Natural Language Processing

DEEPAK KUMAR

Agenda

Introduction

NLP Understanding

Text Pre-processing Techniques for any NLP task

Naïve Bayes Algorithms

Sentiment Analysis using Naïve Bayes Algorithm in Python

What is NLP

Natural Language Processing(NLP) is an area of Computer Science and Artificial Intelligence discipline concerned with interactions between human and machines.

Objective – To build or train intelligent computers that can interact with human like a human being

Few Basic Stats about Data generated

On average, Google now processes more than 40,000 searches ,EVERY second (3.5 billion searches per day)!

Snapchat users share 527,760 photos per minute

More than 120 professionals join LinkedIn per minute

Users watch 4,146,600 YouTube videos per minute

456,000 tweets are sent on Twitter per minute

Instagram users post 46,740 photos per minute

More than 300 million photos get uploaded per day on Facebook

Every minute there are 510,000 comments posted and 293,000 statuses updated every minute

Every minute there are 103,447,520 spam emails sent

Source:- https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#1e7aa7b760ba

NLP Application

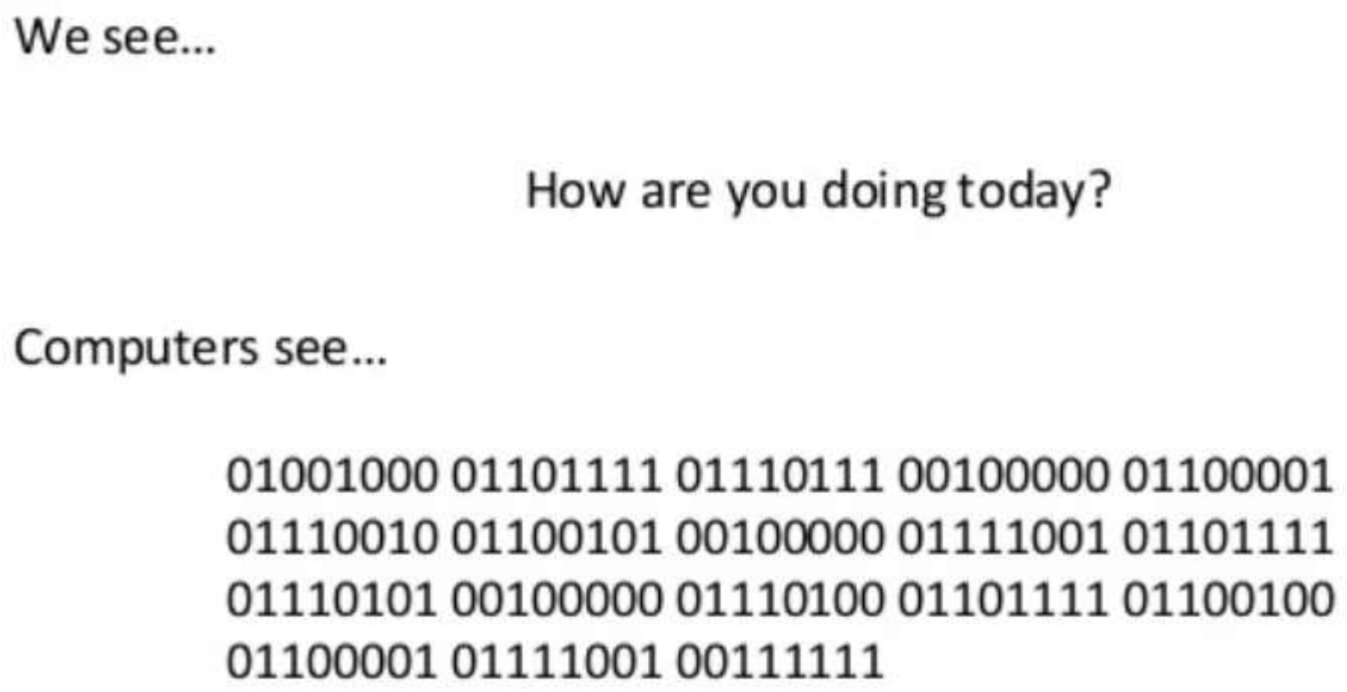
Easy Application :-

Part-of-Speech Tagging, Spam Email Classifications, Sentiment Analysis, Spelling Correction

Complex Application –

Machine Translation, Text Summarization, Text Generation, Document Clustering, Content Segmentation, Chatbot-Questions and Answers

What Makes NLP Challenging?

