In [0]:

```
from google.colab import drive
drive.mount('gdrive',force_remount=True)
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/aut
h?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleu
sercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&respon
se_type=code&scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fd
ocs.test%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%
2f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2
fwww.googleapis.com%2fauth%2fpeopleapi.readonly (https://accounts.goog
le.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc
0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoaut
h%3a2.0%3aoob&response_type=code&scope=email%20https%3a%2f%2fwww.googl
eapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2faut
h%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive.photos.rea
donly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly)

Enter your authorization code:
··········
Mounted at gdrive
```

In [0]:

```
import os
import cv2
import json
import re
import shutil
import numpy as np
import tarfile
import pickle
from bs4 import BeautifulSoup
import sys
import joblib
from functools import reduce
import operator
import multiprocessing


import random

from matplotlib import patches
from itertools import chain
import datetime
from tqdm import tqdm
from zipfile import ZipFile


%matplotlib inline
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

import numpy as np
from pathlib import Path
```

Extracting data from zip file

In [0]:

```
%%time
# Extracting Data
with ZipFile('gdrive/My Drive/emails_.zip') as f:
  f.extractall()
```

```
CPU times: user 1.49 s, sys: 641 ms, total: 2.13 s
Wall time: 3.88 s
```

In [0]:

```
# Checking number of folders in maildir

maildir_l = os.listdir('emails_')
print(len(maildir_l))
```

```
2
```

Below is sample email from a notpad

In [0]:

```
with open('emails_/yernagulahemanth/sent/112.txt','r') as f:
  print(' '.join(f.readlines()))
```

```
To: Applied Course <team@appliedaicourse.com>
 From: Hemanth Yernagula <yernagulahemanth@gmail.com>
 Subject: Re: Doubt regarding Self driving car
 I think my model is over fitting because I'm getting =
 1.7 as predicted angle for all the data points On Sun, 22 Sep 2019,
1:10 pm =
 Hemanth Yernagula, < yernag=
 ulahemanth@gmail.com > wrote: Thanks a lot.=C2=A0 = On Sun, 22 Sep 20
19, 12:54 pm Applied=
  AI Course, < team@appliedaicourse.com > wrote: Hi Hemanth, 1.In v=
 ideo sir asked to change activation function in final layer right? So
can I=
  remove atan and ruin the model? Yes, remove the atan function 2. Sir
asked t=
 o change dropout rate to 0.5 right?I changed keep prob to 0.5, am I c
orrect=
 ? And should I change remaining keep probs also Change remaining keep
probs to 0.5 3.I trained m=
 y model in Collab notebook(with you) and downloaded the model file to
my lo=
 cal system(Local system dues not have gpu) and pasted in save folder
when I=
  run "run. Py" file I'm getting error as illegal instruction =
 what shall I do? Is there any way to run this file also in Collab not
ebook?= In colab, you can't get video. Instead of=
  video, please print actual and predicted steering wheel angles and u
pload =
 screenshots when you submit assignments = Thank you
 CType: text/html
```

Extracted emaill are stored in notepad, some time each notepad is having many number of threads i.e reply to that perticular email is also in same notepad, what I observed is emails are seperated my 'wrote:' as shown in above so we shall find out such emails and consider each thread as seperate email. Along with email there is lot of noise

In [0]:

```
def mkprts(path):
        '''If the path of email(txt file) is passed this returns the number
           of emails present in that file.For exammple in same file if there
           is original email and its replay then it is considered as two emails
           and returns index point of those two emails.For any reason if file is
           not readable function switches to exception part and prints the file nam
           along with it\'s path
        '''

        parts = []
        try:
            with open(path,'r') as file:

                  lines = file.readlines()
                  start = 0
                  first_time = 1
                  for k,i in enumerate(lines):
                      if any(j == 'wrote:' for j in i.split()):
                            if first_time:
                                first_time = 0
                                continue


                            parts.append((start,k))
                            start = 0
                            start +=k


                  if k < len(lines):

                            parts.append((start,len(lines)))
        except Exception as e:
          print('AT {} {}'.format(path,e))

        return parts
```

