NLP Assignment 1

Drishya Uniyal	MT21119
Harshit Gupta	MT21028

Q1. REGULAR EXPRESSION

RegEx or Regular Expression, is a sequence of characters that forms a search pattern. Python has a built-in package called re, that you need to use for this part.

A. Report the following values for each class separately.

a. average number of sentences and tokens.

Average number of sentences: -

```
positive_reactions = df_positive['TEXT'].to_list()
positive_sentences = []

for reaction in positive_reactions:
    x = re.split(r'[.!?]+', reaction)
    positive_sentences.append(len(x))

print("Average number of sentences in positive class = {}".format(sum(positive_sentences)/len(positive_sentences)))
```

```
Average number of sentences in positive class = 2.3935286401399214
```

```
negative_reactions = df_negative['TEXT'].to_list()
negative_sentences = []

for reaction in negative_reactions:
    x = re.split(r'[.!?]+', reaction)
    negative_sentences.append(len(x))

print("Average number of sentences in negative class = {}".format(sum(negative_sentences)/len(negative_sentences)))
```

```
Average number of sentences in negative class = 2.2405
```

Average number of tokens: -

```
positive_reactions = df_positive['TEXT'].to_list()
positive_tokens = []

for reaction in positive_reactions:
    x = re.findall(r'\w+', reaction)
    positive_tokens.append(len(x))

print("Average number of tokens in positive class = {}".format(sum(positive_tokens)/len(positive_tokens)))
```

```
Average number of tokens in positive class = 13.389156099693922
```

```
negative_reactions = df_negative['TEXT'].to_list()
negative_tokens = []

for reaction in negative_reactions:
    x = re.findall(r'\w+', reaction)
    negative_tokens.append(len(x))

print("Average number of tokens in negative class = {}".format(sum(negative_tokens)/len(negative_tokens)))
```

```
Average number of tokens in negative class = 14.147
```

b. total number of words starting with consonants and vowels.

```
positive_tokens = []
negative_tokens = []

for reaction in positive_reactions:
    x = re.findall(r'\w+', reaction)
    positive_tokens.extend(x)

for reaction in negative_reactions:
    x = re.findall(r'\w+', reaction)
    negative_tokens.extend(x)
```

```
re_exp = '^[AEIOUaeiou][A-Za-z0-9_]*'
re_exp1 = '[^AEIOUaeiou][a-zA-Z]*[AEIOUaeiou]'
pos_vowel_count = 0
neg_vowel_count = 0
pos_const_count = 0
neg_const_count = 0

for token in positive_tokens:
    if re.search(re_exp, token):
        pos_vowel_count = pos_vowel_count + 1
    if re.search(re_exp1, token):
        pos_const_count = pos_const_count + 1

for token in negative_tokens:
    if re.search(re_exp, token):
        neg_vowel_count = neg_vowel_count + 1
    if re.search(re_exp1, token):
        neg_const_count = neg_const_count + 1

print("Words starting with vowel in positive class: {}".format(pos_vowel_count))
print("Words starting with vowel in negative class: {}".format(neg_vowel_count))
print("Words starting with consonant in positive class: {}".format(pos_const_count))
print("Words starting with consonant in negative class: {}".format(neg_const_count))
```

```
Words starting with vowel in positive class: 7189
Words starting with vowel in negative class: 6990
Words starting with consonant in positive class: 22922
Words starting with consonant in negative class: 20728
```

c. lowercase the text and report the number of unique tokens present before and after lower casing.

```
positive_lowercase = list(map(lambda x: x.lower(), positive_tokens))
negative_lowercase = list(map(lambda x: x.lower(), negative_tokens))

print("Unique tokens present in positive class before lowercasing : {}".format(len(set(positive_tokens))))
print("Unique tokens present in negative class before lowercasing : {}".format(len(set(negative_tokens))))
print("Unique tokens present in positive class after lowercasing : {}".format(len(set(positive_lowercase))))
print("Unique tokens present in negative class after lowercasing : {}".format(len(set(negative_lowercase))))
```

```
Unique tokens present in positive class before lowercasing : 7852
Unique tokens present in negative class before lowercasing : 6417
Unique tokens present in positive class after lowercasing : 6819
Unique tokens present in negative class after lowercasing : 5620
```

d. count and list all the usernames.

