Analysis

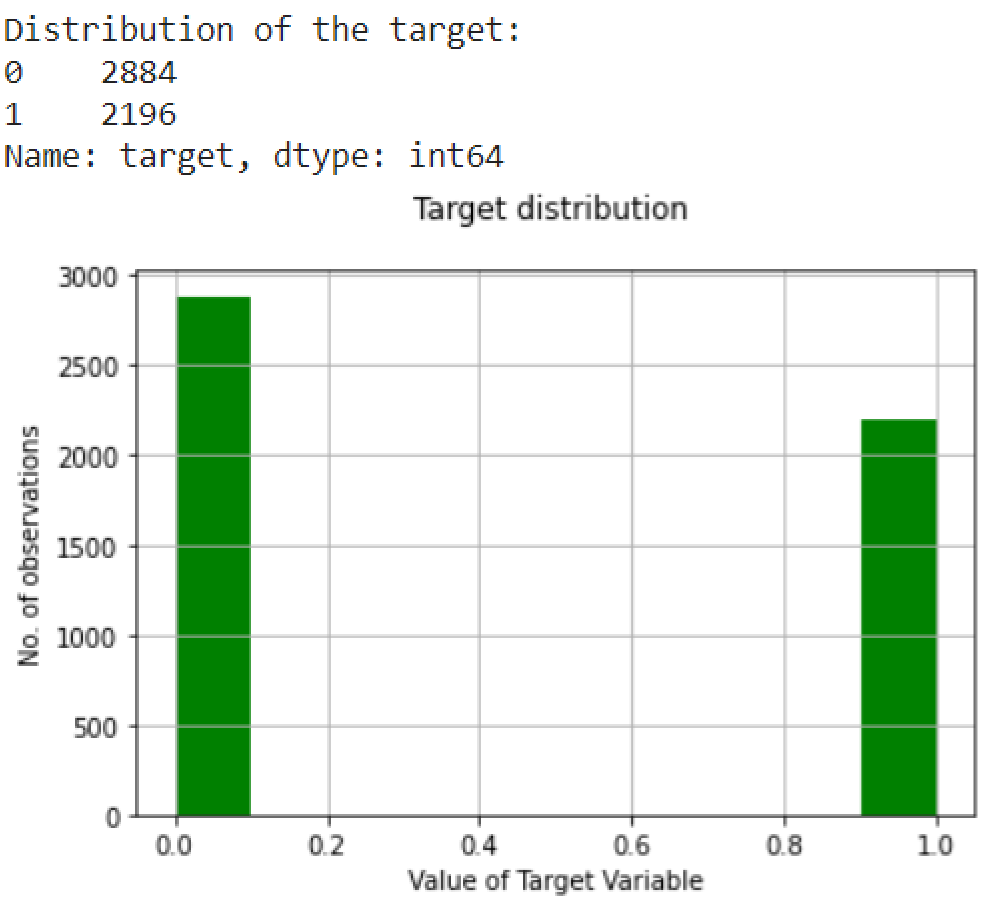
Dataset analysis

The dataset contains target classes of real and fake labeled zero and one with the corresponding tweets.

Our task includes using these tweets over natural language processing to get the corresponding analysis.

Preprocessing and Visualization

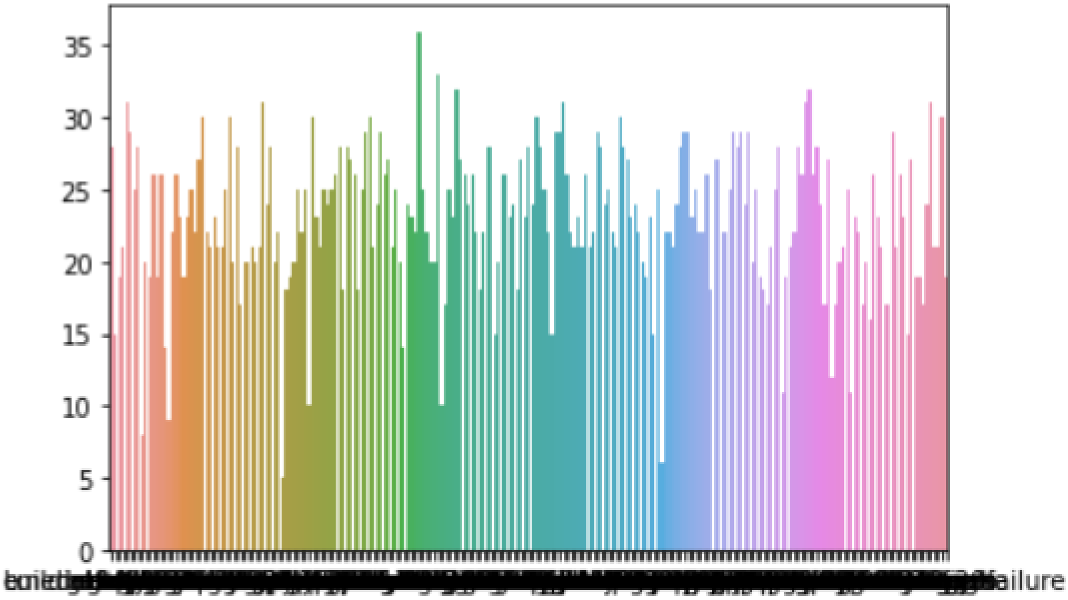
Count of each target variable is being visualized through this figure.



```
Distribution of the target:
0    2884
1    2196
Name: target, dtype: int64
```