In [0]:

```
def return_tag_wise(lines):
        '''
        Given lines of email function detects to,from subject and content of email
        of  to, subject,from and content and applies basic operations on content pa
        '''
        to_ = []
        from_ = []
        subj = []
        content = []
        not_in = ['To:','From:','Subject:','CType:']
        for i in range(len(lines)):
            if not any([j in lines[i] for j in not_in]):
                    content.append(re.sub(r'=\n','',lines[i]))
            elif 'To:' in lines[i]:
                to_.append(lines[i])
            elif 'From:' in lines[i]:
                from_.append(lines[i])
            elif 'Subject:' in lines[i]:
                subj.append(lines[i])
        return {'to':to_[0],'from':from_[0],'subj':subj[0],'content':BeautifulSoup(



def clean_stage_1(str_):
    '''
    Applies basic cleaning operations

    '''

    string = str_.lower()
    string = re.sub(r'=\d\w',' ',string)
    string = re.sub(r'=\w\d',' ',string)
    string = re.sub(r'\\n\\n','',string)
    string = re.sub(r'<=\w+">','',string)
    string = re.sub(r'\\n','',string)
    string = re.sub(r'\\r','',string)
    string = re.sub(r' \\=\\r\\n','',string)
    string = re.sub(r'=c2=a0',' ',string)
    string = re.sub(r'=e2=80=99',"'",string)
    string = re.sub(r"shouldn't","should not",string)
    string = re.sub(r"i'm","i am",string)
    string = re.sub(r"i'll","i will",string)
    string = re.sub(r"'","",string)
    string = re.sub(r"=","",string)
    string = re.sub(r"  "," ",string)
    string = re.sub(r'\s+',' ',string)


    return string
```

Getting DataFrame From Text Files

In [0]:

```
class AsignData:

      def __init__(self,to=[],subject=[],previous_content=[],content=[]):
            '''
            Each email is considered as each object and assigned to
            its corresponding to,from,subject,content and so on
            '''

            self.to = []
            self.from_=[]
            self.type_ =[]
            self.subject = []
            self.prv_cntt= []
            self.content = []
            self.file_nm = []

      def get_data(self):
           return self.to,self.subject,self.prv_cntt,self.content,self.type_,self.f
#_____________________________________________________________________________


data1   = {}
to_     = []
frm     = []
subject_= []
type_   = []
content_= []
prv_content_ =  []
file_name = []
count = 0
cc = 0
def extract_data(p):
        '''
        Given a directory of notepad having emails, extracts fiels of
        email and returns a list of dictionaries of eamils, if the path
        provided is directory itterates through this function until it
        reaches the email files
        '''
        sys.stdout.write('\r')
        l = []
        if os.path.isdir(p):
              p_l = os.listdir(p)

              for i in p_l:

                extract_data(p +'/' + i)
        elif os.path.isfile(p):

              try:

                with open(os.path.join(p),'r') as file:
                    # file = file.astype('U')
                    # print(p)
                    lines = file.readlines()

                    tag_wise = return_tag_wise(lines)

                    l =[]
                    global count
```

```
                # if the content is having wrote: then it is considered as repl

                parts = tag_wise['content'].split('wrote:')

                l = [AsignData()  for i in range(len(parts))]

                for j,k in enumerate(reversed(parts)):
                    if len(parts) == 0:
                        if j == 0:
                            l[j].to         = tag_wise['to']
                            l[j].from_      = tag_wise['from']
                            l[j].subject    = tag_wise['subj']
                            l[j].content    = k
                            l[j].file_nm    = p
                            l[j].type_      = 'c__d'
                            l[j].prv_cntt   = 'nan'

                    else:
                        if j == 0:
                              l[j].to         = tag_wise['to']
                              l[j].from_      = tag_wise['from']
                              l[j].subject    = tag_wise['subj']
                              l[j].content    = k
                              l[j].file_nm    = p
                              l[j].type_      = 'c__d'
                              l[j].prv_cntt   = 'nan'
                        else:
                              l[j].to         = tag_wise['from']
                              l[j].from_      = tag_wise['to']
                              l[j].subject    = tag_wise['subj']
                              l[j].content    = k
                              l[j].file_nm    = p
                              l[j].type_      = 'r__y'
                              l[j].prv_cntt   = l[j-1].content
            except Exception as e:
                global cc
                cc += 1
                print(e)
                sys.stdout.write('\nAt {}'.format(p))


        for i in range(len(l)):
            d = l[i].get_data()
            to_.append(d[1])
            subject_.append(d[2])
            prv_content_.append(d[3])
            content_.append(d[4])
            type_.append(d[5])
            file_name.append(d[6])
            frm.append(d[7])


        data1['File_name'] = file_name
        data1['To']   = to_
        data1['From'] = frm
        data1['Subject']  = subject_
        data1['Prvious_email']   = prv_content_
        data1['Content']   = content_
        data1['Type'] = type_
```

```
        return data1

DATA = pd.DataFrame(extract_data('emails_/yernagulahemanth'))
print('Total number of files failed to read',cc)
```

```
Total number of files failed to read 0
```

In [0]:

```
DATA
```

Out[32]:

	File_name	To	
0	emails_/yernagulahemanth/inbox/355.txt	To: yernagulahemanth@gmail.com\n	team@appliedaicourse.
1	emails_/yernagulahemanth/inbox/274.txt	To: yernagulahemanth@gmail.com\n	From: Sk <skshirajblog@gmail.c
2	emails_/yernagulahemanth/inbox/318.txt	To: yernagulahemanth@gmail.com\n	From: H <noreply@heroku.c
3	emails_/yernagulahemanth/inbox/42.txt	To: Hemanth Yernagula <yernagulahemanth@gmail....	From: Hemanth Yerr <yernagulahemanth@g
4	emails_/yernagulahemanth/inbox/58.txt	To: Hemanth Yernagula <yernagulahemanth@gmail....	From: Applied C <team@appliedaicourse.c
...	...	...	
691	emails_/yernagulahemanth/sent/66.txt	To: Applied AI Course <team@appliedaicourse.co...	From: Hemanth Yerr <yernagulahemanth@g
692	emails_/yernagulahemanth/sent/66.txt	From: Hemanth Yernagula <yernagulahemanth@gmai...	To: Applied AI C <team@appliedaicours
693	emails_/yernagulahemanth/sent/14.txt	To: sheiksaleemraza@gmail.com\n	From: Hemanth Yerr <yernagulahemanth@g
694	emails_/yernagulahemanth/sent/105.txt	To: Applied AI Course <team@appliedaicourse.co...	From: Hemanth Yerr <yernagulahemanth@g
695	emails_/yernagulahemanth/sent/105.txt	From: Hemanth Yernagula <yernagulahemanth@gmai...	To: Applied AI C <team@appliedaicours

696 rows × 7 columns

Cleaning Data

1. Email that are not having content part is removed
2. Any special characters from the subject or previous email or content part
3. Only name part in email is considered i.e for example only myname is considered from myname@project.com

In [0]:

```
DATA  = DATA[DATA.Content != 'nan ']
DATA.index = [i for i in range(DATA.shape[0])]
print('Shape of the data after removing "nan" content',DATA.shape)
```

```
Shape of the data after removing "nan" content (531, 7)
```

In [0]:

```
# saving file

DATA.to_csv('gdrive/My Drive/google/DATA.csv',index=False)
```

In [0]:

```
final_data_unprocessed = pd.read_csv('gdrive/My Drive/google/DATA.csv')
print('Shape of data:',final_data_unprocessed.shape)
```

```
Shape of data: (531, 7)
```

In [0]:

```
# final_data_unprocessed = DATA.copy()
```

In [0]:

```
final_data_unprocessed.head(5)
```

Out[24]:

	File_name	To	Fr
0	emails_/yernagulahemanth/inbox/355.txt	To: yernagulahemanth@gmail.com\n	Fr team@appliedaicourse.co
1	emails_/yernagulahemanth/inbox/318.txt	To: yernagulahemanth@gmail.com\n	From: Her <noreply@heroku.com
2	emails_/yernagulahemanth/inbox/58.txt	To: Hemanth Yernagula <yernagulahemanth@gmail....	From: Applied Cou <team@appliedaicourse.com
3	emails_/yernagulahemanth/inbox/38.txt	To: "Hemanth Yernagula" <yernagulahemanth@gmai...	From: "Jigsaw Acade <info@jigsawacademy.com
4	emails_/yernagulahemanth/inbox/160.txt	To: yernagulahemanth@gmail.com\n	From: "Applied AI Cou (Classroom)" <no-re

