NLP and Text Mining Basics

Venkat Reddy

statinfer

Contents

- What is text mining
- Corpus
- Preparing text data for analysis
  - NLTK package
  - sciPy package
- Pre-processing steps
- WordCloud
- DTM

Text Mining

- **This vacuum cleaner sucks**
- **I never had such pizza before, not sure about future either**
- No action, no drama, no comedy, no romance, just pure horror
- The food was not good, it was bad
  The food was not bad, it was good

What is Text mining?

What is text mining

- Making sense out of text data
- Datamining on text input
- Exploratory data analysis on text data
- Also known as Text Analytics
- What is NLP – Natural Language Processing

Text Data Sources

- Customer Emails
- Customer feedback and reviews
- Blog articles
- Tweets and Facebook posts
- News articles
- Social media comments
- Customer verbatim in a survey
- Scanned documents of the physical forms

Data Preparation for Text Mining

Numerical data is well structured

- rows and columns.
- For every record(row) we have information well organised in the form of columns.
- Each column captures a specific section of information
- Every record has almost all columns available
- Easy to perform mathematical and statistical computations

Text data is unstructured

- Most of the text data has one or two columns
- Whole data is in one column
- Each record might have different length
- Difficult to arrange it as a dataset
- Text data is not very well structured.
- Direct computation on text data not easy

Computers don't Understand Language

- Direct computation on text data not easy.
- Computers don't understand a **Sentence** or a **Word** or any **Underlying Emotion**.
- We need to bring the data structure to a point where computers can convert text data into number, process the numbers, convert those numbers back to Text data.

Two Steps in NLP Model building

Step1=> Convert text data into numerical data

Step2=> Build models on numerical data

Step1

Step2

statinfer

Giving structure to Unstructured text

- Text data is unstructured. We need to add some basic structure.
- How to prepare data for computations.
- How to represent the text?
- There is a lot of pre-processing required before jumping on to analysis

NLP Terminology

Corpus

- Corpus
  - Corpus is collection of text or sometimes text files.
- Document
  - Each document is an entity/observation/ record in corpus.
- You can think corpus as a raw data frame for text data.

NLP Packages

NLTK

- A comprehensive package for NLP tasks
- NLTK has rich documentation and examples
- The package has functions for all text mining tasks like - Reading data, Tokenizing, Stemmers, Taggers, Parsers, WordNet, Evaluation Metrics

SpaCy

- An alternative to NLTK package
- SpaCy is much faster compared to NLTK
- Has few additional options and comparatively easy to use

statinfer

NLTK vs. spaCy – Which package to use?

- Both packages are good.
- Both of them are open source.
- Each one of them have some advantages and disadvantages.
- Better to **use both** of them in our projects.
- Here we will use a mix of both

NLP Packages installation

```
!pip install nltk

!pip install spacy

import pandas as pd
import nltk
import spacy
nltk.download("all")
```

Preparing Data for text mining

RAW Text cleaning data stages

- The raw data need to be cleaned to a great extent
- There are many steps in cleaning the data
  - Tokenizing
  - Stop word removal
  - Removing webpage links
  - Removing Punctuation
  - Lemmatising or root form
- The final accuracy largely depends on quality of input data. The data preparation takes more time than final analysis

Preparing Data for text mining

- There are some common words in every document. They might not have any meaning
  - a, an, the, this, is, was, that, are, you, we
- Few words in the document have same root but used in different ways
  - Buying, bought, buy
- Few documents have numbers and datetime values
- Remove html links, e-mail addresses, image file links from text
- Special characters and punctuation
- Upper case and lower case
- Spelling mistakes correction

Data Importing

- Import twitter data. How many rows and columns are there?

