Group name: Arjohi
Week 9 deliverables

Group members:

Name: Arda Baris Basaran

Email: ardabarisbasaran@hotmail.com

Country: Turkey

College/Company: Middle East Technical University

Specialization: NLP

Name: Jose Luis Castañeda Terrones

Email: joseluiscastanedat@gmail.com

Country: Perú

College/Company: IFT-UNESP (São Paulo)

Specialization: NLP

Name: Hiten Chadha

Email: hitenchadha1995@gmail.com

Country: Denmark

College/Company: Technical University of Denmark

Specialization: NLP

GitHub repository link:
https://github.com/JoseLuisCastanedaT/DG_common_repo_week7-13

Problem description & Business understanding:

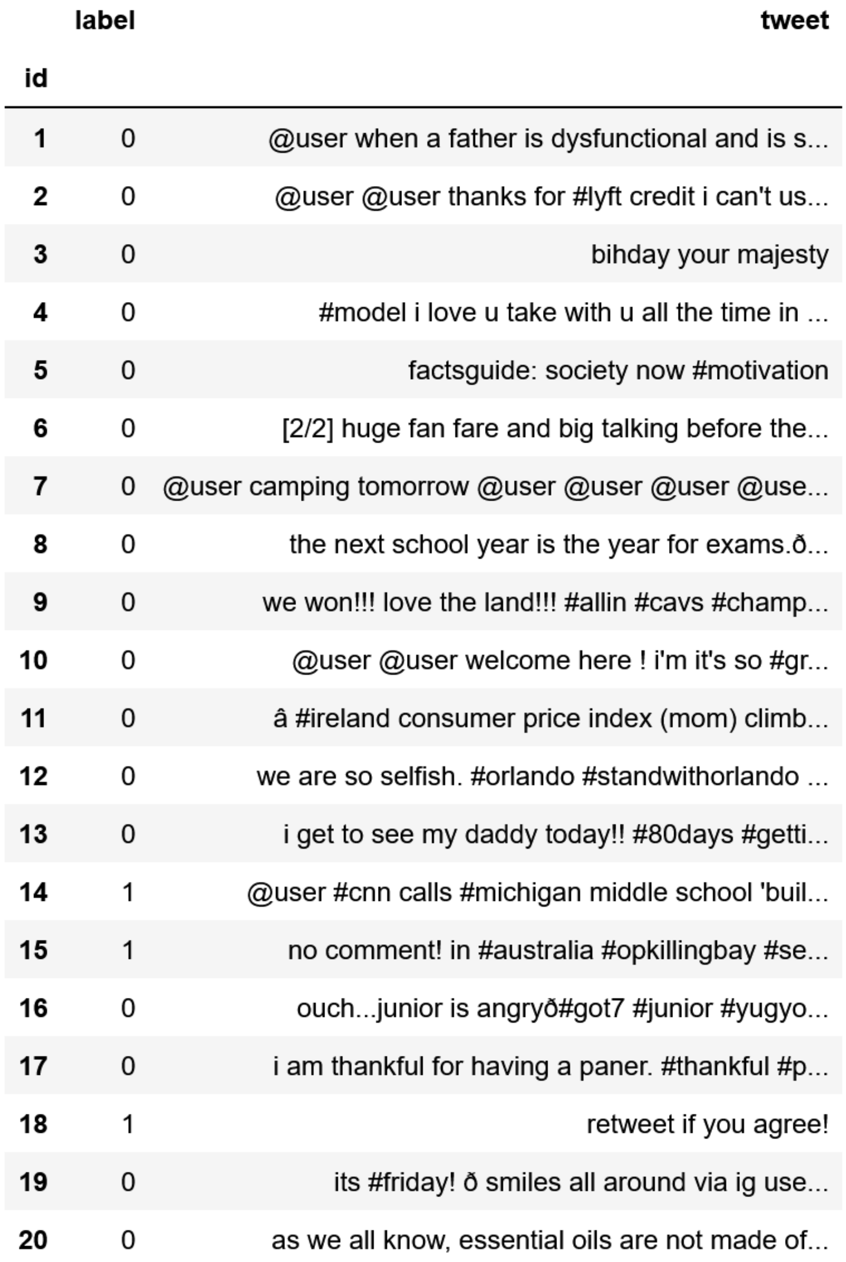
The term hate speech is understood as any type of verbal, written or behavioural communication that attacks or uses derogatory or discriminatory language against a person or group based on what they are, in other words, based on their religion, ethnicity, nationality, race, colour, ancestry, sex or another identity factor. In this problem, We will take you through a hate speech detection model with Machine Learning and Python.

