “a”. Hence its good to filter these words before processing of the text.

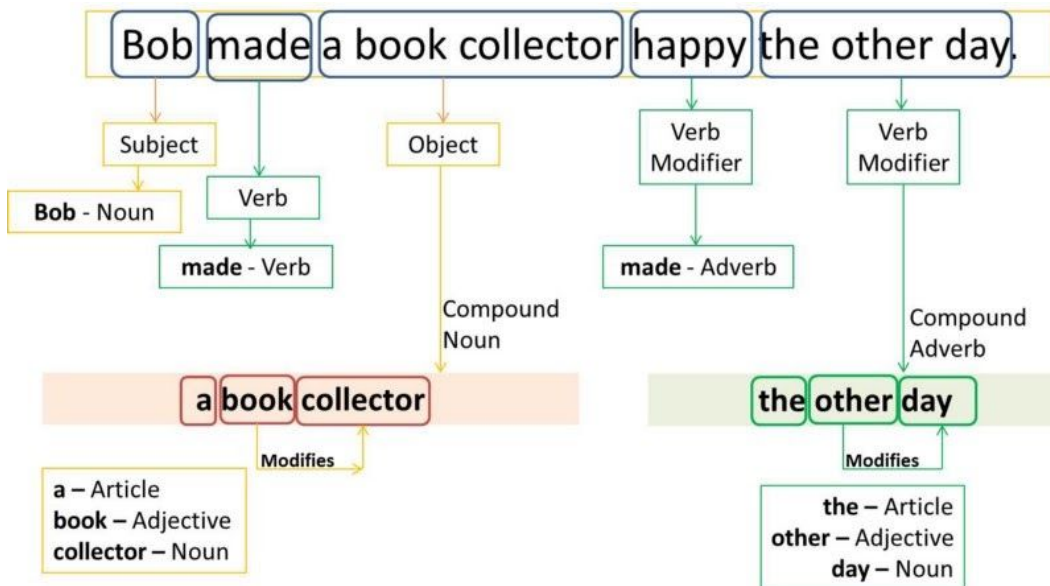
Ex- The quick brown fox jumps over the lazy dog.

In this sentence the is written but is of no use hence it should be removed before text analysis.

Part Of Speech Tagging

This technique is also very useful for identifying different parts of speech like noun, verb, pronoun

Graph 2: Part of Speech Tagging



Bag of Words

Bag of Words are generally based on a set theory .This concept is super useful as it helps to get a frequency count of words . for ex-

"Well, well, well ,well , very well done" said John.

"There, there,there you go" said James. "There, there exactly there."

Bag Of words will be:

{

Well : 5,

Very : 1,

Done :1,

Said :1 ,

John:1,

There:6,

You : 1,

Go :1,

James: 1,

Exactly :1

}

CHAPTER 3 - SYSTEM DEVELOPMENT

Bayes Theorem is the probability theorem used in our Spam Filtration Project . Its the simplest and most powerful theorem.

$$P(A/B) = (P(B / A) * P (A)) / P(B)$$

Where A and B are events and $P(B) \neq 0$

$P(A)$: Probability of A without regard to any other

$P(B)$: Probability of B without regard to any other

$P(A/B)$: Probability of observing event A given B is also true

$P(B/A)$:Probability of observing event B given A is also true

We have a message m and set of unique words ($w_1, w_2, w_3, \dots, w_m$)

$$P(\text{spam} | w_1 \cap w_2 \cap w_3 \cap \dots \cap w_n) =$$

$$P(w_1 \cap w_2 \cap w_3 \dots \cap w_n | \text{spam}) * P(\text{spam}) / P(w_1 \cap w_2 \cap w_3 \dots \cap w_n)$$

If we assume occurrence of words are independent of each other the simplified expression will be :

$$P(w_1 | \text{spam}) P(w_2 | \text{spam}) \dots P(w_n | \text{spam}) / P(w_1) P(w_2) P(w_n)$$

Hence to determine we classify which is greater:

$$P(\text{spam} | w_1 \cap w_2 \cap w_3 \cap \dots \cap w_n) \text{ versus } P(\sim \text{spam} | w_1 \cap w_2 \cap w_3 \cap \dots \cap w_n)$$

Bag Of Words:

This technique helps to find the frequency of words that is the number of occurrences of words in the dataset.

$$P(w) = \text{Total Number Of Occurrences of } w \text{ in a dataset} / \text{Total Number of words in a dataset}$$

And

$$P(w | \text{spam}) = \text{Total number of occurrences of } w \text{ in a spam message} / \text{Total Number of words in spam messages}$$

Term Frequency - Inverse Document Frequency

Inverse Document Frequency (IDF) (w)=

$\log(\text{Total Number Of Messages} / \text{Total Number of messages containing } w)$

For example - if there are 2 messages in the dataset “hello world” and “hello foobar”

Term Frequency of Hello is : 2

IDF (“hello”) = $\log(2/2)$

In case if a word occurs a lot it gives less information about the content. In this IDF model each word has a score which is counted as $TF(w) * IDF(w)$.

Probability of each word :

$P(w) = (TF(w) * IDF(w)) / \sum \forall \text{ words } x \in \text{train dataset } TF(x) * IDF(x)$

$P(w|\text{spam}) = (TF(w|\text{spam}) * IDF(w)) / \sum \forall \text{ words } x \in \text{train dataset } TF(x|\text{spam}) * IDF(x)$

Additive Smoothing

In case we encounter word in a dataset which is not a part of train dataset in that case $P(w) = 0$ which will definitely make $P(w | \text{spam}) = \text{undefined}$. To solve this problem we introduce additive smoothing. So we

1. add a number alpha to the numerator
2. add alpha times number of classes over which the probability is found in the denominator.

$$P(w|\text{spam}) = \frac{TF(w|\text{spam}) + \alpha}{(\sum \forall \text{ words } x \in \text{train dataset } TF(\text{spam in train dataset}) TF(x) + \alpha \sum \forall \text{ words } x \in \text{spam in train dataset } 1)}$$

Then we use TF - IDF:

$$P(w|\text{spam}) = \frac{TF(w|\text{spam}) * IDF(w) + \alpha}{\sum \forall \text{ words } x \in \text{train dataset } TF(x) * IDF(x) + \alpha \sum \forall \text{ words } x \in \text{spam in train dataset } 1}$$

$$\sum \forall \text{ words } x \in \text{train dataset } TF(x) * IDF(x) + \alpha \sum \forall \text{ words } x \in \text{spam in train dataset } 1$$

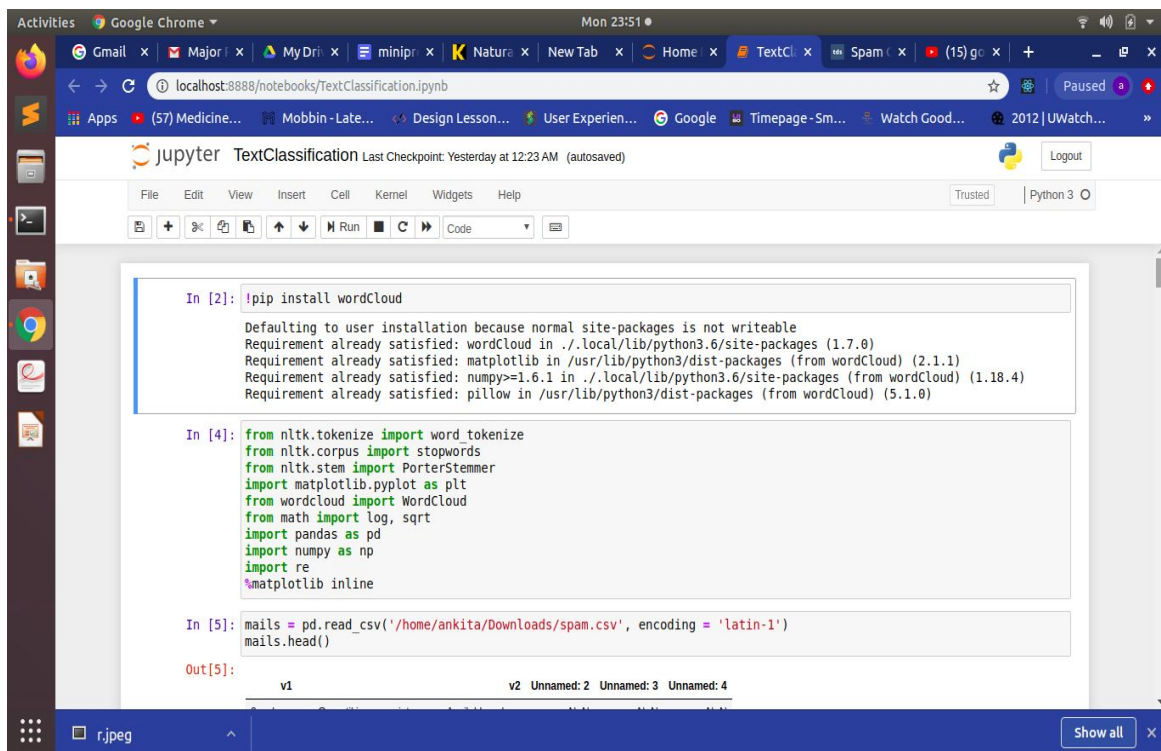
This is done so that probability of any number should be finite number. Addition in the denominator is done to make the resultant sum of all probabilities of words in spam = 1