This is the code we have used for counting and listing all the usernames: -

```
usernames_positive = []

for reaction in positive_reactions:
    u = re.findall('@\w+', reaction)
    if len(u) != 0:
        usernames_positive.extend(u)

print("Count of usernames in positive label: {}".format(len(usernames_positive)))
print(usernames_positive)

usernames_negative = []

for reaction in negative_reactions:
    u = re.findall('@\w+', reaction)
    if len(u) != 0:
        usernames_negative.extend(u)

print("Count of usernames in negative label: {}".format(len(usernames_negative)))
print(usernames_negative)
```

Output: - For positive label

```
Count of usernames in positive label: 1305
['@awaisnaseer', '@Marama', '@gfalcone601', '@mrstessyman', '@GetMeVideo', '@tb78', '@RealDeal32', '@yoginifoodie', '@mileycyrus', '@SCTunstal
l', '@IHauntWizards', '@soycamo', '@Liverpool_TX', '@domkoenig', '@Cyberela', '@spencerpratt', '@Bossmobb', '@cmrush', '@nachojohnny', '@teambu
dd1', '@mrskutcher', '@EastCoastGamblr', '@mitchelmusso', '@nessie_111', '@nakulshenoy', '@jeddimkins', '@DannyMcEvoy', '@Courtney_182', '@Davi
dArchie', '@Smithycurt', '@chuckiem', '@MicheleKnight', '@jacdo', '@Dj_SportsChick', '@HisFitness', '@AmazingPhil', '@Wendym00n', '@nathalichri
sty', '@MATTHARDYBRAND', '@MaryJoRs', '@bedoggtde', '@mommo9000', '@SupaSash11', '@MAYAHZONFIYA', '@egoodlett', '@angelajames', '@rafaelvandyk
e', '@rainbowsleeve', '@shaundiviney', '@stephenfry', '@zate', '@edlee', '@DonnieWahlberg', '@softandpoofyone', '@twishmay', '@TheRoundDiet',
'@romaineami', '@Joenid', '@mahika', '@calvinharris', '@jazzyfizza', '@DsBabyGirl', '@LamarLee', '@traceyfalk', '@shellrawlins', '@TheRealJordi
n', '@poetrysue', '@LittleLisa69', '@busydiscoball', '@diamondblvd', '@ChicagoLatina80', '@sj32', '@tommcfly', '@caribouboy', '@frommystudio',
'@DHughesy', '@kristenstewart9', '@linux_nut', '@gkarageorge', '@Harkaway', '@WelshDrew', '@DufalBagEnt', '@drewie123', '@DavidArchie', '@Gailp
orter', '@LaniiBlunts', '@bruxedo', '@windy6', '@KathyBuckworth', '@tiiiink', '@peterfacinelli', '@microgeist', '@djtracyyoung', '@djannalyze',
'@theRoose', '@bigbadbob75', '@LowcountryBBQ', '@HMXCasey', '@Chantalalalaxo', '@Yasmimmm', '@Fejennings', '@hollienicole', '@KatieP2008', '@te
enhearts', '@katgkionis', '@jakesonaplane', '@FrankMillar', '@em26miles', '@smoshian', '@epiphanygirl', '@Yvette_Syversen', '@Sexyjoy386', '@sm
```

For negative label

```
Count of usernames in negative label: 803
['@sokendrakouture', '@flyingbolt', '@digitallearnin', '@Luke', '@buckhollywood', '@alix_says', '@mykiaisosm', '@Sally_That_Girl', '@marginatas
naily', '@NewerDeal', '@meggles89', '@ferrite', '@karon', '@IngaDurgin', '@OfficialAS', '@The_Gov', '@uyennguyen_', '@Peace_P', '@markvanbaal
e', '@Fashionsourcing', '@alexispratsides', '@johnny_trouble', '@omerrr', '@BabyBree96', '@David_Henrie', '@ChrisCavs', '@pinkkpaiige12', '@Bre
ttKloesel', '@laurenbotzspons', '@tommcfly', '@dougiemcfly', '@dannymcfly', '@mcflyharry', '@RealKidPoker', '@rayamartin', '@Affan', '@chriswor
thy', '@kevridesthecab', '@allergist', '@johnpapa', '@writeplayrepeat', '@crunchpow', '@HautePersian', '@MrBenRubery', '@pickassoreborn', '@ste
phenfry', '@musewire', '@rosaliiinda', '@courtney319', '@Mr_Marty', '@gavin8', '@DelbertShoopman', '@pirrofina', '@ShannynB', '@cultureshockma
g', '@colinmunroe', '@waledc', '@MatchesMalone', '@dianalogs', '@miiikeo', '@MaryLandrum', '@tonycarrera', '@jonasbrothers', '@shamara99', '@ki
kuuu', '@mcflymusic', '@KatherineLunt', '@nicmoneymil', '@kadi707', '@lizlove_', '@thatchickleelee', '@prinzita', '@maddow', '@mfgreer', '@Jenn
ybeeean', '@fabuleuxdestin', '@mishok13', '@tashadhanraj', '@KING617', '@thatiusstatic', '@jelly1996', '@thekeenanator', '@MsUndrstood', '@Zomb
ie_Claire', '@annzoo', '@kidghost_', '@legendaryswag', '@keithnolan', '@chopsuey2e', '@anai_mrsarm', '@predschickidee', '@ChelRo', '@aafreen',
```

