Sentimental Analysis

Twitter sentiment analysis

imports

pandas for data process

In [20]:

```
import warnings
warnings.filterwarnings("ignore")

import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud

import re
import string
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem.porter import *
```

Data process

train data: https://www.kaggle.com/arkhoshghalb/twitter-sentiment-analysis-hatred-speech

In [11]:

```
data = pd.read_csv('data/train.csv', encoding = "ISO-8859-1", engine='python')
data
```

Out[11]:

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation
...	...	...	...
31957	31958	0	ate @user isz that youuu?Ã°ÂÂÂÃ°ÂÂÂÃ°ÂÂ...
31958	31959	0	to see nina turner on the airwaves trying to...
31959	31960	0	listening to sad songs on a monday morning otw...
31960	31961	1	@user #sikh #temple vandalised in in #calgary,...
31961	31962	0	thank you @user for you follow

31962 rows × 3 columns

most repeated words

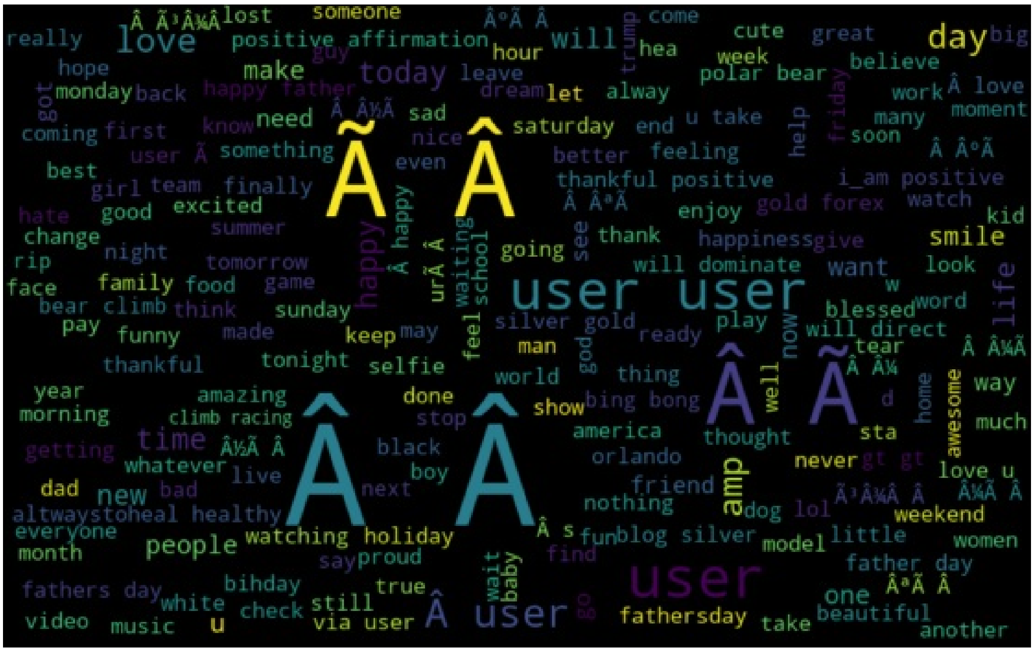
With the help of wordcloud we can find most repeated words easily from the whole dataset. By the image we can come to a conclusion that in our data 'user' is the word which repeated more number of times. And we have some unknown symbols also in our data. So, befor proceeding with modelling, we should first clean the data

some unknown symbols also in our data. So, befor proceeding with modelling, we should first clean the data.

In [28]:

```
sentences = data['tweet'].tolist()
sentences_ss = " ".join(sentences)

plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```



clean tweet text

In [13]:

```
data['cleanTweet'] = data['tweet']
data.head()
```

Out[13]:

	id	label	tweet	cleanTweet
0	1	0	@user when a father is dysfunctional and is s...	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty	bihday your majesty
3	4	0	#model i love u take with u all the time in ...	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation	factsguide: society now #motivation

In [14]:

```
#remove @user
data['cleanTweet'] = data['tweet'].map(lambda x: re.sub('@\S+', ' ', x))
data.head()
```

Out[14]:

	id	label	tweet	cleanTweet
0	1	0	@user when a father is dysfunctional and is s...	when a father is dysfunctional and is so se...
1	2	0	@user @user thanks for #lyft credit i can't us...	thanks for #lyft credit i can't use cause ...
2	3	0	bihday your majesty	bihday your majesty
3	4	0	#model i love u take with u all the time in ...	#model i love u take with u all the time in ...

	id	label	tweet	cleanTweet
4	5	0	factsguide: society now #motivation	factsguide: society now #motivation

In [15]:

```
#upper case to lower case
data['cleanTweet'] = data['cleanTweet'].map(lambda x: x.lower())

#remove number
data['cleanTweet'] = data['cleanTweet'].map(lambda x: re.sub(r'\d+', '', x))

#remove punctuation
data['cleanTweet'] = data['cleanTweet'].map(lambda x: x.translate(x.maketrans('', '', st
ring.punctuation)))

#remove whitespace
data['cleanTweet'] = data['cleanTweet'].map(lambda x: x.strip())

#remove url
url_cleaner = "https?:\S+|http?:\S|[^A-Za-z0-9]+"
data['cleanTweet'] = data['cleanTweet'].map(lambda x: re.sub(url_cleaner, ' ', x))

#removing small words
data['cleanTweet'] = data['cleanTweet'].apply(lambda x: ' '.join([w for w in x.split() i
f len(w)>3]))
data
```

Out[15]:

	id	label	tweet	cleanTweet
0	1	0	@user when a father is dysfunctional and is s...	when father dysfunctional selfish drags kids i...
1	2	0	@user @user thanks for #lyft credit i can't us...	thanks lyft credit cant cause they dont offer ...
2	3	0	bihday your majesty	bihday your majesty
3	4	0	#model i love u take with u all the time in ...	model love take with time
4	5	0	factsguide: society now #motivation	factsguide society motivation
...	...	...	...	...
31957	31958	0	ate @user isz that youuu?Ã°ÂÂÂÃ°ÂÂÂÃ°ÂÂ...	that youuu
31958	31959	0	to see nina turner on the airwaves trying to...	nina turner airwaves trying wrap herself mantl...
31959	31960	0	listening to sad songs on a monday morning otw...	listening songs monday morning work
31960	31961	1	@user #sikh #temple vandalised in in #calgary,...	sikh temple vandalised calgary condemns
31961	31962	0	thank you @user for you follow	thank follow

31962 rows × 4 columns

In [16]:

```
#Tokenization
tokenized_tweet = data['cleanTweet'].apply(lambda x: x.split())
tokenized_tweet.head()
```

Out[16]:

```
0    [when, father, dysfunctional, selfish, drags, ...
1    [thanks, lyft, credit, cant, cause, they, dont...
2                               [bihday, your, majesty]
3                     [model, love, take, with, time]
4                   [factsguide, society, motivation]
Name: cleanTweet, dtype: object
```

In [22]:

```
# stemming
stemmer = PorterStemmer()
tokenized_tweet = tokenized_tweet.apply(lambda x: [stemmer.stem(i) for i in x])
tokenized_tweet
```

Out[22]:

```
0        [when, father, dysfunct, selfish, drag, kid, i...
1        [thank, lyft, credit, cant, cau, they, dont, o...
2                                   [bihday, your, majesti]
3                           [model, love, take, with, time]
4                               [factsguid, societi, motiv]
                                ...
31957                                           [that, youuu]
31958    [nina, turner, airwav, tri, wrap, herself, man...
31959                   [listen, song, monday, morn, work]
31960             [sikh, templ, vandali, calgari, condemn]
31961                                          [thank, follow]
Name: cleanTweet, Length: 31962, dtype: object
```

In [23]:

```
for i in range(len(tokenized_tweet)):
    tokenized_tweet[i] = ' '.join(tokenized_tweet[i])

data['cleanTweet'] = tokenized_tweet
```

Understanding tweets

Now we check cleanTweet

In [26]:

```
all_words = ' '.join([text for text in data['cleanTweet']])
from wordcloud import WordCloud
wordcloud = WordCloud(width=800, height=500, random_state=21, max_font_size=110).generate
(all_words)

plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```