When $\alpha = 1$ it is called Laplace smoothing.

CHAPTER - 4 PERFORMANCE ANALYSIS

Step 1: The most important step is to import the libraries like

- wordcloud for visual representation of the words
- nltk that is the natural language processing which is the most important library and helps import stopwords dataset , Porterstemmer algorithm
- Pandas is used for data manipulation and data analysis
- Numpy is a multidimensional array used to store values of same datatype.



```
In [2]: !pip install wordCloud

Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: wordCloud in ./local/lib/python3.6/site-packages (1.7.0)
Requirement already satisfied: matplotlib in /usr/lib/python3/dist-packages (from wordCloud) (2.1.1)
Requirement already satisfied: numpy>=1.6.1 in ./local/lib/python3.6/site-packages (from wordCloud) (1.18.4)
Requirement already satisfied: pillow in /usr/lib/python3/dist-packages (from wordCloud) (5.1.0)

In [4]: from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from math import log, sqrt
import pandas as pd
import numpy as np
import re
%matplotlib inline

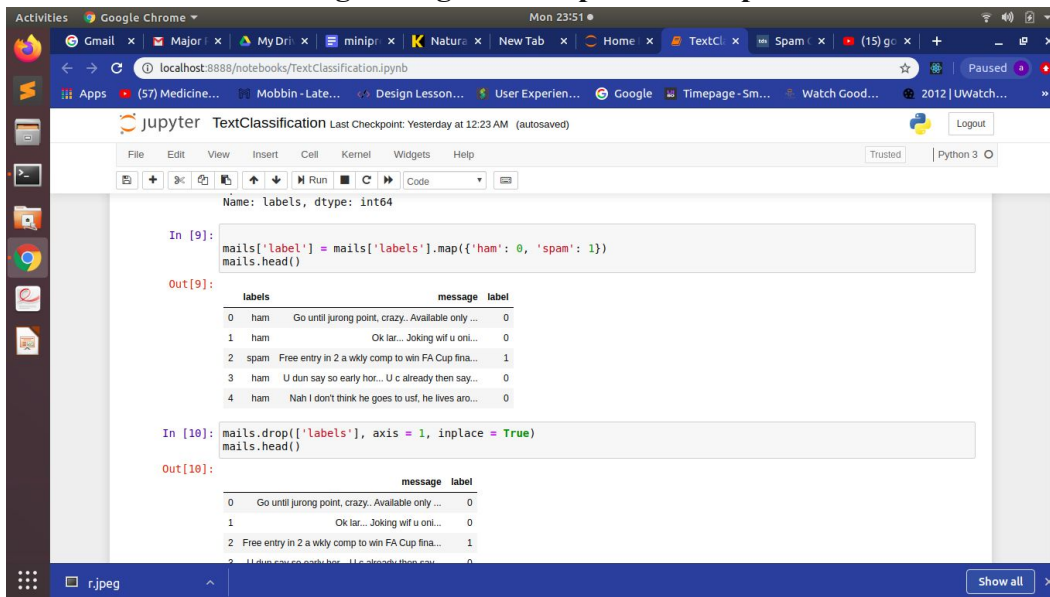
In [5]: mails = pd.read_csv('/home/ankita/Downloads/spam.csv', encoding = 'latin-1')
mails.head()

Out[5]:
```

Image 1: Step 1 showed with output

Step 2: Now lets load the database set downloaded from google and count the labels of spam and ham .Also we view the first 5 entries.