e. count and list all the urls.

This is the code we have used for counting and listing all the urls: -

```
urls_positive = []

for reaction in positive_reactions:
    #u = re.findall('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+', reaction)
    u = re.findall(r'https?://\S+|www\.\S+', reaction)

    if len(u) != 0:
        urls_positive.extend(u)

print("Count of usernames in positive label: {}".format(len(urls_positive)))
print(urls_positive)

urls_negative = []

for reaction in negative_reactions:
    u = re.findall('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+', reaction)
    if len(u) != 0:
        urls_negative.extend(u)

print("Count of usernames in negative label: {}".format(len(urls_negative)))
print(urls_negative)
```

Output: - For positive label

```
Count of usernames in positive label: 136
['http://blip.fm/~4lfcc', 'http://bit.ly/rwoHR', 'http://su.pr/1rXuPY', 'http://twitpic.com/6b03x', 'http://tinyurl.com/dk5p9
4)', 'http://leo.lobato.org/Blipster/', 'http://bit.ly/nZZQV', 'http://bit.ly/etD3a', 'http://dontkillspike.proboards.com/', 'h
ttp://fuzz-ball.com/twitter', 'http://twendz.com', 'http://twitpic.com/66zex', 'http://twitpic.com/6fs89', 'http://www.myspace.
com/xautomaticgirlx', 'www.musiqtone.com', 'http://plurk.com/p/z0xer', 'http://blip.fm/~4kokb', 'http://tinyurl.com/pboph6', 'h
ttp://is.gd/QAaz', 'www.paramore.net/shows', 'http://twitpic.com/680rp', 'http://tinyurl.com/mtq5u2', 'http://twitpic.com/6ddo
x', 'www.m2e.asia', 'http://ustre.am/2txz)', 'http://bit.ly/H6RNb', 'http://yfrog.com/0fvoqj', 'http://plurk.com/p/xfn8o', 'htt
p://bnup2.com/p/569236', 'http://myloc.me/21Sd', 'http://twitpic.com/6dty8', 'http://bit.ly/Jsxm', 'http://tr.im/n0Cy', 'htt
p://tweet.sg', 'http://plurk.com/p/xt8cm', 'http://bit.ly/2mFB2', 'http://myloc.me/41Kq', 'http://plurk.com/p/wy2m1', 'http://t
inyurl.com/qx38my', 'http://tinyurl.com/le9d4f', 'http://bit.ly/9ougy', 'www.myspace.com/tynishakeli', 'http://twitpic.com/6iyt
x', 'http://bit.ly/W6fxT', 'www.iamsoannoyed.com', 'http://ur1.ca/5b4m', 'http://twitpic.com/7h3dq', 'http://snurl.com/feo4p',
'http://twitpic.com/6a28u', 'http://twitpic.com/6iyah', 'http://bit.ly/IQPPD', 'http://nightmoves.me', 'http://bit.ly/Bfy9B',
'http://foamslidefactory.blogspot.com/', 'http://tinyurl.com/cp5yhr', 'http://tinyurl.com/n7wk2x', 'http://twurl.nl/8tb0wf', 'h
ttp://bit.ly/5JFuk', 'http://flickr.com/labelsphotography!', 'http://bit.ly/WdBZk', 'http://twitpic.com/5esg1', 'http://bit.ly/
DmvFY', 'http://twitpic.com/6hs1b', 'http://twitpic.com/5cqrn', 'http://blip.fm/~7rhzx', 'http://is.gd/JKhP', 'http://blip.fm/~
60p98', 'http://www.moteldemoka.com/', 'http://bit.ly/ozTuO', 'http://bit.ly/15yZMq', 'http://overheardinwow.wordpress.com/',
'www.m2e.asia', 'http://bittenbybooks.com/?p=8041', 'http://www.spiritisup.com/ahuginyourinboxgmb.html', 'www.myspace.com/mdadi
nosaur', 'http://twitpic.com/6p6ae', 'http://twitpic.com/6v7qi', 'http://blip.fm/~7at6t', 'http://twitpic.com/7cnge', 'http://b
lip.fm/~68tu1', 'http://bit.ly/ZsnZf', 'http://tinyurl.com/dkpvt7', 'http://bit.ly/19UgRP', 'http://plurk.com/p/xf7i8', 'htt