In [0]:

```
n_points = 0
def clean_string(sentance,type_):
    '''When a string is passed to this function and if type of the string is given
    of the email, subject  of the email then the clean string is returned'''
    global n_points
    # if any(sentance):

    sentance = str(sentance)
    # print('\n\nstarting sent',sentance)
    sys.stdout.write('{} Remaining-{}'.format(type_,n_points))


    sentance = sentance.lower()
    sentance = re.sub(r'=\?utf-8\?q\?\w*','',sentance)
    sentance = re.sub(r'=\?utf-8\?b\?\w*','',sentance)
    sentance = re.sub(r'\\r','',sentance)
    sentance = re.sub(r'\\n','',sentance)
    sentance = re.sub(r'\\b','',sentance)
    sentance = re.sub(r'\\t','',sentance)
    sentance = re.sub(r"to:",'',sentance)
    sentance = re.sub(r"from:",'',sentance)
    sentance = re.sub(r"subject:",'',sentance)
    sentance = re.sub(r"won't", "will not", sentance)
    sentance = re.sub(r"what's", "whats", sentance)
    sentance = re.sub(r"email's", "emails", sentance)
    sentance = re.sub(r"can\'t", "can not", sentance)
    sentance = re.sub(r"\'ve", " have", sentance)
    sentance = re.sub(r'^https?:\/\/.*[\r\n]*',' ',sentance)
    sentance = re.sub(r"^(https?:\/\/)?([\da-z\.-]+)\.([a-z\.]{2,6})([\/\w \.-]*)\/
    sentance = re.sub(r"\'m", " am", sentance)
    sentance = re.sub(r"=?utf-8?q?", " ", sentance)
    sentance = re.sub(r"=?utf-8?", " ", sentance)

    sentance = re.sub(r"\'d", " would", sentance)
    sentance = re.sub(r"\'ll", " will", sentance)
    sentance = re.sub(r"\'t", " not", sentance)
    sentance = re.sub(r"n\'t", " not", sentance)
    sentance = re.sub(r"\'re", " are", sentance)
    sentance = re.sub(r"\'re", " are", sentance)



    if type_ == 'subject':

          "['Subject: Start Date: 4/25/01; HourAhead hour: 3;  <CODESITE>\\n']"
          sentance = sentance.lower()
          sentance = re.sub(r'\w*@\w*\son\s\w{3},\s\w{3}\s\d*,\s\d*\sat\s\d*:\d*\s\
          sentance = re.sub(r'on\s\w{3},\s\w{3}\s\d*,\s\d*\sat\s\d*:\d*\s\w*\s\w*\s
          sentance = re.sub(r'on\s\w{3},\s\w{3}\s\d*,\s\d*,\s\d*:\d*\s\w*\s\w*\s\w*
          sentance =re.sub(r'\w*@\w*.com\s>','',sentance)
          sentance = re.sub(r"^(https?:\/\/)?([\da-z\.-]+)\.([a-z\.]{2,6})([\/\w \.
          sentance = re.sub("https?://.*",'',sentance)
          # sentance = re.sub('[^A-Za-z0-9]+', ' ', sentance)
          sentance = BeautifulSoup(sentance).get_text()
          sentance = re.sub(r"subject:",'',sentance)
          sentance = re.sub(r";",' ',sentance)
          sentance = re.sub(r"start date:",'',sentance)
          # sentance = re.sub(r"\d{1}/\d{2}/\d{2}"," ",sentance)
```

```
        # sentance = re.sub(r"\d{2}/\d{2}/\d{2}"," ",sentance)
        # sentance = re.sub(r"\d{1}/\d{1}/\d{2}"," ",sentance)
        sentance = re.sub("fw:"," ",sentance)
        sentance = re.sub(r"re:"," ",sentance)
        sentance = re.sub('[^A-Za-z0-9]+', ' ', sentance)
        sentance = re.sub(" hemanth yernagula"," yernagulahemanth ",sentance)
        sentance = re.sub(" hemanth"," yernagulahemanth ",sentance)
        sentance = re.sub(" hemanth "," yernagulahemanth ",sentance)
        sentance = re.sub("saiteja","saitejapsk",sentance)
        # sentance = re.sub("re tw"," ",sentance)
        # sentance = re.sub(r",",'',sentance).strip()
        sentance = re.sub(r"  "," ",sentance)

    elif type_ == 'to':

        # sentance = re.search(r'<\w*\@\w*.com>',sentance).group()
        try:
          sentance = re.search(r'\w*\@',sentance).group()[:-1]
          sentance = re.sub("psksaiteja1","saitejapsk",sentance)
          # sentance = re.sub(r"^(https?:\/\/)?([\da-z\.-]+)\.([a-z\.]{2,6})([\/\
          # sentance = sentance.split('@')[0]
          # sentance = re.sub('[^A-Za-z]+', ' ', sentance)
          # sentance = re.sub(r",",'',sentance).strip()
        except:
          pass


    elif type_ == 'content':

        # sentance = re.sub(r"\\n",'',s).lower()
        sentance = BeautifulSoup(sentance).get_text()
        sentance = re.sub(r'\w*@\w*\son\s\w{3},\s\w{3}\s\d*,\s\d*\sat\s\d*:\d*\s\
        sentance = re.sub(r'on\s\w{3},\s\w{3}\s\d*,\s\d*\sat\s\d*:\d*\s\w*\s\w*\s
        sentance = re.sub(r'on\s\w{3},\s\w{3}\s\d*,\s\d*,\s\d*:\d*\s\w*\s\w*\s\w*
        sentance = re.sub(r'on\s\w{3},\s\w{3}\s\d*,\s\d*\s\w*\s\d*:\d*\s\w*','',s
        sentance = re.sub("https?://.*",'',sentance)#removing urls
        sentance = re.sub(r'\w*@\w*.com\s>','',sentance)#removing urls
        sentance = re.sub(r'\w*_\d*.\w{3}','',sentance)
        sentance = re.sub(r'\www.\w*.com','',sentance)#removing urls
        sentance = re.sub(r"\d+",'',sentance)
        sentance = re.sub(r"td>",' ',sentance)
        sentance = re.sub(r"div>",' ',sentance)
        sentance = re.sub(r"^(https?:\/\/)?([\da-z\.-]+)\.([a-z\.]{2,6})([\/\w \.
        sentance = re.sub(r"\w{2}\s\d{2}/\d{2}/\d{4}\s\d{2}:\d{2}:\d{2}\s\w{2},",
        sentance = re.sub(r"[^A-Za-z0-9]+",' ',sentance)
        sentance = re.sub(r"\d*",'',sentance)
    n_points -= 1
    sentance = re.sub(r'  ',' ',sentance)
    sys.stdout.write('\r')


    return sentance
```