Hate Speech Detection is generally a task of sentiment classification. So for training, a model that can classify hate speech from a certain piece of text can be achieved by training it on a data that is generally used to classify sentiments. So for the task of hate speech detection model, We will use the Twitter tweets to identify tweets containing Hate speech.

We will analyze a dataset CSV file from Kaggle containing 31,935 tweets. The dataset is heavily skewed with 93% of tweets or 29,720 tweets containing non-hate labeled Twitter data and 7% or 2,242 tweets containing hate-labeled Twitter data. We will try different classification algorithms after the preprocessing and data cleaning steps.

Type of data:

- 1 boolean column (1 representing hate speech tweet and 0 non-hate speech tweet)
- 1 string column (the tweet itself)
- 1 numerical column (index column, representing the id)

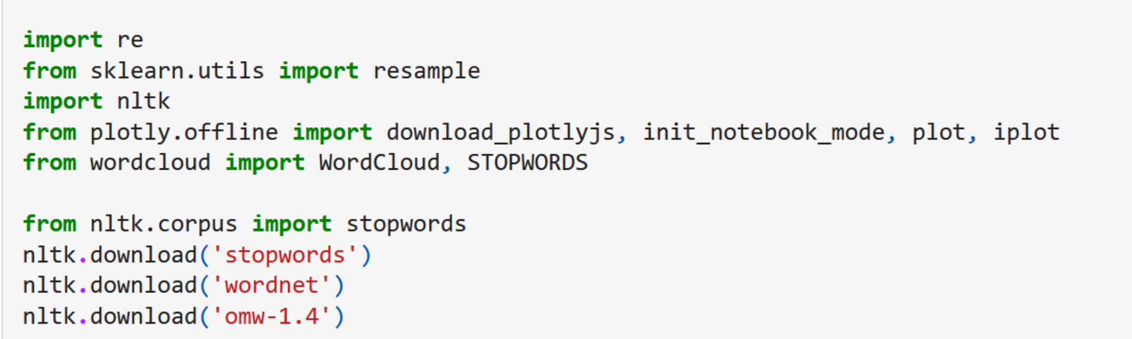


	label	tweet
id		
1	0	@user when a father is dysfunctional and is s...
2	0	@user @user thanks for #lyft credit i can't us...
3	0	bihday your majesty
4	0	#model i love u take with u all the time in ...
5	0	factsguide: society now #motivation
6	0	[2/2] huge fan fare and big talking before the...
7	0	@user camping tomorrow @user @user @user @use...
8	0	the next school year is the year for exams.ð...
9	0	we won!!! love the land!!! #allin #cavs #champ...
10	0	@user @user welcome here ! i'm it's so #gr...
11	0	â #ireland consumer price index (mom) climb...
12	0	we are so selfish. #orlando #standwithorlando ...
13	0	i get to see my daddy today!! #80days #getti...
14	1	@user #cnn calls #michigan middle school 'buil...
15	1	no comment! in #australia #opkillingbay #se...
16	0	ouch...junior is angryð#got7 #junior #yugyo...
17	0	i am thankful for having a paner. #thankful #p...
18	1	retweet if you agree!
19	0	its #friday! ð smiles all around via ig use...
20	0	as we all know, essential oils are not made of...

Arda's approach (best approach):

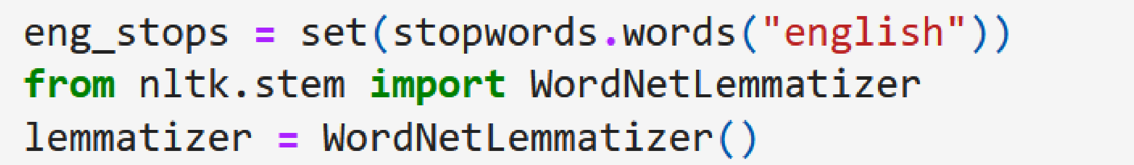
Cleaning:

Importinng neccesary libraries



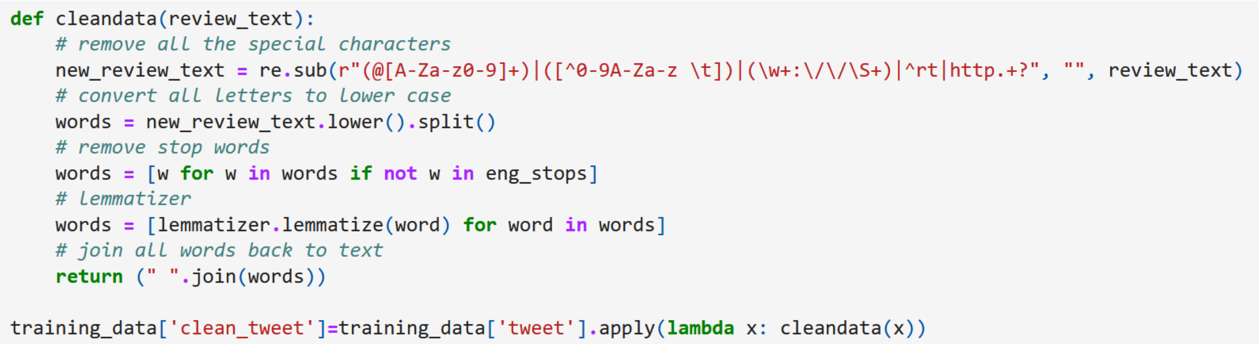
```
import re
from sklearn.utils import resample
import nltk
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
from wordcloud import WordCloud, STOPWORDS

from nltk.corpus import stopwords
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

Storing stopwords and instantiating lemmatizer:



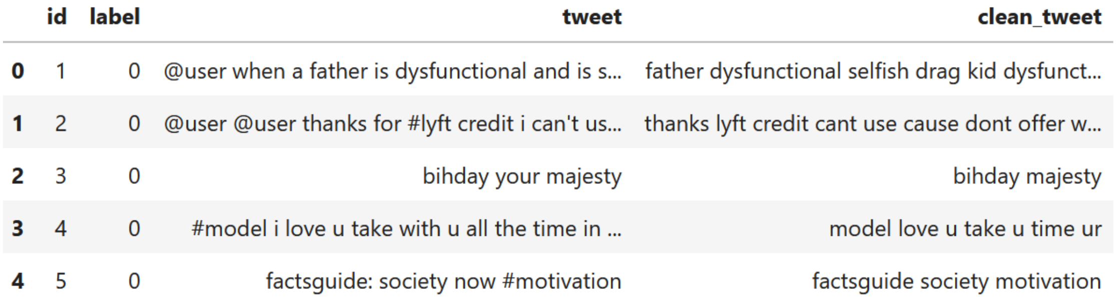
```
eng_stops = set(stopwords.words("english"))
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
```

Pipeline for cleaning data:



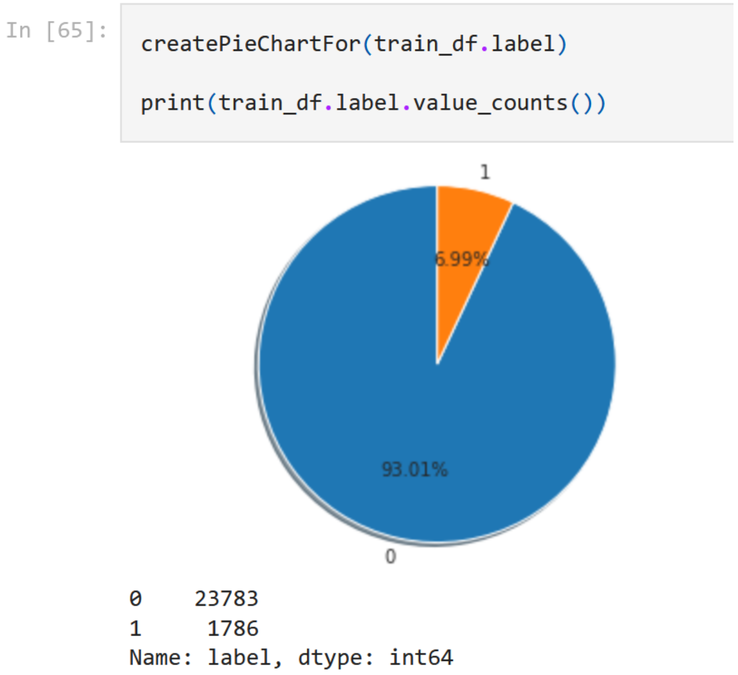
```
def cleandata(review_text):
    # remove all the special characters
    new_review_text = re.sub(r"(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)|^rt|http.+?", "", review_text)
    # convert all letters to lower case
    words = new_review_text.lower().split()
    # remove stop words
    words = [w for w in words if not w in eng_stops]
    # lemmatizer
    words = [lemmatizer.lemmatize(word) for word in words]
    # join all words back to text
    return (" ".join(words))