	tweet_ID	raw_tweet	sentiment_label
23115	dc5ee645fb	Why can`t you? Are you rooting for Ali in the...	neutral
16873	cfb49dff56	no i mean 2moz. I`m workin` 7-1 in a bakers t...	neutral
1887	b83cfda047	Listening to the new Green Day album Fingers ...	neutral
14335	a99ebc9c3f	just woke up with a cat in my face haha	neutral
22756	8ab49e0790	rentaphone yesy have	neutral
25313	0a512e8e0d	Why am I posting so late? Just got back from s...	positive
4569	ac41ba79fe	im so tired of my responsabilities i wish i wa...	negative
27251	7031fb45da	come back for another year pllleeeaaasse	neutral
4353	5215109d35	make a prank call for me	neutral
27242	48a317e32c	Happy Mother`s Day !	positive

Convert everything to string and lowercase

- All the text documents must be in the format of strings and lower case for further processing

```
twitter_data["tweet_lowcase"]=twitter_data["raw_tweet"].apply(la
mbda x:str(x).lower())
```

Tokenizing

- A token is the technical name for a sequence of characters/words/sentences.
- Each “Entity" that is making a sentence or a paragraph when kept in a sequence would be called token.
- Word Token:
  - Each word is a token when a sentence is "tokenized" into words.
  - Tokenization is based on specific split rule:
  - word_tokenize: split would generally be ‘Space’ along with additional splitting logic

Tokenizing - Code

```
from nltk.tokenize import word_tokenize

twitter_data["word_tokens"] = twitter_data["tweet_lowcase"].apply(
lambda x:word_tokenize(str(x)))

#lambda function to apply on all rows
#str() function to avoid numeric and other errors
print(twitter_data[["raw_tweet","word_tokens"]].sample(10))
```

Tokenizing - Code

```
from nltk.tokenize import word_tokenize
twitter_data["word_tokens"] = twitter_data["raw_tweet"].apply(lambda x:word_tokenize(str(x)))
#lambda function to apply on all rows
#str() function to avoid numeric and other errors
print(twitter_data[["raw_tweet","word_tokens"]].sample(10))
```

	raw_tweet	word_tokens
18303	my car is possessed and won`t stop honking at me	[my, car, is, possessed, and, won`t, stop, hon...
12999	Happy Mother`s Day!!	[Happy, Mother`s, Day, !, !]
10865	he keeps makingfun of my typos!	[he, keeps, makingfun, of, my, typos, !]
1288	Nice! You should submit that to failblog.org	[Nice, !, You, should, submit, that, to, failb...
21111	Missed the hello kitty not enough time oh well	[Missed, the, hello, kitty, not, enough, time,...
8576	Going on 1,116 days still no new tattoo.	[Going, on, 1,116, days, still, no, new, tatto...
7138	morning, still trying to find a babysitter, th...	[morning, ,, still, trying, to, find, a, babys...
4444	It was BuckFast. Brain just went blank	[It, was, BuckFast, ., Brain, just, went, blank]
6661	Ugh worried about my math test	[Ugh, worried, about, my, math, test]
18642	I quite like baseball and bball oh and the odd...	[I, quite, like, baseball, and, bball, oh, and...

Expanding short forms

- "ain`t": "am not",
- "aren`t": "are not",
- "can`t": "cannot",
- "can`t`ve": "cannot have",
- "`cause": "because",
- "could`ve": "could have",
- "couldn`t": "could not",
- "couldn`t`ve": "could not have",
- "didn`t": "did not",
- "doesn`t": "does not",
- "don`t": "do not",
- "hadn`t": "had not",
- "hadn`t`ve": "had not have"