p://www.carolinamusicawards.com/', 'http://twitpic.com/6p0rw', 'http://www.facebook.com/banpei', 'http://twitpic.com/3kygb', 'h
ttp://emonky.deviantart.com', 'www.flickr.com/emonky', 'http://tinyurl.com/djgyf7', 'http://tinyurl.com/25giveaway', 'http://bi
t.ly/blizzconticket09', 'http://bit.ly/14xvke', 'http://twitpic.com/6d2kp', 'www.tweeterfollow.com', 'http://www.myspace.com/of
machinestheband', 'http://blip.fm/~4mwmi', 'http://twitpic.com/2t5nz', 'http://twitpic.com/69ffo', 'http://tr.im/m4Hs', 'htt
p://myloc.me/29e6', 'http://twitpic.com/6om7s', 'http://is.gd/RmKw', 'www.tweeteradder.com', 'http://tinyurl.com/ry9wap', 'htt
p://blip.fm/~7nyu0', 'http://bit.ly/BPP4d', 'http://bit.ly/gtEB', 'http://tweet.sg', 'http://in.groups.yahoo.com/group/TheComic
sClub/', 'http://plurk.com/p/xdjio', 'http://twitpic.com/66580', 'http://twitpic.com/6g0m7', 'http://www.modelhomeproject.co
m/', 'http://blip.fm/~6st7k', 'http://bit.ly/7l9s1', 'http://bit.ly/ZCYEE', 'http://bit.ly/dB3Tx', 'http://bit.ly/13Jtir', 'htt
p://bit.ly/RElFH', 'http://bit.ly/FVFwq', 'http://tinyurl.com/mrp4x6', 'http://plurk.com/p/110kmy', 'http://ustre.am/3pO8)', 'w
ww.disneycollegeprogram.com', 'http://twitpic.com/6onox', 'http://twitpic.com/69ey9', 'http://bit.ly/7rL9S', 'http://pau27figur
eskater.multiply.com', 'http://blip.fm/~7rfdl', 'http://tinyurl.com/cs73el', 'http://mypict.me/QR7', 'www.stringbeancoffeeshop.
com', 'http://blip.fm/~7aytk', 'http://www.hakkastudy.in.th/)']
```

For negative label: -

```
Count of usernames in negative label: 58
['http://bit.ly/AEbs3', 'http://twitpic.com/3l589', 'http://bit.ly/n4wL4', 'http://twitpic.com/4ijt4', 'http://tinyurl.com/ncbmmo', 'http://twitpic.com/6u8ht', 'http://bit.ly/47etHn', 'http://bit.ly/i9lsr', 'http://twitpic.com/5exx2', 'http://mypict.me/2dG2', 'http://twitpic.com/54r0g', 'http://apps.facebook.com/dogbook/profile/view/6391349', 'http://ustre.am/2FUW)', 'http://twitpic.com/6h6aw', 'http://bit.ly/1G5txF', 'http://tinyurl.com/nsfan3', 'http://plurk.com/p/sy92g', 'http://twitpic.com/6uohm,', 'http://twitpic.com/54r6e', 'http://twitpic.com/5ddsi', 'http://bit.ly/icbfj', 'http://twitpic.com/7lqs3', 'http://myloc.me/teY', 'http://myloc.me/4rxt', 'http://www.dryjuly.com/', 'http://plurk.com/p/z3z6p', 'http://is.gd/16lr2', 'http://twitpic.com/6fsl4', 'http://twitpic.com/8cin5', 'http://twitpic.com/7rwwa', 'http://bit.ly/16Z4xZ', 'http://twitpic.com/4gzhk', 'http://tinyurl.com/pemwuh', 'http://plurk.com/p/xnsr2', 'http://twitpic.com/6tnnx', 'http://twitpic.com/7uis6', 'http://tinyurl.com/c6d3mv', 'http://yfrog.com/0y7wcvj', 'http://twitpic.com/7ogps', 'http://twitpic.com/6bqli', 'http://tinyurl.com/ZeniGeba', 'http://plurk.com/p/rr5fj', 'http://tinyurl.com/ndnb75', 'http://kl.am/Uln', 'http://apps.facebook.com/catbook/profile/view/620328', 'http://plurk.com/p/z2laq', 'http://twitpic.com/6qdq6', 'http://mypict.me/54O0', 'http://plurk.com/p/1100lw', 'http://myloc.me/1XIz', 'http://twitpic.com/7tvhv', 'http://twitpic.com/7h5bf', 'http://twitpic.com/7g6v1', 'http://twitpic.com/7mt6v', 'http://plurk.com/p/rjw8t', 'http://tinyurl.com/ku9yks', 'http://twitpic.com/80023', 'http://tr.im/nqmj']
```