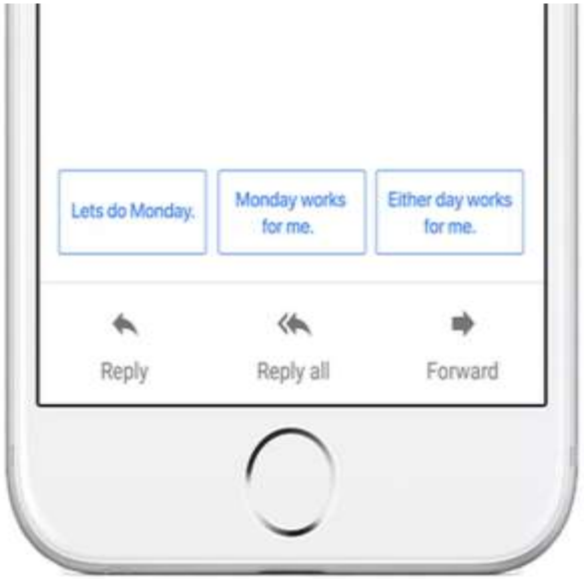


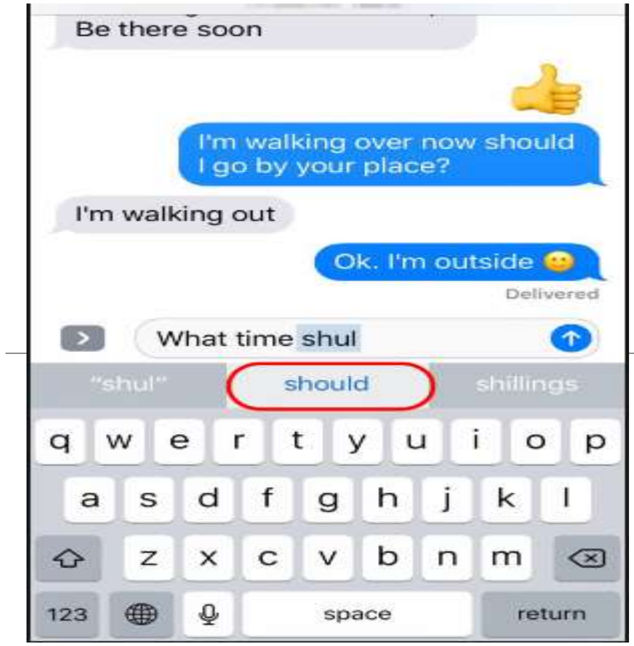
We see…

How are you doing today?

Computers see…

```
01001000 01101111 01110111 00100000 01100001
01110010 01100101 00100000 01111001 01101111
01110101 00100000 01110100 01101111 01100100
01100001 01111001 00111111
```