We shall have a look at how this clean function is working on each column

On To Column

In [0]:

```
print(re.search(r'\w*\@','d@').group()[:-1])
print('n',final_data_unprocessed.To.iloc[73])
print(clean_string(final_data_unprocessed.To.iloc[73],type_='to'))

for i in range(10):
  print('-'*100)
  r = random.randint(10,100)
  print('Before cleaning')

  print(r,final_data_unprocessed.To.iloc[r])

  print('After cleaning')
  print(clean_string(final_data_unprocessed.To.iloc[r],type_='to'))
```

```
d
n To: Hemanth Yernagula <yernagulahemanth@gmail.com>

yernagulahemanth
----------------------------------------------------------------------
--------------------------------
Before cleaning
97 To: Hemanth Yernagula <yernagulahemanth@gmail.com>

After cleaning
yernagulahemanth
----------------------------------------------------------------------
--------------------------------
Before cleaning
45 To: yernagulahemanth@gmail.com

After cleaning
yernagulahemanth
----------------------------------------------------------------------
```

In [0]:

```
# final_data_unprocessed = final_data_unprocessed.drop([890],axis=0)
# final_data_unprocessed.to_csv('gdrive/My Drive/google/my_final_data_unprocessed.c
final_data_unprocessed.Subject.iloc[151]
```

Out[284]:

```
'Subject: =?UTF-8?B?4oCcMzYgQW1hemluZyBQeXRob24gT3BlbiBTb3VyY2UgUHJvam
VjdHMgKHYuMjAxOSnigJ0gcHVibGlzaGVkIGluIE15YnJpZGdlIA==?= =?UTF-8?B?Zm9
yIFByb2Zlc3Npb25hbHMgYnkgTXlicmlkZ2U=?=\n'
```

On subject Column

In [0]:

```
print(final_data_unprocessed.Subject.iloc[151])
print(clean_string(final_data_unprocessed.Subject.iloc[151],type_='subject'))
for i in range(10):
  print('-'*100)
  r = random.randint(10,500)
  print('Before cleaning')

  print(r,final_data_unprocessed.Subject.iloc[r])

  print('After cleaning')
  print(clean_string(final_data_unprocessed.Subject.iloc[r],type_='subject'))
```

```
Subject: Your Coupons

your coupons ning--21
----------------------------------------------------------------------------------------------------
--------------------------------
Before cleaning
330 Subject: Welcome to Learn to Chant Ashtadhyayi

After cleaning
welcome to learn to chant ashtadhyayi
----------------------------------------------------------------------------------------------------
--------------------------------
Before cleaning
303 Subject: 4 Ways to Get Started with AWS

After cleaning
4 ways to get started with aws
----------------------------------------------------------------------------------------------------
--------------------------------
Before cleaning
```

On Content Column

In [0]:

```
for i in range(10):
  print('-'*100)
  r = random.randint(10,500)
  print('Before cleaning')

  print(r,final_data_unprocessed.Content.iloc[r])

  print('After cleaning')
  print(clean_string(final_data_unprocessed.Content.iloc[r],type_='content'))
```

```
----------------------------------------------------------------------------------------------------
Before cleaning
53 yes, its a good one :) but we feel there are few tools available
for this task, like ocr. can you explain to us what are steps you ar
e planning to take while solving this problem? thank you
After cleaning
yes its a good one but we feel there are few tools available for thi
s task like ocr can you explain to us what are steps you are plannin
g to take while solving this problem thank you
----------------------------------------------------------------------------------------------------
Before cleaning
54 aaic classroom app hi , classroom: appliedaicourse: baymax post t
itle: docboyz company post content: preview content about company :
docboyz role: ml intern location: pune   interviews: 2 to 3 ml inter
views . requirements: 1. very good programming knowledge p> 2. very
good knowledge of ml and deep learning 3. proven track record of app
lying deep learning and machine learning   note: preference would be
```