training_data['clean_tweet']=training_data['tweet'].apply(lambda x: cleandata(x))
```

Clean tweets:



	id	label	tweet	clean_tweet
0	1	0	@user when a father is dysfunctional and is s...	father dysfunctional selfish drag kid dysfunct...
1	2	0	@user @user thanks for #lyft credit i can't us...	thanks lyft credit cant use cause dont offer w...
2	3	0	bihday your majesty	bihday majesty
3	4	0	#model i love u take with u all the time in ...	model love u take u time ur
4	5	0	factsguide: society now #motivation	factsguide society motivation

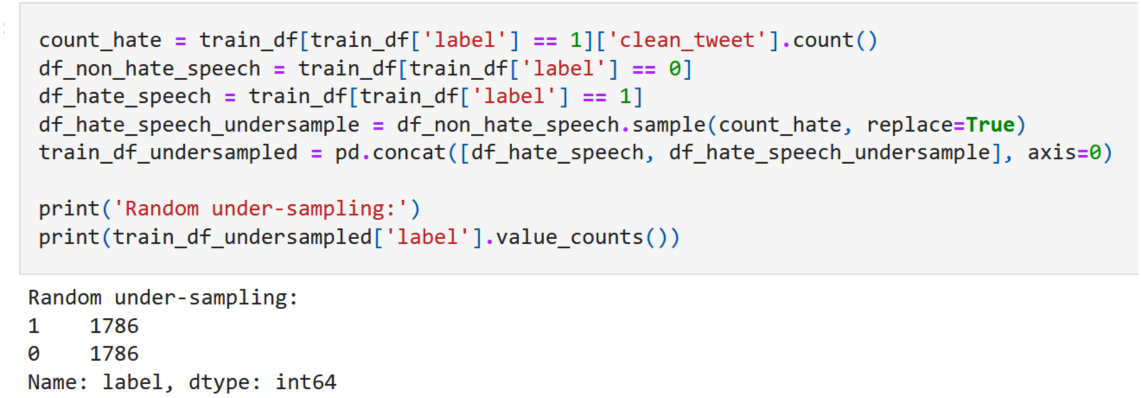
Data imbalance handling:



```
In [65]: createPieChartFor(train_df.label)

         print(train_df.label.value_counts())
```

```
0    23783
1     1786
Name: label, dtype: int64
```

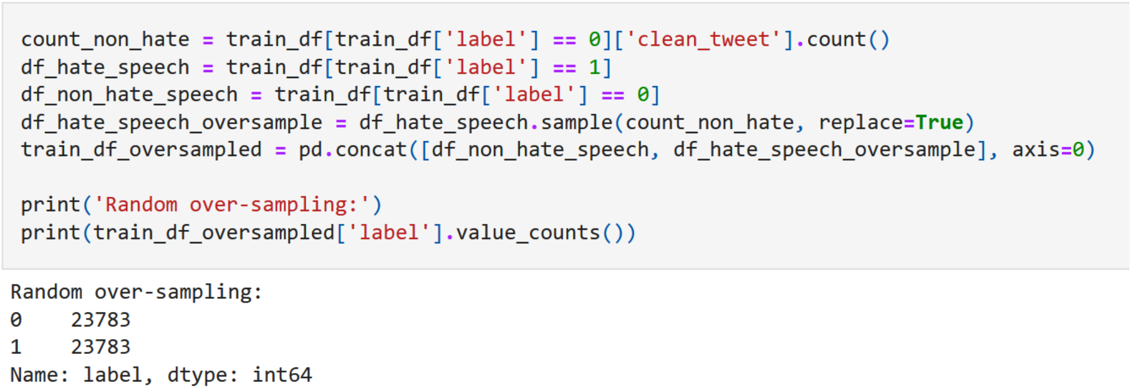
Downsampling:



```
count_hate = train_df[train_df['label'] == 1]['clean_tweet'].count()
df_non_hate_speech = train_df[train_df['label'] == 0]
df_hate_speech = train_df[train_df['label'] == 1]
df_hate_speech_undersample = df_non_hate_speech.sample(count_hate, replace=True)
train_df_undersampled = pd.concat([df_hate_speech, df_hate_speech_undersample], axis=0)

print('Random under-sampling:')
print(train_df_undersampled['label'].value_counts())
```