Be there soon

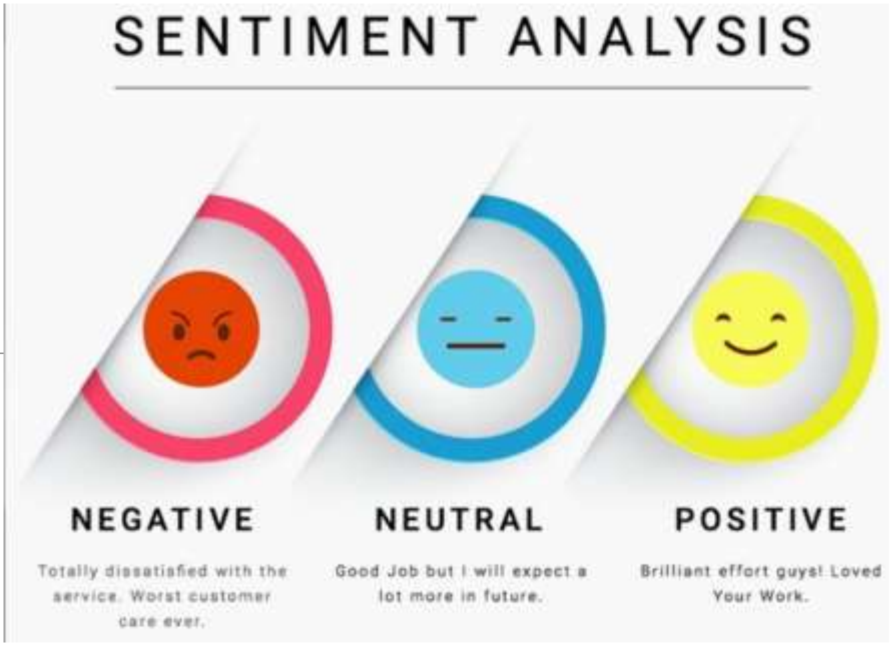
I'm walking over now should I go by your place?

I'm walking out

Ok. I'm outside 😊

Delivered

What time shul


SENTIMENT ANALYSIS

NEGATIVE
Totally dissatisfied with the service. Worst customer care ever.

NEUTRAL
Good Job but I will expect a lot more in future.

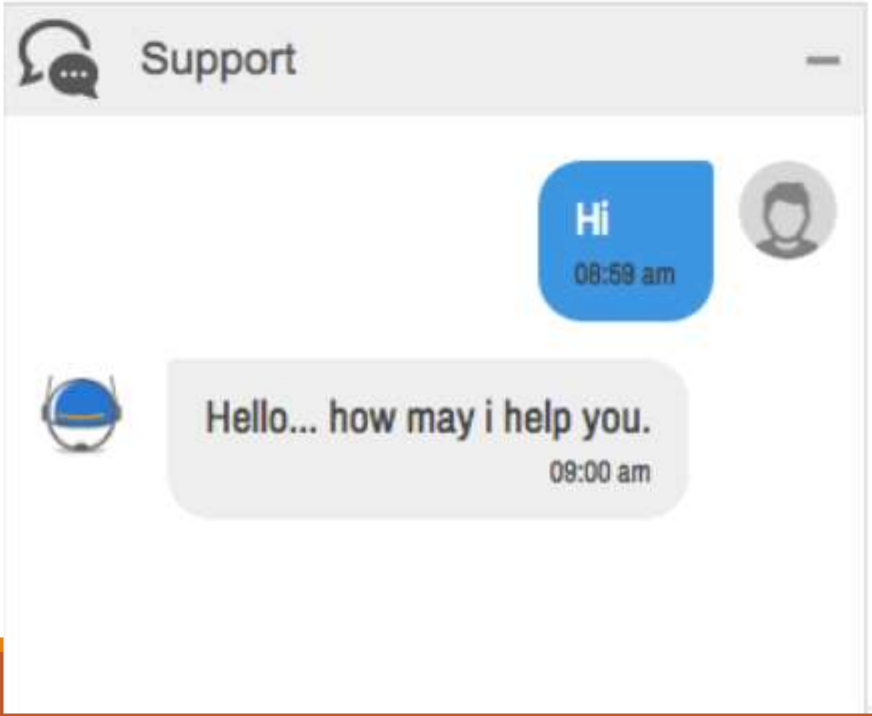
POSITIVE
Brilliant effort guys! Loved Your Work.

मेरा नाम रजत है | => My name is Rajat.

Extraction of meaning from email :-

We have decided to meet tomorrow at 10:00am in the lab.