Image 2: Figure of step 2 with output



Step 3 :Now we map the ham with 0 and spam with 1 label and count respectively

Image 3: Figure of step 3 with output

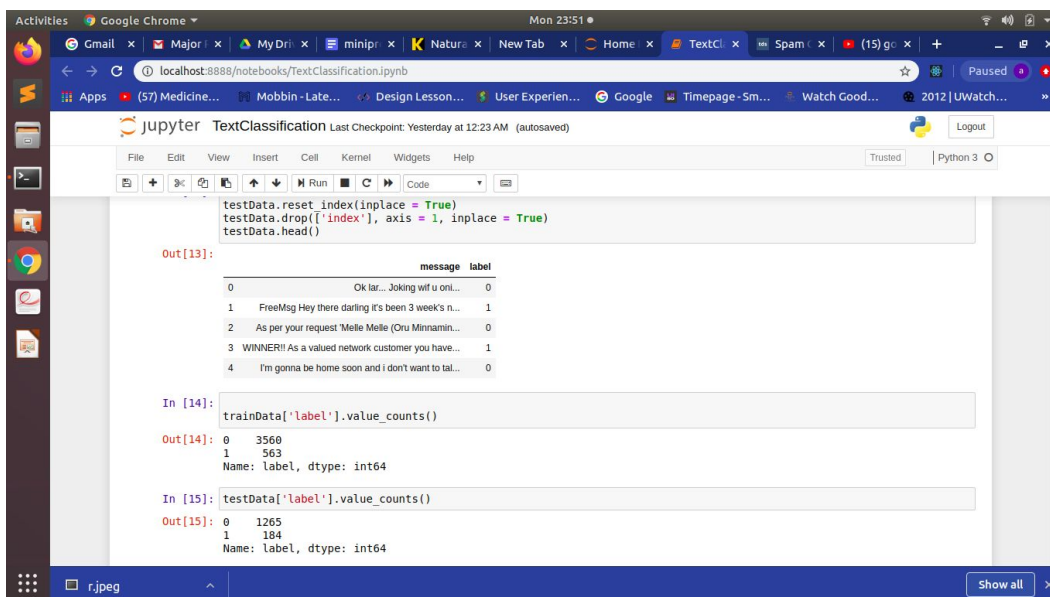
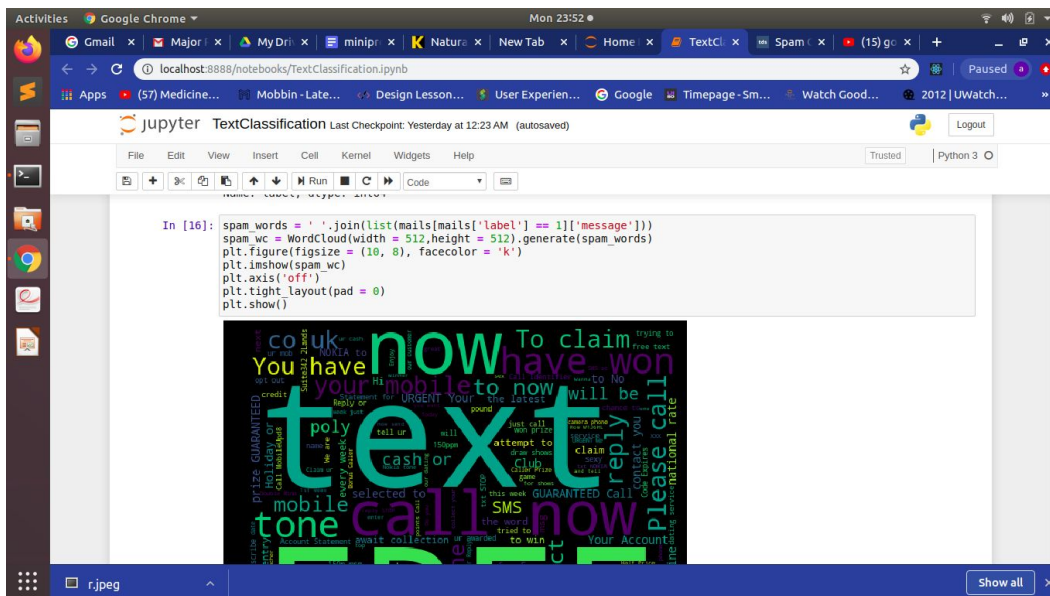