```
Random under-sampling:
1    1786
0    1786
Name: label, dtype: int64
```

Oversampling:

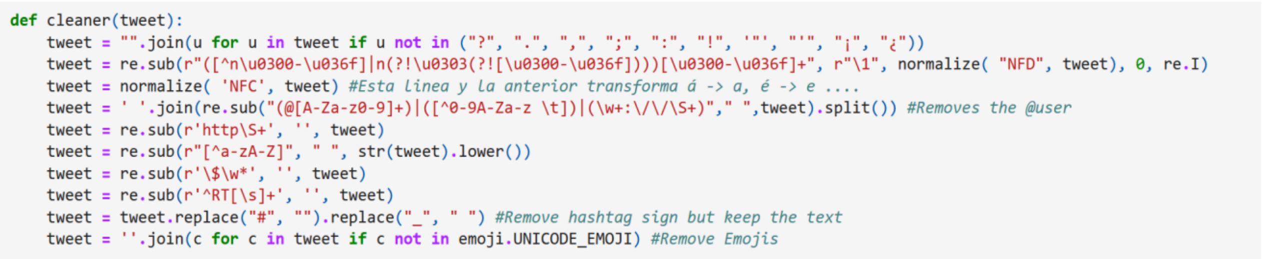


```
count_non_hate = train_df[train_df['label'] == 0]['clean_tweet'].count()
df_hate_speech = train_df[train_df['label'] == 1]
df_non_hate_speech = train_df[train_df['label'] == 0]
df_hate_speech_oversample = df_hate_speech.sample(count_non_hate, replace=True)
train_df_oversampled = pd.concat([df_non_hate_speech, df_hate_speech_oversample], axis=0)

print('Random over-sampling:')
print(train_df_oversampled['label'].value_counts())
```

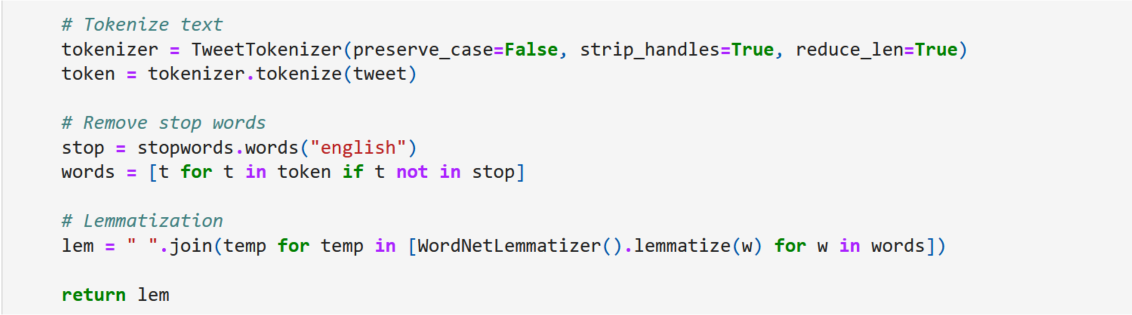
```
Random over-sampling:
0    23783
1    23783
Name: label, dtype: int64
```

Jose’s approach (Alternative for cleaning):

Cleaning:

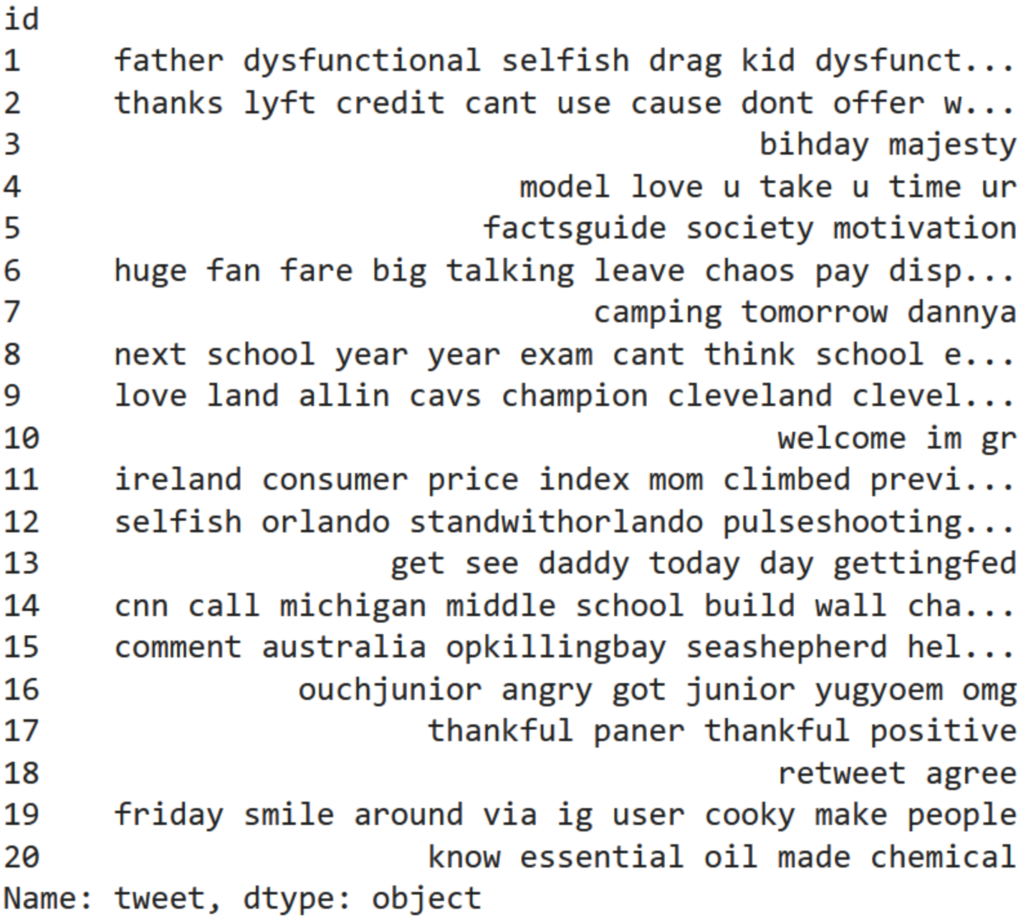