On Previous Eamil Column

In [0]:

```
## Cleaning previous email column

for i in range(10):
  print('-'*100)
  r = random.randint(10,100)
  print('Before cleaning')

  print(r,final_data_unprocessed.Prvious_email.iloc[r])

  print('After cleaning')
  print(clean_string(final_data_unprocessed.Prvious_email.iloc[r],type_='content'))
```

```
----------------------------------------------------------------------------------------------------
------------------------------
Before cleaning
41 nan
After cleaning
nantent Remaining--42
----------------------------------------------------------------------------------------------------
------------------------------
Before cleaning
18 nan
After cleaning
nantent Remaining--43
----------------------------------------------------------------------------------------------------
------------------------------
Before cleaning
76 nan
After cleaning
nantent Remaining--44
----------------------------------------------------------------------------------------------------
------------------------------
Before cleaning
60 nan
After cleaning
nantent Remaining--45
----------------------------------------------------------------------------------------------------
------------------------------
Before cleaning
36 nan
After cleaning
nantent Remaining--46
----------------------------------------------------------------------------------------------------
------------------------------
Before cleaning
14 nan
After cleaning
nantent Remaining--47
----------------------------------------------------------------------------------------------------
------------------------------
Before cleaning
98 nan
After cleaning
nantent Remaining--48
----------------------------------------------------------------------------------------------------
------------------------------
Before cleaning
23 nan
After cleaning
```

```
nantent Remaining--49
------------------------------------------------------------------------
-----------------------------
Before cleaning
23 nan
After cleaning
nantent Remaining--50
------------------------------------------------------------------------
-----------------------------
Before cleaning
12  hello mrs. hemanthcasestudy4 , are will you come by car, bus or by
train?
After cleaning
hello mrs yernagulahemanth casestudy are will you come by car bus or b
y train
```

Cleaninig All Columns

In [0]:

```
n_points = final_data_unprocessed.shape[0]
final_data_unprocessed['clean_to'] = final_data_unprocessed.To.apply(lambda x: clea

sys.stdout.write('\r')
sys.stdout.write('Done with to')

n_points = final_data_unprocessed.shape[0]
final_data_unprocessed['clean_subject'] = final_data_unprocessed.Subject.apply(lamb

sys.stdout.write('\r')
sys.stdout.write('Done with subject')

n_points = final_data_unprocessed.shape[0]
final_data_unprocessed['clean_content'] = final_data_unprocessed.Content.apply(lamb

sys.stdout.write('\r')
sys.stdout.write('Done with content')

n_points = final_data_unprocessed.shape[0]
final_data_unprocessed['clean_previous_email'] = final_data_unprocessed.Prvious_ema
```

In [0]:

```
final_data_unprocessed.head()
```

Out[39]:

	File_name	To	Fr
0	emails_/yernagulahemanth/inbox/355.txt	To: yernagulahemanth@gmail.com\n	Fr team@appliedaicourse.co
1	emails_/yernagulahemanth/inbox/318.txt	To: yernagulahemanth@gmail.com\n	From: Her <noreply@heroku.com
2	emails_/yernagulahemanth/inbox/58.txt	To: Hemanth Yernagula <yernagulahemanth@gmail....	From: Applied Cou <team@appliedaicourse.com
3	emails_/yernagulahemanth/inbox/38.txt	To: "Hemanth Yernagula" <yernagulahemanth@gmai...	From: "Jigsaw Acade <info@jigsawacademy.com
4	emails_/yernagulahemanth/inbox/160.txt	To: yernagulahemanth@gmail.com\n	From: "Applied AI Cou (Classroom)" <no-re

In [0]:

```
final_data_unprocessed.to_csv('gdrive/My Drive/google/my_final_data_unprocessed.csv
```

In [0]:

```
final_data_unprocessed = pd.read_csv('gdrive/My Drive/google/my_final_data_unproces
```

Lets store this cleaned columns separetely

In [0]:

```
final_data_processed = pd.DataFrame()

final_data_processed['File_name'] = final_data_unprocessed['File_name']
final_data_processed['To'] = final_data_unprocessed['clean_to']
final_data_processed['Subject'] = final_data_unprocessed['clean_subject']
final_data_processed['Previous_email'] = final_data_unprocessed['clean_previous_ema
final_data_processed['Content'] = final_data_unprocessed['clean_content']
final_data_processed['Type'] = final_data_unprocessed['Type']
```

In [0]:

```
print('Shape of the final processed data is ',final_data_processed.shape)
```

```
Shape of the final processed data is  (531, 6)
```

We shall consider emails that are having atleast 7 words in content part and emails that are having less

than 5 words because subject part must be explained with in least possible words how ever if the subject part is having more number of words even we human does not consider to read instead we read content of email

In [0]:

```
print("Shape of the data frame whose content part is greater  than 7 words",final_d
```

```
Shape of the data frame whose content part is greater  than 7 words (4
81, 6)
```