Code - Expanding short forms

```
def expanded_form(x):
  if x in contra_Expan_Dict.keys():
    return(contra_Expan_Dict[x])
  else:
    return(x)

x=str(twitter_data["tweet_lowcase"][6207])
x=x.split()
[expanded_form(t) for t in x]
```

```
original tweet ==>  it`s under a honeymoon by the good life
Expanded form ==> ['it is', 'under', 'a', 'honeymoon', 'by', 'the', 'good', 'life']
```

statinfer

Code - Expanding short forms

```
twitter_data["tweet_expanded"]=twitter_data["tweet_lowcase"].apply
(lambda x:[expanded_form(t) for t in str(x).split()])
```

raw_tweet	tweet_expanded
It should take you back to colorado where yo...	[it, should, take, you, back, to, colorado, wh...
OPS sorry Queen Mom	[ops, sorry, queen, mom]
A singing girl. Talented, yeah. Good to compla...	[a, singing, girl., talented,, yeah., good, to...
Feel like **** today Got a speeding ticket 1...	[feel, like, ****, today, got, a, speeding, ti...
just in imagination	[just, in, imagination]
Don`t know yet Lemme know if you come up wit...	[do not, know, yet, lemme, know, if, you, come...
Clive it`s my birthday pat me http://apps.fac...	[clive, it is, my, birthday, pat, me, http://a...
i think june gloom has arrived	[i, think, june, gloom, has, arrived]

Stop Words

- There are some common words in every document.
- These words are not really informative
- Most of the times they are irrelevant for document representation
  - Eg: a, an, the, this, is, was, for, are
- These words carries importance for humans but for analysis these words doesn’t give any insights.
- It’s better to remove these words form our documents.

Demo: Stop Words - NLTK

Number of Stop words in NLTK ==> 179

```
 ['a', 'about', 'above', 'after', 'again', 'against', 'ain', 'all', 'am', 'an', 'and', 'any',
'are', 'aren', "aren't", 'as', 'at', 'be', 'because', 'been', 'before', 'being', 'below',
'between', 'both', 'but', 'by', 'can', 'couldn', "couldn't", 'd', 'did', 'didn', "didn't",
'do', 'does', 'doesn', "doesn't", 'doing', 'don', "don't", 'down', 'during', 'each', 'few',
'for', 'from', 'further', 'had', 'hadn', "hadn't", 'has', 'hasn', "hasn't", 'have', 'haven',
"haven't", 'having', 'he', 'her', 'here', 'hers', 'herself', 'him', 'himself', 'his', 'how',
'i', 'if', 'in', 'into', 'is', 'isn', "isn't", 'it', "it's", 'its', 'itself', 'just', 'll',
'm', 'ma', 'me', 'mightn', "mightn't", 'more', 'most', 'mustn', "mustn't", 'my', 'myself',
'needn', "needn't", 'no', 'nor', 'not', 'now', 'o', 'of', 'off', 'on', 'once', 'only', 'or',
'other', 'our', 'ours', 'ourselves', 'out', 'over', 'own', 're', 's', 'same', 'shan',
"shan't", 'she', "she's", 'should', "should've", 'shouldn', "shouldn't", 'so', 'some',
'such', 't', 'than', 'that', "that'll", 'the', 'their', 'theirs', 'them', 'themselves',
'then', 'there', 'these', 'they', 'this', 'those', 'through', 'to', 'too', 'under', 'until',
'up', 've', 'very', 'was', 'wasn', "wasn't", 'we', 'were', 'weren', "weren't", 'what',
'when', 'where', 'which', 'while', 'who', 'whom', 'why', 'will', 'with', 'won', "won't",
'wouldn', "wouldn't", 'y', 'you', "you'd", "you'll", "you're", "you've", 'your', 'yours',
'yourself', 'yourselves']
```