```
def cleaner(tweet):
    tweet = "".join(u for u in tweet if u not in ("?", ".", ",", ";", ":", "!", '"', "'", "¡", "¿"))
    tweet = re.sub(r"([^n\u0300-\u036f]|n(?!\u0303(?![\u0300-\u036f])))[\u0300-\u036f]+", r"\1", normalize( "NFD", tweet), 0, re.I)
    tweet = normalize( 'NFC', tweet) #Esta linea y la anterior transforma á -> a, é -> e ....
    tweet = ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)"," ",tweet).split()) #Removes the @user
    tweet = re.sub(r'http\S+', '', tweet)
    tweet = re.sub(r"[^a-zA-Z]", " ", str(tweet).lower())
    tweet = re.sub(r'\$\w*', '', tweet)
    tweet = re.sub(r'^RT[\s]+', '', tweet)
    tweet = tweet.replace("#", "").replace("_", " ") #Remove hashtag sign but keep the text
    tweet = ''.join(c for c in tweet if c not in emoji.UNICODE_EMOJI) #Remove Emojis
```



```
    # Tokenize text
    tokenizer = TweetTokenizer(preserve_case=False, strip_handles=True, reduce_len=True)
    token = tokenizer.tokenize(tweet)

    # Remove stop words
    stop = stopwords.words("english")
    words = [t for t in token if t not in stop]

    # Lemmatization
    lem = " ".join(temp for temp in [WordNetLemmatizer().lemmatize(w) for w in words])

    return lem
```

Clean tweets:



```
id
1     father dysfunctional selfish drag kid dysfunct...
2     thanks lyft credit cant use cause dont offer w...
3                                        bihday majesty
4                                model love u take u time ur
5                               factsguide society motivation
6     huge fan fare big talking leave chaos pay disp...
7                                    camping tomorrow dannya
8     next school year year exam cant think school e...
9     love land allin cavs champion cleveland clevel...
10                                          welcome im gr
11    ireland consumer price index mom climbed previ...
12    selfish orlando standwithorlando pulseshooting...
13                      get see daddy today day gettingfed
14    cnn call michigan middle school build wall cha...
15    comment australia opkillingbay seashepherd hel...
16               ouchjunior angry got junior yugyoem omg
17                 thankful paner thankful positive
18                                          retweet agree
19    friday smile around via ig user cooky make people
20                     know essential oil made chemical
Name: tweet, dtype: object
```