Target distribution

After this we plotted the count of each keyword.

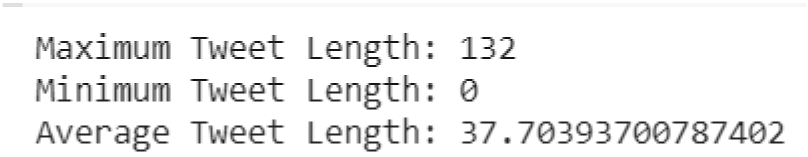


Steps involved in Preprocessing

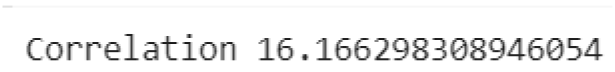
1. Removing the null values.
2. Removing Double Spaces, Hyphens and arrows, Emojis, URL, another Non-English or special symbol
3. Replacing wrong spellings with correct ones.
4. Removing all columns except text and target.
5. Splitting the dataset into train and test sets.

The length of tweets for the false target is always greater than that of the real one . We can draw a conclusion that we will be getting more unique words of tweets for treal targets hence there might be a greater prior probability for the same real target.

We also analysed the the correlation between the length of tweet and the target



```
Maximum Tweet Length: 132
Minimum Tweet Length: 0
Average Tweet Length: 37.70393700787402
```

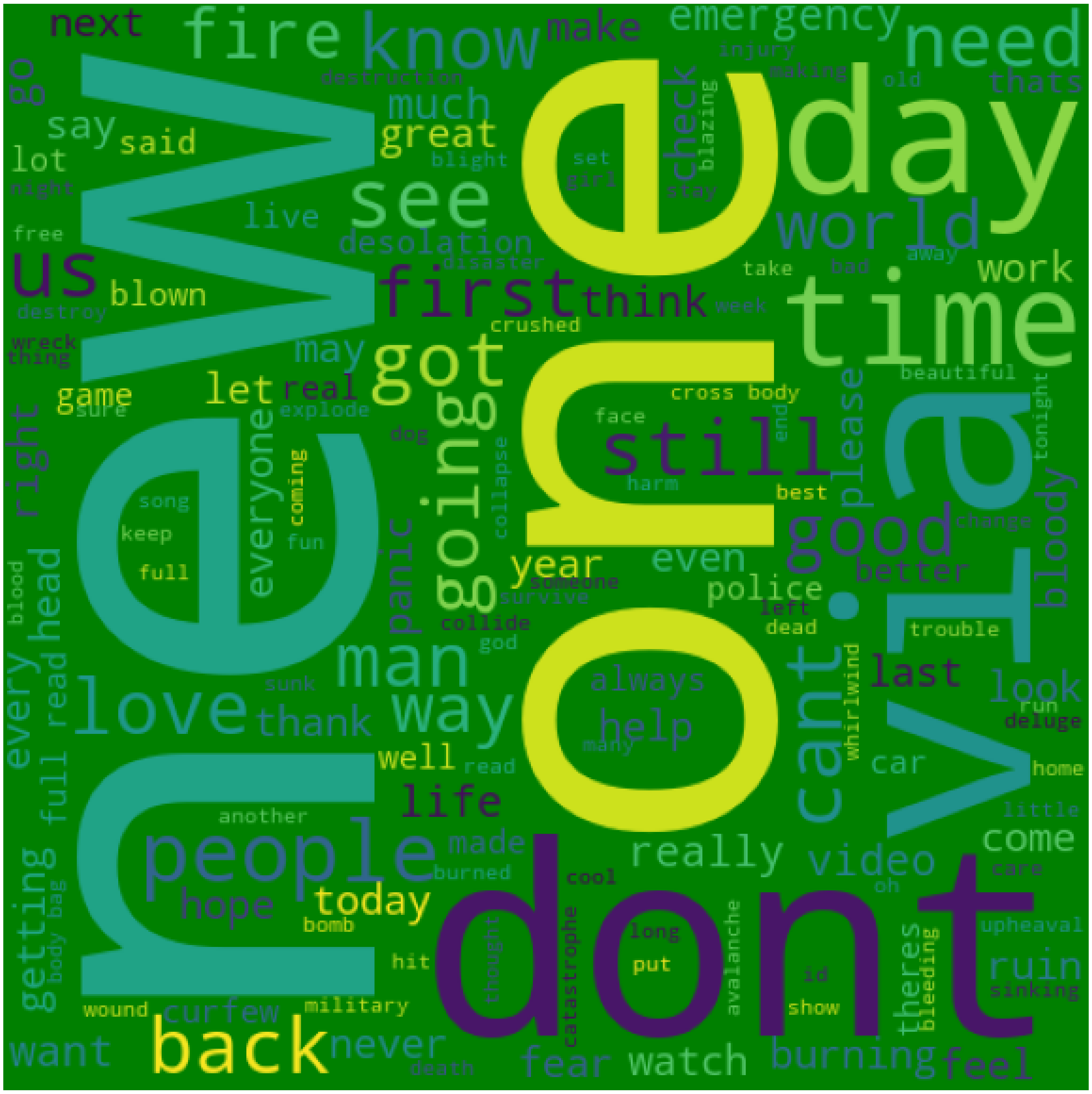


```
Correlation 16.166298308946054
```