statinfer

Demo: Stop Words - spaCy

Number of Stop words in spaCy ==> 326

```
['a', 'about', 'above', 'after', 'again', 'against', 'ain', 'all', 'am', 'an',
'and', 'any', 'are', 'aren', "aren't", 'as', 'at', 'be', 'because', 'been',
'before', 'being', 'below', 'between', 'both', 'but', 'by', 'can', 'couldn',
"couldn't", 'd', 'did', 'didn', "didn't", 'do', 'does', 'doesn', "doesn't",
'doing', 'don', "don't", 'down', 'during', 'each', 'few', 'for', 'from', 'further',
'had', 'hadn', "hadn't", 'has', 'hasn', "hasn't", 'have', 'haven', "haven't",
'having', 'he', 'her', 'here', 'hers', 'herself', 'him', 'himself', 'his', 'how',
'i', 'if', 'in', 'into', 'is', 'isn', "isn't", 'it', "it's", 'its', 'itself',
'just', 'll', 'm', 'ma', 'me', 'mightn', "mightn't", 'more', 'most', 'mustn',
"mustn't", 'my', 'myself', 'needn', "needn't", 'no', 'nor', 'not', 'now', 'o',
'of', 'off', 'on', 'once', 'only', 'or', 'other', 'our', 'ours', 'ourselves',
'out', 'over', 'own', 're', 's', 'same', 'shan', "shan't", 'she', "she's",
'should', "should've", 'shouldn', "shouldn't", 'so', 'some', 'such', 't', 'than',
'that', "that'll", 'the', 'their', 'theirs', 'them', 'themselves', 'then', 'there',
'these', 'they', 'this', 'those', 'through', 'to', 'too', 'under', 'until', 'up',
've', 'very', 'was', 'wasn', "wasn't", 'we', 'were', 'weren', "weren't", 'what',
'when', 'where', 'which', 'while', 'who', 'whom', 'why', 'will', 'with', 'won',
"won't", 'wouldn', "wouldn't", 'y', 'you', "you'd", "you'll", "you're", "you've",
'your', 'yours', 'yourself', 'yourselves']
```

statinfer

Demo: Stop Words removal

```
twitter_data["After_Removing_Stopwords"] = twitter_data["word_tokens"].appl
y(lambda x:[t for t in x if t not in spacy_stopwords ])

print(twitter_data[["raw_tweet","After_Removing_Stopwords"]].sample(10))
```

Output

```
original tweet ==> ['i', 'cannot', 'believe', 'how', 'tired', 'i', 'am',
'right', 'now...', 'i', 'do not', 'know', 'if', 'i', 'can', 'go', 'out',
'tonight...', 'exhaaaausted!!’]

After Removing Stopwords ==> ['believe', 'tired', 'right', 'now...', 'do not',
'know', 'tonight...', 'exhaaaausted!!']
```

Demo: Stop Words removal

	raw_tweet	After_Removing_Stopwords
17138	Here`s a definition of network neutrality for ...	[here`s, definition, network, neutrality, inte...
23452	Happy Mothers Day to all Mom`s	[happy, mothers, day, mom`s]
12719	Goodnight everyone Happy Mothers Day to all o...	[goodnight, happy, mothers, day, mothers]
6568	Wat a nice day it was 2day!me and rachel decid...	[wat, nice, day, 2day, !, rachel, decided, wal...
2295	ahh! Yay! so you`re gonna get it?	[ahh, !, yay, !, you`re, gon, na, ?]
12402	endodontist should be able to do it without r...	[endodontist, able, removing, ., hope, ceramic...
8756	O and i have to wear a **** jacket today cos i...	[o, wear, ****, jacket, today, cos, didnt, not...
9375	waaah.. I can`t open my eyes wider! i wanna go...	[waaah.., can`t, open, eyes, wider, !, wan, na...
24357	New pic.... Twitter is finally letting me chan...	[new, pic, ..., ., twitter, finally, letting, ...
25695	Oh good God crampsss...	[oh, good, god, crampsss, ...]

statinfer

Add custom stop words

```
from spacy.lang.en.stop_words import STOP_WORDS as spacy_stopwords
spacy_stopwords.update({"would", "rt","like", "ha", "lol", "need", "do"})
print("New Number of Stop words in spaCy ==>", len(spacy_stopwords))
print(sorted(spacy_stopwords))
```

RegEx : Regular Expressions

- How to identify/ remove numbers from before analysis
- How to identify/ remove currency symbols?
- How to identify/ remove punctuation ?
- How to identify/ Http, email address, other symbols etc.,