To do : meeting
Time : 10:00 am, 22/3/2012
Venue : Lab


Support

Hi
08:59 am

Hello... how may i help you.
09:00 am

Text Modeling

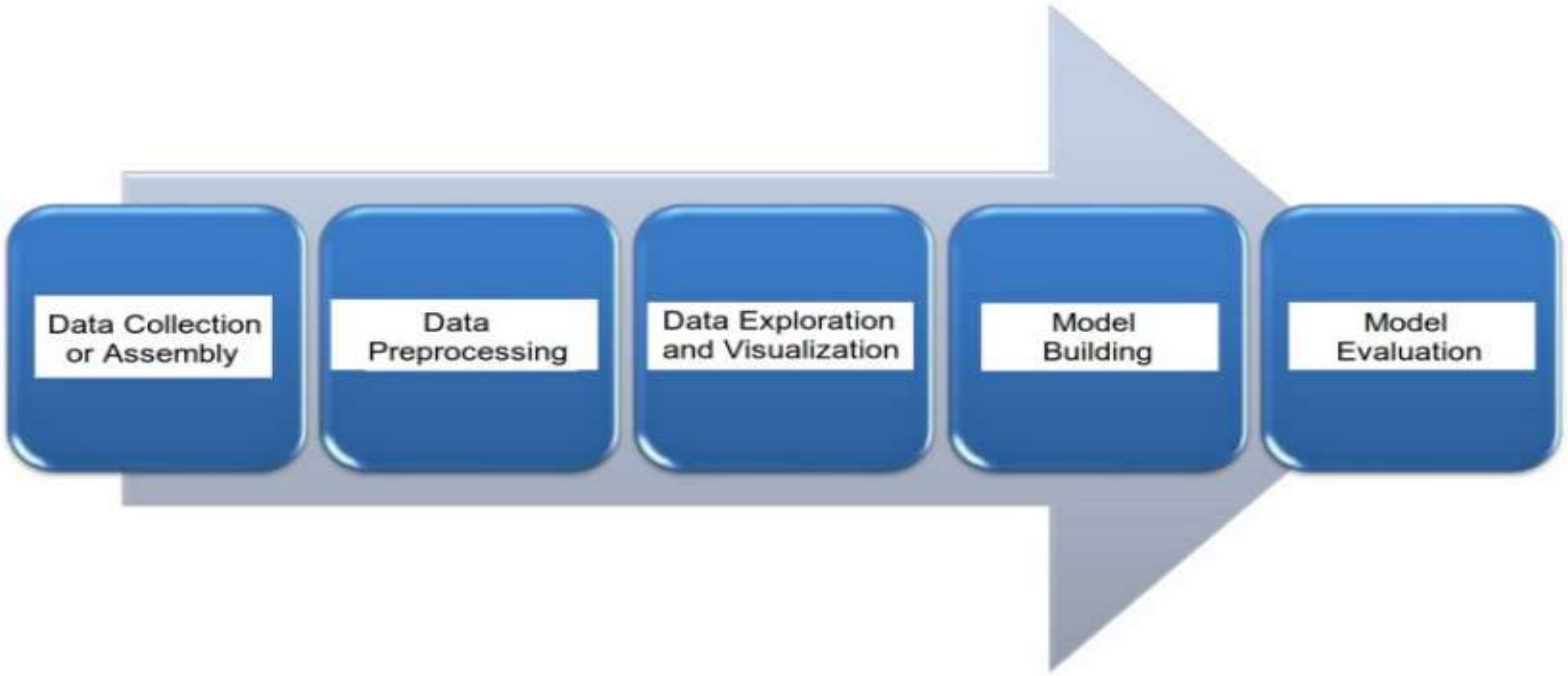
Supervised Techniques :

1. Sentiment Classifications
2. Document Classifications
3. Content Classifications

Un-Supervised Techniques

1. Document Clustering
2. Topic Modeling
3. Text Generation
4. Text Summarization

Text Modeling Process


Data Collection or Assembly
Data Preprocessing
Data Exploration and Visualization
Model Building
Model Evaluation

Importance of text pre-processing

Transforming text into something an algorithm can digest is a complicated process.