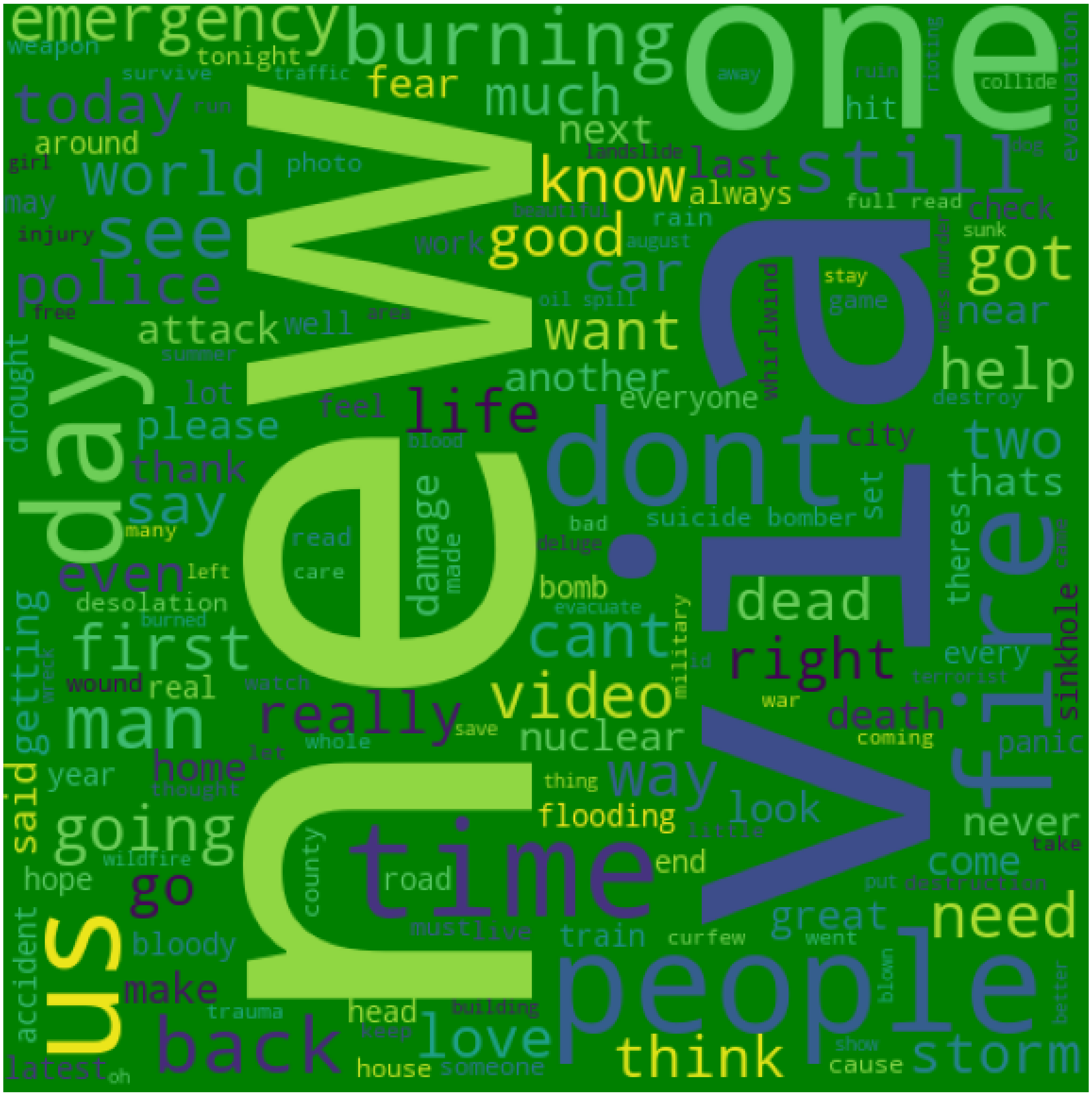
Word Cloud

A word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it’s mentioned within a given text and the more important it is.

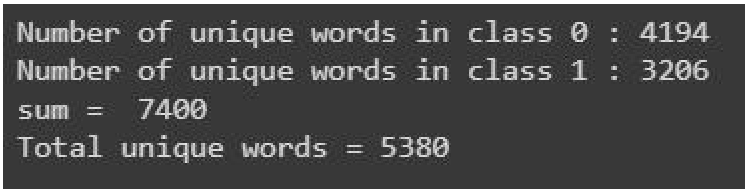
Target 0


new
one
dont
via
day
time
people
back
know
fire
need
first
still
see
got
going
good
cant
love
man
way
world
us
emergency
next
make
much
great
say
said
lot
think
blown
live
desolation
disaster
destruction
injury
making
old
thats
check
blazing
blight
set
girl
stay
night
free
take
bad
away
work
week
destroy
crushed
wreck
thing
may
let
real
game
sure
explode
dog
right
everyone
coming
collapse
please
cross body
face
end
beautiful
tonight
harm
best
song
keep
fun
full
bloody
change
year
someone
survive
even
police
better
left
panic
collide
dead
trouble
blood
head
every
read
full
god
sunk
last
always
look
run
deluge
help
many
thank
well
read
whirlwind
car
home
life
another
little
made
really
video
come
burned
cool
care
oh
getting
bag
body
hope
today
bomb
thought
catastrophe
long
put
avalanche
id
theres
bleeding
upheaval
ruin
sinking
hit
wound
curfew
military
show
want
never
death
fear
watch
burning
feel

Target 1


new
via
one
people
time
dont
fire
day
us
emergency
burning
today
much
fear
tonight
weapon
survive
traffic
away
ruin
rioting
evacuation
collide
hit
run
next
around
girl
world
photo
know
landslide
last
still
always
full read
beautiful
rain
check
may
see
injury
good
august
sunk
got
police
car
stay
oil spill
whirlwind
game
mass murder
near
free
area
attack
well
want
summer
drought
work
lot
another
help