In [0]:

```
final_data_processed = final_data_processed[final_data_processed.Content.apply(lamb
final_data_processed.index = [i for i in range(final_data_processed.shape[0])]
print("After removing content rows having less than 7 words:",final_data_processed.
```

```
After removing content rows having less than 7 words: (481, 6)
```

In [0]:

```
final_data_processed[final_data_processed.Subject.apply(lambda x:len(str(x).split()
```

Out[107]:

	File_name	To	Subject	Previous_email	
3	emails_/yernagulahemanth/inbox/160.txt	yernagulahemanth	22assign	nan	q me
4	emails_/yernagulahemanth/inbox/299.txt	yernagulahemanth	git hub	nan	man pl
7	emails_/yernagulahemanth/inbox/181.txt	yernagulahemanth	to list	nan	cjxodgs
8	emails_/yernagulahemanth/inbox/168.txt	yernagulahemanth		nan	sto
10	emails_/yernagulahemanth/inbox/412.txt	yernagulahemanth	how are you comminig	nan	he
...	...	...	...	...	
469	emails_/yernagulahemanth/sent/117.txt	yernagulahemanth	regarding sql assignment	thank you for your response on sat sep pm appl...	hello te
470	emails_/yernagulahemanth/sent/77.txt	team	about final project	nan	can y
471	emails_/yernagulahemanth/sent/77.txt	yernagulahemanth	about final project	can you explain more about the problem stateme...	it is ju
476	emails_/yernagulahemanth/sent/116.txt	team	regarding sql assignment	nan	you n
477	emails_/yernagulahemanth/sent/116.txt	yernagulahemanth	regarding sql assignment	you need to be comfortable with writing nested...	thanl

198 rows × 6 columns

In [0]:

```
final_data_processed = final_data_processed[final_data_processed.Subject.apply(lamb
final_data_processed.index = [i for i in range(final_data_processed.shape[0])]
print("After removing subject rows that are not having atleast one word:",final_dat
```

```
After removing subject rows that are not having atleast one word: (46
2, 6)
```

In [0]:

```
final_data_processed = final_data_processed[final_data_processed.Subject.apply(lamb
final_data_processed.index = [i for i in range(final_data_processed.shape[0])]
print("After removing subject rows having more than 5 words:",final_data_processed.
```

```
After removing subject rows having more than 5 words: (179, 6)
```

In [0]:

```
final_data_processed.index = [i for i in range(final_data_processed.shape[0])]
```

In [0]:

```
final_data_processed.to_csv('gdrive/My Drive/google/my_final_data_processed.csv',in
```

In [0]:

```
print('Shape of processed data is :-',final_data_processed.shape)
```

```
Shape of processed data is :- (179, 6)
```

In [0]:

```
final_data_processed.head()
```

Out[113]:

	File_name	To	Subject	Previous_email	
0	emails_/yernagulahemanth/inbox/160.txt	yernagulahemanth	22assign	nan	q mer
1	emails_/yernagulahemanth/inbox/299.txt	yernagulahemanth	git hub	nan	man ple
2	emails_/yernagulahemanth/inbox/181.txt	yernagulahemanth	to list	nan	cjxodgsp
3	emails_/yernagulahemanth/inbox/412.txt	yernagulahemanth	how are you comminig	nan	hel
4	emails_/yernagulahemanth/inbox/412.txt	hemanthcasestudy4	how are you comminig	hello mrs yernagulahemanth casestudy are will ...	hello y

There are some unknown sentances which is shown below

In [0]:

```
final_data_processed.Content.iloc[2]
```

Out[114]:

```
'pcfetnuwvbfighbww cjxodgspgoagvhzdkpgldgegbmftztidmlldbvcnqiignvbnr
l bnqindpzhropwrldmljzsawracwgawpdglhbczyfsztxijkphnewxlpgpibrihsk i
cbtyxjnawidaciagbwlulxdpzhrooiayntbwedskfqoklyogswjbhvkzsbagugcgfkzg
lu zybhbmqgymyzgvyigluigfuigvszwlbnqncybbrhbcbawracbhbmqgagvpzhicovc
iog ewogigjveczaxppbmcigjvcmrlciibgcnkciqifjlbwzsbtyxjnawzigfuzcbwyw
rk awnigzybgdghligxpcqgkikdwwgewogighcmdpbjogmdskicbwywrkawnoiawowpc
gov kibtdhlszsbagugbglzdcbpdgvtcyaqlwpbcbsasbciagyvycyoibwbludgvyowo
gihbv clawuoibyzwxhdglztskicbwywrkawnoiaxmnbidhwecaxmnbidqwchgciagbg
lzdcz dhlszsexbloibublowogigjhytncmbmqicnlzwuciagzmudczaxploiaxohbow
og ihryywzaxrpbidaumnmciagciaglyogbwfrzsbagugbglzdcbpdgvtcybbnnlbgvj
dgfi bgugkikicatdviallxvzzxitcvszwnoibublowogictbotdxnlcizzwxlyqigv
bmuciaglwzlxvzzxitcvszwnoibublowogihvzzxitcvszwnoibublowpcgovkibt zx
qgywxsigkzcbsaxniglzwzihrvigegzglmzmvyzwignvbgyichzwjyyszdhjpcgvz ks
aqlwpbcbsatpudggtyhpbgqobrkksbciagymfjadybvuzdogiyzjlmotskfqoklyog r
gfyavyigjhytncmbmqtysbigbgagzxigkikdwwgbgkagzxigewogigjhytn cmbmqicn
kzgqcnkciqifdozwgyxpytlzcbvbiwgywrkigegymfjadybvuzcbjbxv cibhbmqgcry
awtligdcbzxhicovcnvsigxplmnozwnrzwqgewogigjhytncmbmqicm odgciagysbii
cnmzmvciagdgvdckzwnvcmfawuoibsawllxrocmzgcnkcig iefkzcbhicjiagvjavki
```