RegEx : Regular Expressions

- As we are working with text data, a bit of manipulation is needed.
- Regular Expressions can help us finding pattern and replace or modify them.
- Regex is it's own language, and is basically the same no matter what programming language we are using with it.
- As Regex can be too vast, we will just go through very minimal to help us find and replace some bits and pieces in our data.
- We will use python package re and it's function re.sub()

Syntax

re.sub(pattern, replacement, string)

statinfer

RegEx Basic Symbols

REGEX BASICS	DESCRIPTION
^	The start of a string
$	The end of a string
.	Wildcard which matches any character, except newline (\n).
|	Matches a specific character or group of characters on either side (e.g. a|b corresponds to a or b)
\	Used to escape a special character
a	The character "a"
ab	The string "ab"

statinfer

RegEx- Syntax

- Numeric String /^[0-9]+$/
- An Identifier (or Name) /[a-zA-Z_][0-9a-zA-Z_]*/
- An Image Filename /^\w+\.(gif|png|jpg|jpeg)$/I
- Email address detection /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/
- HTTP Address /^http:\/\/\S+(\/\S+)*(\/)?$/

statinfer

Commonly used regular expressions

- Digits
  - Whole Numbers - /^\d+$/
  - Decimal Numbers - /^\d*\.\d+$/
  - Whole + Decimal Numbers - /^\d*(\.\d+)?$/
  - Negative, Positive Whole + Decimal Numbers - /^-?\d*(\.\d+)?$/
  - Whole + Decimal + Fractions - /[-]?[0-9]+[,.]?[0-9]*([\/][0-9]+[,.]?[0-9]*)*/
- Alphanumeric Characters
  - Alphanumeric without space - /^[a-zA-Z0-9]*$/
  - Alphanumeric with space - /^[a-zA-Z0-9]*$/
- Email
  - Common email Ids - /^([a-zA-Z0-9._%-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6})*$/
  - Uncommon email ids - /^([a-z0-9_\.\+-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/
- URLs
  - /https?:\/\/(www\.)?[-a-zA-Z0-9@:%._\+~#=]{2,256}\.[a-z]{2,6}\b([-a-zA-Z0-9@:%_\+.~#()?&//=]*)/

statinfer

Code - Regular Expressions

```
def clean_with_re(x):
  x=str(x)

  x=re.sub(r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+'," ", x) #Remove URLs

  x=re.sub(r'[^\w ]+', "", x) # Remove Punctuation-1

  x=re.sub(r"[,!@&\'?\.$%_]"," ", x) # Remove Punctuation-2

  x=re.sub(r"\d+"," ", x) #Remove digits

  return(x)
```

Code - Regular Expressions

	After_Removing_Stopwords	tweet_cleaned_Regex
7779	[Mmm,, comfort, junk, food., That, sounds, goo...	Mmm comfort junk food That sounds good I prefe...
15411	[Sweet, USB, Charles, Marie, site,, bought, ht...	Sweet USB Charles Marie site bought
6393	[It`s, perfect, outside, work]	Its perfect outside work
3239	[First, Day, new, job!, Yeah!]	First Day new job Yeah
3057	[Watching, 3rd, episode, JONAS]	Watching rd episode JONAS
10202	[Bass, drum, heads, breaking, equals, bust., W...	Bass drum heads breaking equals bust Who needs...
2499	[http://twitpic.com/4t4jv, -, No, joke, puppy,...	No joke puppy Maddie looks EXACTLY like Mau...
2793	[Sitting,, wondering,, &, wishing, bunch, stuff]	Sitting wondering wishing bunch stuff
10616	[J, Ross, can`t, leave, killers, singing, run,...	J Ross cant leave killers singing run titles ...
25541	[Playing, 'The, Simpsons, Game', Nitentdo, DS,...	Playing The Simpsons Game Nitentdo DS Light