destroy
feel
life
everyone
please
blood
city
two
thank
thats
say
set
suicide bomber
read
bad
theres
many
deluge
care
damage
made
left
even
sinkhole
desolation
burned
bomb
evacuate
dead
getting
wreck
first
cant
military
id
right
every
terrorist
wound
real
watch
video
war
man
really
save
death
nuclear
panic
coming
whole
let
said
year
home
thing
way
thought
look
going
flooding
little
never
take
wildfire
end
come
hope
road
county
put destruction
go
must
live
great
need
train
curfew
went
bloody
accident
blown
better
make
trauma
head
building
keep
love
show
think
cause
storm
latest
oh
house
someone

Does the sum of the unique words in target 0 and 1 sum to the total number of unique words in the whole document? Why or why not? Explain in the report.



```
Number of unique words in class 0 : 4194
Number of unique words in class 1 : 3206
sum =  7400
Total unique words = 5380
```

As you can see, the number of unique words in target 0 + number of unique words in target 1 is more than the total number of unique words in the whole document. It is because some words are common for both targets.

Applying Bayes

$$P(positive|review)\ \alpha\ P(w_1|positive) * P(w_2|positive) * P(w_3|positive) * P(w'|positive) * P(positive)$$

We have calculated the probability of occurrence of the word in a class, we can now substitute the values in the Bayes equation.

After doing the predictions we calculated the True Positive, True Negative, False Positive and False Negative for the dataset which we then used to calculate the precision, recall, accuracy, etc.