f. count the number of tweets for each day of the week. Eg Mon: 58, Tues: 20, Wed...
This is the code we have used for counting the tweets for each of the day of the week for positive tweets: -

```
date_positive = df_positive['DATE_TIME'].to_list()
counts_positive = {'Mon': 0, 'Tue': 0, 'Wed' : 0, 'Thu': 0, 'Fri': 0, 'Sat': 0, 'Sun': 0}

for date in date_positive:
    for key in counts_positive.keys():
        if re.search(key, date):
            counts_positive[key] = counts_positive[key] + 1

print(counts_positive)
```

```
{'Mon': 481, 'Tue': 132, 'Wed': 172, 'Thu': 50, 'Fri': 391, 'Sat': 298, 'Sun': 763}
```

This is the code we have used for counting the tweets for each of the day of the week for negative tweets: -

```
date_negative = df_negative['DATE_TIME'].to_list()
counts_negative = {'Mon': 0, 'Tue': 0, 'Wed' : 0, 'Thu': 0, 'Fri': 0, 'Sat': 0, 'Sun': 0}

for date in date_negative:
    for key in counts_negative.keys():
        if re.search(key, date):
            counts_negative[key] = counts_negative[key] + 1

print(counts_negative)
```

```
{'Mon': 391, 'Tue': 154, 'Wed': 127, 'Thu': 171, 'Fri': 473, 'Sat': 119, 'Sun': 565}
```

	Positive Class	Negative Class
Sentences	2.3935	2.2405
Tokens	13.3891	14.147
Word with Vowel	7189	6990
Word with consonant	22922	20728
Unique Tokens Before Lowercasing	7852	6417
Unique Tokens After Lowercasing	6819	5620
list of Usernames	1305	803
URL count	136	58

B. You will be given a word x and a class label during the demonstration, and your programme must be able to output the following.

a. total number of occurrences of the given word and sentences containing that word.
b. number of sentences starting with the given word.
c. number of sentences ending with the given word.
Ans: -

```
def printOut(class_label, word):
    total_occurrences = 0
    start_occurrences = 0
    end_occurrences = 0
    if class_label == 0:
        reactions = negative_reactions
    else:
        reactions = positive_reactions
        # Negative class
    for reaction in reactions:
        sentences = re.split(r'[.!?]+', reaction)
        for sentence in sentences:
            occur = re.findall('\w*{}\w*'.format(word), sentence)
            if len(occur) != 0:
                print(sentence)
                total_occurrences = total_occurrences + len(occur)
            start_occur = re.search('^{}'.format(word), sentence)
            if start_occur is not None:
                start_occurrences = start_occurrences + 1
            end_occur = re.search('{}$'.format(word), sentence)
            if end_occur is not None:
                end_occurrences = end_occurrences + 1
    print("Total Occurrences = {}, Starting Occurrences = {}, Ending Occurrences = {}".format(total_occurrences,
                                                                                              start_occurrences, end_occurrences))

printOut(0, 'this')
```