Spelling Correction

- We need to install text blob package for spell check and correction
- It is not 100% accurate but its good enough for most of the datasets.
- We can avoid this step on datasets like twitter where the text is often written in the short forms and informal language.

```
from textblob import TextBlob
sample_tweet="What an grat and amazimg week. I am excited to learn data scienec"
corrected_tweet=TextBlob(sample_tweet).correct()
corrected_tweet
```

```
TextBlob("That an great and amazing week. I am excited to learn data science")
```

Lemmatization or Root form

- Few words in the document have same root but used in different ways
- For example
  - Create, creating, created, creates
  - Buying, Bought, Buy

- Different words are derived from same root word.
- Same root word but different forms like plural form, adverb form, present tense, past tense form, continuous tense form.
- Lemmatization gives us Lemma, which **can be looked up** in a dictionary.
- Mostly word and it's generated Lemma are very similar words.

statinfer

Lemmatization or Root form

```
spacy_model = spacy.load('en_core_web_sm')

sample_tweet=twitter_data["tweet_cleaned_Regex"][0]
print("Original Text", sample_tweet)

print("Lemmatization Results", " ".join([t.lemma_ for t in spacy_model(str(samp
le_tweet))]))

print("Lemmatization PRON removed", " ".join([t.lemma_ for t in spacy_model(str
(sample_tweet)) if t.lemma_ !="-PRON-" ]))
```

```
Original Text ==> i would responded going
Lemmatization Results ==> i would respond go
```

statinfer

Lemmatization or Root form

```
twitter_data["Lemmatized_tweet"] = twitter_data["tweet_cleaned_Regex"].
apply(lambda x:" ".join([t.lemma_ for t in spacy_model(str(x))if t.lemm
a_ !="-PRON-" ]))
```

raw_tweet	Lemmatized_tweet
Since obviously living in Alaska, only a few ...	since obviously live Alaska radio station here
Cashflow forecasts are fun but big red numbers...	cashflow forecast fun big red number be not
doing my english essay (on r&j...wtf) that i s...	english essay on rjwtf long time ago assign th...
stopped working on the image database awhile ...	stop work image database awhile ago help Greg ...
Gonna nap n chill then probably go to the movi...	go to nap n chill probably movie later Ugh hea...
Trickery? No, just exasperation at seeing **...	trickery no exasperation see pride hijacked ...
http://twitpic.com/4vw9a - lol wow I`m watchi...	lol wow be watch Xmen the Stand right
Finished my Lunch	Finished lunch
I`m bummed that I can`t wear my sweet Nike kic...	be bummed can not wear sweet Nike kick work
webcam still wont work evil stuffs.	webcam will not work evil stuff

statinfer

Final Cleaning after Lemmatization

```
spacy_stopwords.update({"would", "rt","like", "ha", "lol", "need", "do"})

twitter_data["Final_Cleaned_Tweet"] = twitter_data["Lemmatized_tweet"].apply(lambda x:[
t for t in str(x).split() if t not in spacy_stopwords ])

twitter_data["Final_Cleaned_Tweet_tokens"]=twitter_data["Final_Cleaned_Tweet"].apply(la
mbda x: " ".join(x) )

twitter_data[["raw_tweet","Final_Cleaned_Tweet_tokens"]].sample(10)
```

Word Cloud

```
#!pip install wordcloud
from wordcloud import WordCloud
import matplotlib.pyplot as plt
%matplotlib inline

final_text="".join(twitter_data["Final_Cleaned_Tweet_tokens"])
len(final_text)

plt.figure(figsize = (15, 15), facecolor = None)
wc=WordCloud(colormap='Set2').generate(final_text)
plt.imshow(wc)
plt.axis("off")
plt.show()
```