Image 4: Test and train the dataset



Step 4: Using word cloud to find the most of words in th spam category

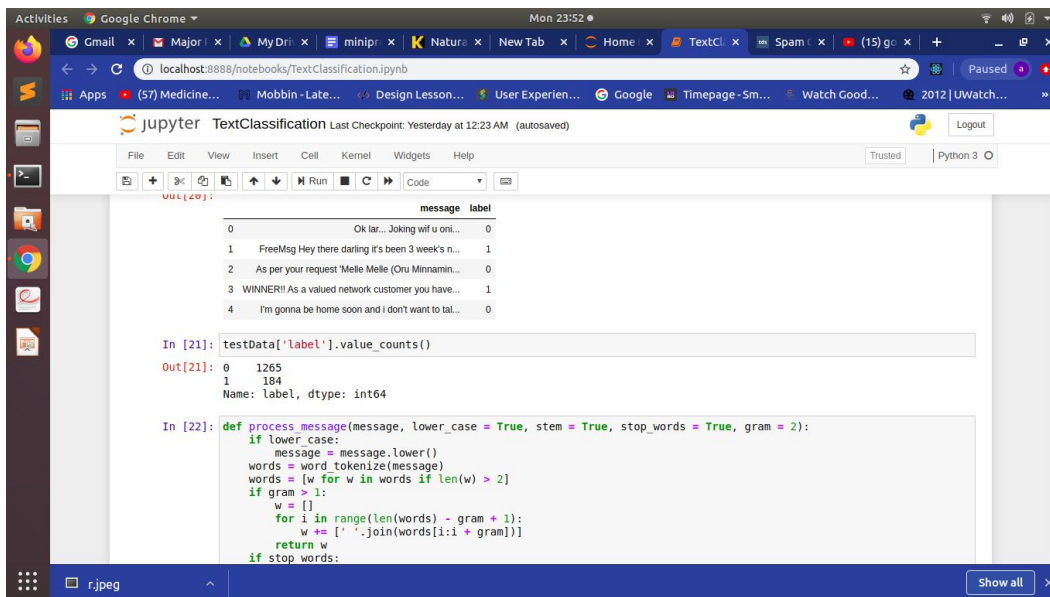
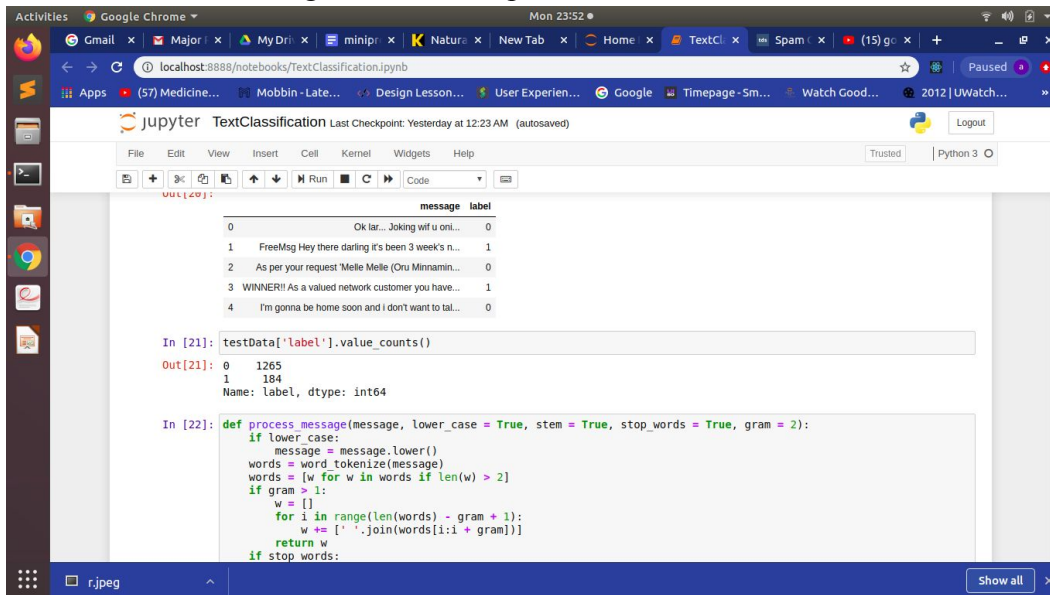