Output: -

```
this whole iphoto face recodnition doesnt work as advertized
dropped her camera outside last nite and didnt know it was missin till this mornin
com/7tvhv - this is what caden made for his daddy
i blew up this balloon that tasted and smelt like burnt rubber and now i have a fricken headache
ok have to accept  before the storm - @mileycyrus and @jonasbrothers (8)  this a bit stupid
 no star trek this weekend
What I'm gonna do  life is not good:'( no more Exit in this hallway I'm stuck in my world
 no fireworks and no electrical parade this sucks now I'm going to dca
ok soo im scared  ewwwwy this movies insssane
@steveray feel so bad I missed musicmatch breakfast this AM
 Hope there aren't any more unpleasant surprises from this ordeal
rohit could not finish it off this time  #ilp
@ladykillerr   janeeeyy, ive got this lama figurine and it sorta looks like a camel baha so it reminded me of you lolll and i miss
@fbb420 not competing this time  im working the tight curves booth
com/80023 - Im going to miss this girl like you have no idea
im/nqmj  I'm sure I'm not the first to make this joke
Total Occurrences = 93, Starting Occurrences = 5, Ending Occurrences = 1
```

Q2. Text Pre-Processing

Sol:

Steps:

Functions have been made for each processing step and have been applied to the column.

1. **Lower Casing**

   This step is done to convert all the words to lower case for better pre processing for further steps. (additional step added)

   Example:

   Class 0

About to get threaded and scared

about to get threaded and scared

Class 1

@awaisnaseer I like Shezan Mangooo too!!! I ha...

@awaisnaseer i like shezan mangooo too!!! i ha...

NOTE : The cleaning of HTML tags and URL's is done before as if we remove the punctuations, remove whitespaces and perform other pre processing steps first the, the symbols in urls's are destroyed and hence it will not be able to find it.

2. **Cleaning the html tags (using regular expression)**
   Normal Html tags like and other have been taken care of along with special html tags like &lt &gt etc.
   Class 0

   stephanie pratt tells mtv news "the hills' did not make me bulimic" http://bit.ly/n4wl4 better but still > half th
   e story is missing
   stephanie pratt tells mtv news "the hills' did not make me bulimic" http://bit.ly/n4wl4 better but still half the story is mi
   ssing

   Class 1

   worked on my car after work. showering then going to bed. sooooooooooo tired. sparrow signing out <cowboy up>
   worked on my car after work. showering then going to bed. sooooooooooo tired. sparrow signing out cowboy up

3. **Cleaning the URL's (using regular expression)**
   https?://\S+|www\.\S+
   Class 0

   http://bit.ly/AEbs3 I can only be sad. #iranelection
   i can only be sad. #iranelection

   Class 1

   @luvyaa you can get coffee @ my family's coffee shop, we have a tea room too. www.stringbeancoffeeshop.com
   @luvyaa you can get coffee @ my family's coffee shop, we have a tea room too.

4. **Remove punctuations**

   Class 0

   http://bit.ly/AEbs3 I can only be sad. #iranelection
   i can only be sad iranelection

   Class 1

   @awaisnaseer I like Shezan Mangooo too!!! I ha...

   awaisnaseer i like shezan mangooo too i had on

5. **Remove extra whitespaces (using regular expression)**
   ' +' remove multiple whitespaces

r"^\s+|\s+$",remove white spaces at start and end

Class 0

```
    i can only be sad  iranelection
i can only be sad iranelection
```

Class 1

```
qandq my performances on my clep tests  qshock
qandq my performances on my clep tests qshock
```

6. **Remove stopwords(using regular expression)**

Class 0

```
i can only be sad iranelection
sad iranelection
```

Class 1

```
awaisnaseer i like shezan mangooo too i had on

awaisnaseer like shezan mangooo one yesterday
```

7. **Spelling Correction**

For spelling correction autocorrect Speller has been used as it produced result faster than textblob and other.

Class 0

```
minecart ride sarahs still afraid ride anything fun
minecraft ride sarah still afraid ride anything fun
```

Class 1

```
@awaisnaseer I like Shezan Mangooo too!!! I had one yesterday
awaisnaseer like sedan mango one yesterday
```

8. **Tokenization**

Class 0

```
['get', 'threaded', 'scared']
```

Class 1

```
['bathroom', 'sparkling', 'btw', 'since', '7', 'thats', 'got', 'ta', 'wrong', 'right', 'next']
```