Word Cloud

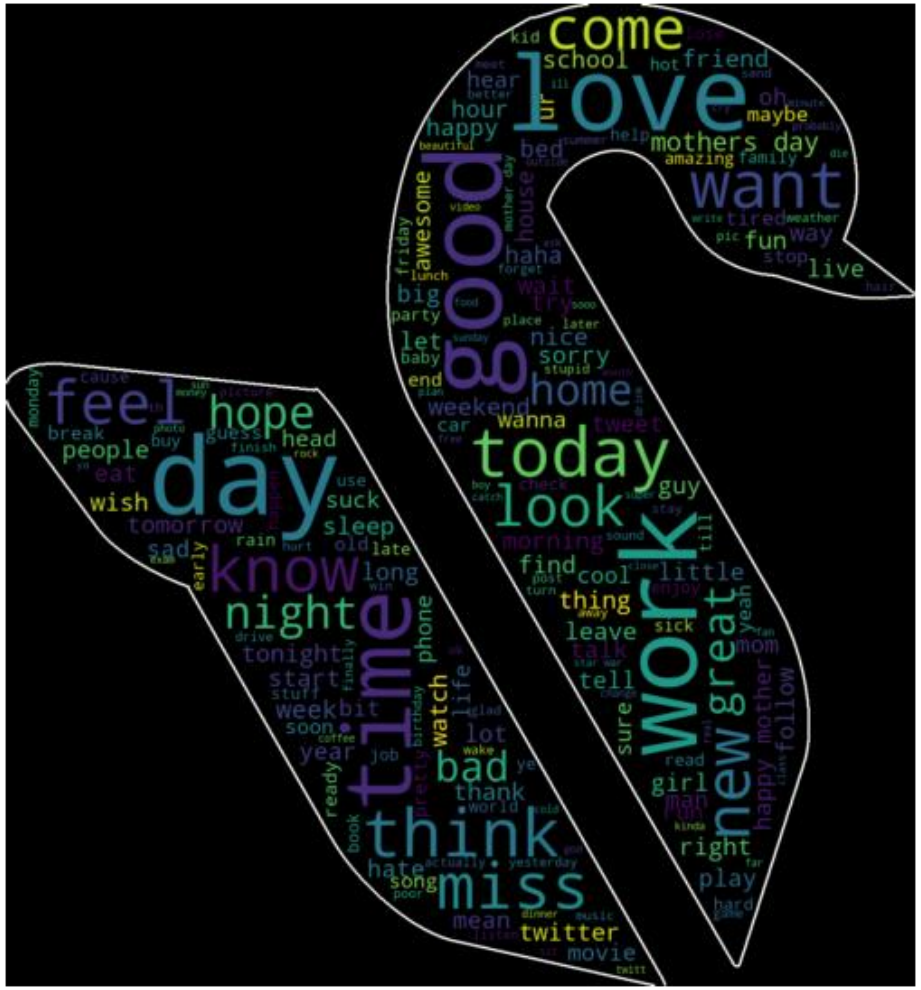
stop
little
tweet
start
thank
come
look
wait
play
time
work
new
tonight
nice
book
break
kid
right
morning
hate
rain
people
read
let
oh
man
ready
tired
school
away
weekend
sad
wish
watch
tomorrow
long
meet
house
life
fun
good
night
today
know
friday
use
hot
bit
cool
mom
send
late
amazing
job
hope
mothers day
happy mother
yesterday
help
live
monday
head
find
leave
big
movie
try
sound
feel
thing
great
sorry
drive
family
phone
guy
sleep
soon
tell
run
sure
hour
follow
girl
th
till
way
want
day
yeah
friend
baby
love
awesome
bed
cause
wanna
hard
eat
party
haha
world
miss
old
talk
think
ur
song
lot
week
happy
home
buy
twitter
check
lose
stuff
enjoy
bad

Word Cloud - Formatted

```
BG_image = np.array(Image.open("statinfer-logo-transparent_v1.png"))
plt.imshow(BG_image)
plt.axis("off")

plt.figure(figsize = (10, 10))
wc=WordCloud(mask=BG_image, contour_color='white', contour_width=3).ge
nerate(final_text)
plt.imshow(wc, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Word Cloud - Formatted



statinfer

A Single function for pre-processing

- Club all the pre-processing steps into a single function