Image 7: the fig shows the processing of data

Step 6: Find the count of spam and ham in the train and test data

Image 8: Count of spam and ham in dataset



The screenshot shows a Jupyter Notebook interface with the following content:

```
Out[20]:
```

message	label
0 Ok lar... Joking wif u oni...	0
1 FreeMsg Hey there darling it's been 3 week's n...	1
2 As per your request Melle Melle (Oru Minnamin...	0
3 WINNER!! As a valued network customer you have...	1
4 I'm gonna be home soon and i don't want to tal...	0

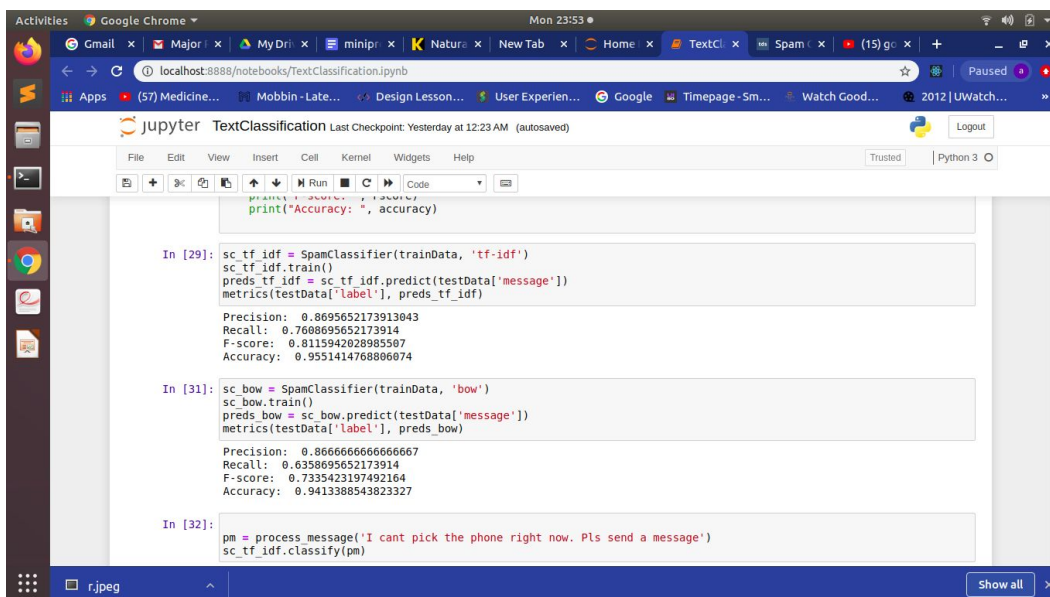
```
In [21]: testData['label'].value_counts()
```

```
Out[21]: 0    1265  
         1     184  
         Name: label, dtype: int64
```

```
In [22]: def process_message(message, lower_case = True, stem = True, stop_words = True, gram = 2):  
         if lower_case:  
             message = message.lower()  
         words = word_tokenize(message)  
         words = [w for w in words if len(w) > 2]  
         if gram > 1:  
             w = []  
             for i in range(len(words) - gram + 1):  
                 w += [' '.join(words[i:i + gram])]  
         return w  
         if stop_words:
```