9. **Lemmatization**

Class 0

```
get threaded scared
get thread scar
```

Class 1

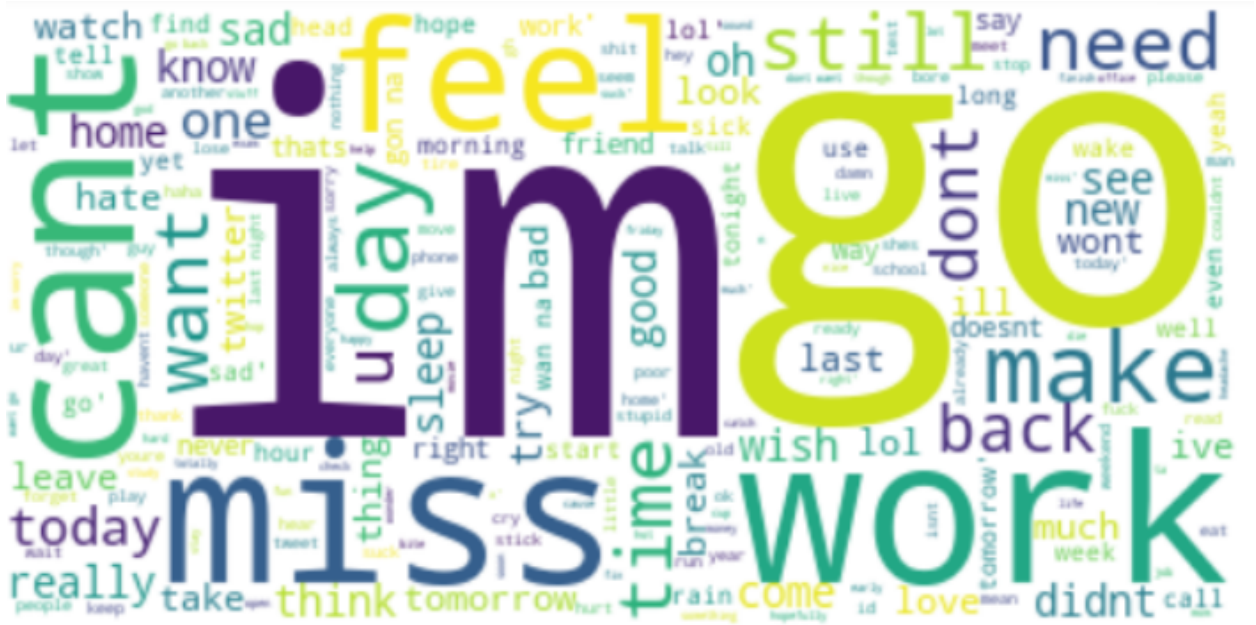
```
bathroom sparkling btw since 7 thats gotta wrong right next
bathroom sparkle btw since 7 thats get ta wrong right next
```

Q3. Visualisation

a. Word Cloud

Made from pre processed data.

For Negative Class Label 0



For Positive Class Label 1

The comparisons between the two word clouds can be done by checking the frequency of the words in both the clouds respectively.

b. Comparison

Negative Class 0

```
Word: get, count: 258
Word: im, count: 207
Word: go, count: 207
Word: work, count: 98
Word: dont, count: 96
Word: like, count: 95
Word: want, count: 95
Word: feel, count: 92
Word: cant, count: 85
Word: miss, count: 83
```

Positive Class 1

```
Word: get, count: 189
Word: im, count: 183
Word: go, count: 173
Word: good, count: 147
Word: thank, count: 126
Word: love, count: 122
Word: like, count: 91
Word: day, count: 89
Word: see, count: 87
Word: new, count: 73
```

Observations:

1. There seems to be very less difference in the two word clouds formed from the respective classes. Through the word frequencies of both the clouds we can see that the most common word are same.
2. The negative class 0 have words like dont, cant with higher frequencies considered to be in a negative sense along with other words like sad, gloomy with are in lesser frequencies.
3. The positive class contains words like, good, thank, love, like with high frequencies showing its sense is good.

So. the positive class seems to be showing its sense but the negative does not show a lot.

Q4.RULE-BASED SENTIMENT ANALYSIS

Sol:

Using VADER (in-built package) retrieve a class label for every instance:

Methodology:

1. SentimentIntensityAnalyzer() has been used directly from Vader.
2. The polarity of the text has been found for raw and pre-processed text both.
3. Vader takes string and returns a dictionary of scores with four categories, negative, positive, neutral and compound (computed by normalizing positive, negative and neutral scores).
4. We have appended these scores and based on the compound score found, if the score is greater than 0 is is positive else negative.