```
def pre_processing(input_data, text_col):
  input_data["text_col_clean"]=input_data[text_col].apply(lambda
x:str(x).lower())
  input_data["text_col_clean"]=input_data["text_col_clean"].apply
(lambda x:[expanded_form(t) for t in str(x).split()])
  input_data["text_col_clean"]=input_data["text_col_clean"].apply
(lambda x:[t for t in x if t not in spacy_stopwords ])
  input_data["text_col_clean"]=input_data["text_col_clean"].apply
(lambda x:clean_with_re(x))
  input_data["text_col_clean"]=input_data["text_col_clean"].apply
(lambda x:" ".join([t.lemma_ for t in spacy_model(str(x))if t.lem
ma_ !="-PRON-" ]))
  input_data["text_col_clean"]=input_data["text_col_clean"].apply
(lambda x:[t for t in str(x).split() if t not in spacy_stopwords
])
  input_data["text_col_clean"]=input_data["text_col_clean"].apply
(lambda x: " ".join(x) )
```

Document Term Matrix

Document Term Matrix

- Document – text document
- Can we consider each sentence as document? Can we call a sentence as a basic form of document
- We can create DTM and work with sklearn and other regular packages

- Doc1: Loved this place
- Doc2: At this place, crust is not good.
- Doc3: Loved it, good thin crust pizza.

Document Term Matrix

Doc1: Loved this place, good pizza
Doc2: At this place, crust is not good. pizza is not good.
Doc3: Loved it, good thin crust pizza.

Terms

Documents

	loved	this	place	at	crust	is	not	good	it	thin	pizza
Doc1	1	1	1					1			1
Doc2		1	1	1	1	2	2	2			1
Doc3	1				1			1	1	1	1

statinfer

Document Term Matrix - Code

```
from sklearn.feature_extraction.text import CountVectorizer

countvec1 = CountVectorizer(min_df= 5) #minimum word freq=5
dtm_v1 = pd.DataFrame(countvec1.fit_transform(twitter_data['Final_Clea
ned_Tweet_tokens']).toarray(), columns=countvec1.get_feature_names(),
index=None)

print(dtm_v1.shape)
dtm_v1
```

Document Term Matrix - Code

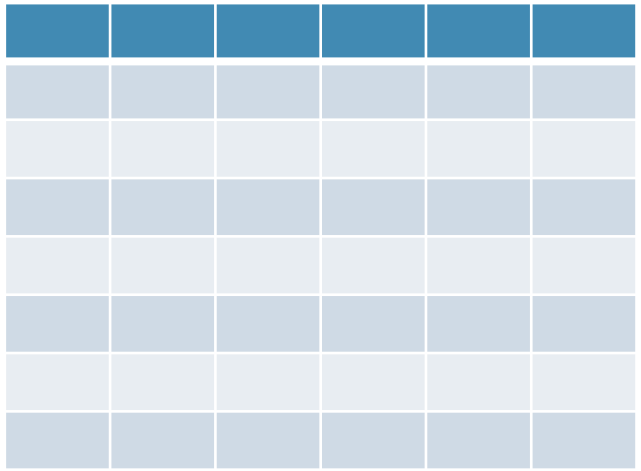
	aaaah	aah	abandon	ability	abit	able	absolutely	abt	ac	academy	accept	access	accident	accidentally	accord	account	acct	ace
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
27476	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27477	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27478	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27479	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Two Steps in NLP Model building

Step1=> Convert text data into numerical data - **Done**

Step2=> Build models on numerical data - **Upcoming**

Step1



Step2

Conclusion

Conclusion

- Text is unstructured / semi structed data.
- Preparing data analysis is the key step in text mining
- Text mining involves lot of customisation
- We discussed very basic steps of data preparation and summarisation of text
- Word2Vec is an alternative to DTM