Step 7: Apply function to classify words as spam and ham

Image 9: Applying classification on dataset



Step 8: Find out the precision ,accuracy , f-score ,recall

Image 10 : Finding out precision , accuracy ,f - score of the dataset

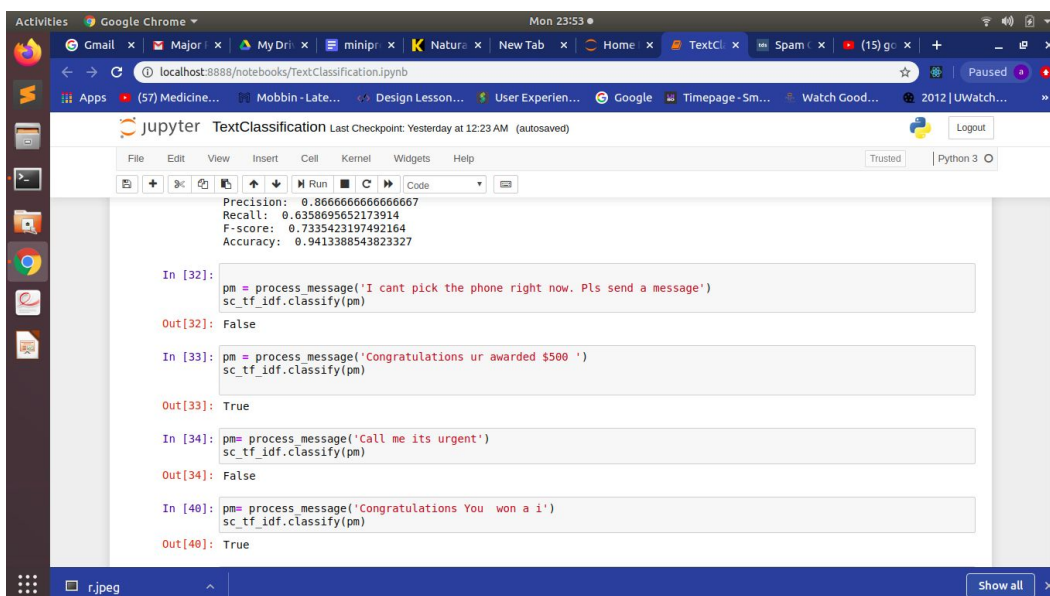


Image 11: Finally checking the trained test by testing it with input

Step 9: Finally testing the trained data by passing our messages in string format we get output as boolean .

Step 10 : Finally we have trained and tested our data.

Conclusion

In the new tech savvy world data is something most essential in the world new so my project is very useful in that sense. Spam Filtration is project which can be very useful in the world full of waste and unwanted ,untreated data .It can be used at all levels be it personal, professional global .This can act as a boon for the people to make life simpler and easier .Its a effort to make life easier and manage time .

REFERENCES

- www.youtube.com
- www.google.com
- www.wikipedia.com
- www.medium.com
- www.kdnuggets.com
- www.geeksforgeeks

PLAG REPORT

The screenshot displays a plagiarism report titled "Report-Plag". At the top, it shows an "ORIGINALITY REPORT" with four categories: SIMILARITY INDEX (20%), INTERNET SOURCES (13%), PUBLICATIONS (8%), and STUDENT PAPERS (17%). Below this, a section titled "PRIMARY SOURCES" lists four items:

Rank	Source	Percentage
1	Submitted to Ardhi University of Tanzania Student Paper	4%
2	www.datacamp.com Internet Source	3%
3	www.kdnuggets.com Internet Source	2%
4	Submitted to Le Jardin Academy Student Paper	1%

At the bottom of the report, it indicates "Page 31 / 33". The browser's address bar shows several tabs, including "Mobbin - Late...", "Design Lesson...", "User Experien...", "Google", "Timepage - Sm...", "Watch Good...", and "2012 | UWatch...".

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date:.....15 July 2020.....

Type of Document PhD M Tech Dissertation / B Tech Project (Tick):

Name: ANKITA CHAUHAN Department: CSE EnrolmentNo
 161254 Contact No.9805102318

[E-mail.chauhanankita1821@gmail.com](mailto:chauhanankita1821@gmail.com)

Name of the Supervisor: Yugal kumar Title of the Thesis/Dissertation/Project Report/Paper (in Capital letters):

spam filtration classification using nlp

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages=50
- Total No. of Preliminary pages= 19
- Total No. of pages accommodate bibliography/references=3



(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as-per-norms and found **Similarity Index at 20(%)**. Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none"> ● All Preliminary Pages ● Bibliography/Images/Quotes ● 14 WordsString 		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com