The scores are generated by Vader and then we calculated the compound score.
Polarity column is the same as LABEL column.
Data['comp_score'] column contains the labels generated by prediction of vader.
comp_score and polarity have been used for calculating accuracies.

a)

i. for preprocessed text (obtained in part II)

For the pre-processed part the accuracy reached is

```
66.27011896431071
```

Class 0

	LABEL	DATE_TIME	TEXT	scores	compound	comp_score	polarity
0	0	Fri Jun 05 14:26:50 2009	About to get threaded and scared	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	0.0000	pos	neg
9	0	Wed Jun 17 09:18:19 2009	Needs to shake this gloomy feeling!! Maybe ...	{'neg': 0.452, 'neu': 0.548, 'pos': 0.0, 'comp...	-0.3182	neg	neg
10	0	Mon Jun 22 13:51:56 2009	Minecart ride now. Sarah's still too afraid to...	{'neg': 0.0, 'neu': 0.68, 'pos': 0.32, 'compou...	0.5106	pos	neg

Class 1

	LABEL	DATE_TIME	TEXT	scores	compound	comp_score	polarity
1	1	Thu May 14 10:13:55 2009	@awaisnaseer I like Shezan Mangooo too!!! I ha...	{'neg': 0.0, 'neu': 0.667, 'pos': 0.333, 'comp...	0.3612	pos	pos
2	1	Fri Jun 05 21:02:20 2009	worked on my car after work. showering then go...	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	0.0000	pos	pos
3	1	Sun Jun 14 22:25:52 2009	@Marama Actually we start this afternoon! I w...	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	0.0000	pos	pos

ii. for raw text

For the raw data the accuracy reached is

```
68.34616281782132
```

Class 0

	LABEL	DATE_TIME	TEXT	scores	compound	comp_score	polarity
0	0	Fri Jun 05 14:26:50 2009	About to get threaded and scared	{'neg': 0.367, 'neu': 0.633, 'pos': 0.0, 'comp...	-0.4404	neg	neg
9	0	Wed Jun 17 09:18:19 2009	Needs to shake this gloomy feeling!! Maybe ...	{'neg': 0.35, 'neu': 0.535, 'pos': 0.115, 'com...	-0.4721	neg	neg
10	0	Mon Jun 22 13:51:56 2009	Minecart ride now. Sarah's still too afraid to...	{'neg': 0.0, 'neu': 0.784, 'pos': 0.216, 'comp...	0.5106	pos	neg

Class 1

	LABEL	DATE_TIME	TEXT	scores	compound	comp_score	polarity
1	1	Thu May 14 10:13:55 2009	@awaisnaseer I like Shezan Mangooo too!!! I ha...	{'neg': 0.0, 'neu': 0.675, 'pos': 0.325, 'comp...	0.5229	pos	pos
2	1	Fri Jun 05 21:02:20 2009	worked on my car after work. showering then go...	{'neg': 0.146, 'neu': 0.854, 'pos': 0.0, 'comp...	-0.4404	neg	pos
3	1	Sun Jun 14 22:25:52 2009	@Marama Actually we start this afternoon! I w...	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	0.0000	pos	pos

Observations

1. We can see from the above that for the same text the values of compound changes before and after the pre processing steps.

b) Accuracy is calculated as correct classifications/all classifications.

Methodology:

1. Here the original labels were defined in the LABEL column of the dataset which we mapped to negative for 0 and positive for 1.
2. The values calculated by the vader sentiment analyzer were stored in comp_score column of the dataset and were mapped to compound score >0 positive else negative. (this is the predicted value)
3. These two columns were used to calculate the actual accuracy.

Code:

```
def accuracy_metric(actual, predicted):
    correct = 0
    for i in range(len(actual)):
        if actual[i] == predicted[i]:
            correct += 1
    return correct / float(len(actual)) * 100.0

# Test accuracy
actual = data['polarity']
predicted = data['comp_score']
accuracy = accuracy_metric(actual, predicted)
print(accuracy)
```

Observations

1. The accuracy of raw data comes out to be more as when we pre-process the data, in the stemming part the meaning of the word might change, other processing steps may have an impact on the sentiment of the sentence hence the accuracy comes to be a bit less.

Members contribution

Drishya Uniyal (MT21119) – Q2 and Q4
Harshit Gupta(MT21028) – Q1 and Q3
The members did the above mentioned code and made a report for the same.