To eliminate sentance like above we can define a new ratio as shown below.

$$f' \frac{numberofwordsinsentance}{numberofcharectersinthesentances}$$

I observed this ratio is less than 0.09 if a sentance is not a genuine one

In [0]:

```
final_data_processed[[len(str(final_data_processed.Content.iloc[i]).split())/len(st
```

Out[115]:

	File_name	To	Subject	Previous_email	
2	emails_/yernagulahemanth/inbox/181.txt	yernagulahemanth	to list	nan	cjxodgspgoag

Lets see the ratio of random sentaces

In [16]:

```
for i in range(5):
    print('-'*50)
    print(final_data_processed.Content.iloc[i+42])
    print(len(str(final_data_processed.Content.iloc[i+42]).split())/len(str(final_d
```

```
--------------------------------------------------
hi yernagulahemanth sir what is the significance of data iris if i do
not mention data also it is working properly there is no need to menti
on the parameter explicitly but it helps to retain the parameter name
for readability regards team appliedai
0.17269076305220885
--------------------------------------------------
google verification code dear google user we received a request to acc
ess your google account through your email address your google verific
ation code is if you did not request this code it is possible that som
eone else is trying to access the google account do not forward or giv
e this code to anyone sincerely yours the google accounts team this em
ail cant receive replies for more information visit the google account
s help center google inc amphitheatre parkway mountain view ca usa
0.16632443531827515
--------------------------------------------------
read books with scribd youre welcome back anytime view this email in b
rowser hi yernagulahemanth your scribd membership has been cancelled e
ffective april if youd like to resume your scribd membership in the fu
ture all you have to do is sign back up your scribd account will be he
re and waiting to give you access to the best books audiobooks magazin
es and more resume membership and dont forget were always here to help
visit our help center for more info or feel free to contact us directl
y with any questions comments or concerns thanks team scribd join us o
n social refer a friend manage preferences contact us this email was s
ent to yernagulahemanth gmail com this email was sent by scribd bush s
treet san francisco ca united states
0.17344173441734417
--------------------------------------------------
yes yernagulahemanth can you share us the link of the problem statemen
ts can you tell us how many course case studies you have completed
0.17518248175182483
--------------------------------------------------
account dear user you are in the process of signing in to your account
the verification code is it will be valid for minutes to protect your
account do not disclose it to anyone the email is sent by the system a
utomatically please do not reply thank you
0.1889763779527559
```

Lets only consider whose ratio is more than 0.09

In [7]:

```
final_data_processed = final_data_processed[[len(str(final_data_processed.Content.i
final_data_processed.shape
```

Out[7]:

```
(164, 6)
```

In [8]:

```
final_data_processed.head()
```

Out[8]:

	File_name	To	Subject	Previous_email	
0	emails_/yernagulahemanth/inbox/299.txt	yernagulahemanth	git hub	NaN	man ple this link d
1	emails_/yernagulahemanth/inbox/412.txt	yernagulahemanth	how are you comminig	NaN	yernagul casestu
2	emails_/yernagulahemanth/inbox/412.txt	hemanthcasestudy4	how are you comminig	hello mrs yernagulahemanth casestudy are will ...	yernagul i will be
3	emails_/yernagulahemanth/inbox/381.txt	yernagulahemanth	regarding deep learning project	NaN	yes t fine words i
4	emails_/yernagulahemanth/inbox/400.txt	yernagulahemanth	webinar	NaN	hello this ever yo

Lets save the file

In [0]:

```
final_data_processed.to_csv('gdrive/My Drive/google/final_data_processed_my_emails.
```

Lets prepare this data in next document

In [0]:

```
#Done_____________________________________
```