1. Cleansing- get rid of less useful text like removing of certain texts and it depends business to business or objective to perform this process
2. Annotation – Its basically use parts-of-speech tagging
3. Normalization – Stemming & Lemmatization
4. Analysis - consists of statistically probing, manipulating and generalizing from the dataset for feature analysis.

Cleansing

Cleaning consist of getting rid of the less useful parts of text through stop word removal, dealing with capitalization and characters and other details.

1.Removal or Extraction of certain noise texts from data

E.g.

S="2013-02-20T17:24:33Z“. I want to extract date from it.

re.findall(r'\d{4}-\d{2}-\d{2}',s)

['2013-02-20’]

z='us is a developed country. USA is a great country to live. Donald Trump is president of US.’

re.sub(r'USA|us|US','USA’,z)

'USA is a developed country. USA is a great country to live. Donald Trup is president of USA.'

Stop Words Removals

Stop Words: A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

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

e.g.

sentence = "This is a sample sentence, showing off the stop words filtration.“

['This', 'sample', 'sentence', ',', 'showing', 'stop', 'words', 'filtration', '.']

Annotation

POS -Understand parts of speech can make difference in determining the meaning of a sentence.

IN:

['And', 'from', 'their', 'high', 'summits', ',', 'one', 'by', 'one', ',', 'drop', 'everlasting', 'dews', '.’]

```
OUT:
[('And', 'CC’),           CC-Cordinating Conjunction
 ('from', 'IN’),          IN preposition/subordinating conjunction
('their', 'PRP$’), PRP$ possessive pronoun my, his, hers
 ('high', 'JJ’),                      JJ adjective 'big'
 ('summits', 'NNS’),         NNS noun plural 'desks'
 (',', ','),
 ('one', 'CD’),                       CD cardinal digit
 ('by', 'IN'),
 ('one', 'CD'),
 (',', ','),
 ('drop', 'NN’),              NN noun, singular  ‘desk
 ('everlasting', 'VBG’), verb, gerund/present participle
('dews', 'NNS’),              noun plural 'desks'
 ('.', '.')]
```

https://pythonprogramming.net/natural-language-toolkit-nltk-part-speech-tagging/

Normalization

Stemming - Stemming is a process where words are reduced to a root by removing inflection through dropping unnecessary characters, usually a suffix. There are several stemming models, including Porter and Snowball.

IN:

["It never once occurred to me that the fumbling might be a mere mistake."]

OUT:

['it', 'never', 'onc', 'occur', 'to', 'me', 'that', 'the', 'fumbl', 'might', 'be', 'a', 'mere', 'mistake.'],

Normalization Contd.

Lemmatization - is an alternative approach from stemming to removing inflection. By determining the part of speech and utilizing WordNet’s lexical database of English, lemmatization can get better results.

better → good

The stemmed form of leafs is: leaf

The stemmed form of leaves is: leav

The lemmatized form of leafs is: leaf

The lemmatized form of leaves is: leaf

Lemmatization is a more intensive and therefor slower process, but more accurate. Stemming may be more useful in queries for databases whereas lemmatization may work much better when trying to determine text sentiment.

Features Creations - Ngrams

N-grams of texts are extensively used in text mining and natural language processing tasks. They are basically a set of co-occuring words within a given window and when computing the n-grams you typically move one word forward (although you can move X words forward in more advanced scenarios). For example, for the sentence *"The cow jumps over the moon"*. If N=2 (known as bigrams), then the ngrams would be:
the cow

cow jumps

jumps over

over the

the moon

Tf-Idf(Term Frequency-Inverse Document Frequency) Matrix

Term Frequency (TF) – TF for a term “t” is defined as the count of a term “t” in a document “D”

Inverse Document Frequency (IDF) – IDF for a term is defined as logarithm of ratio of total documents available in the corpus and number of documents containing the terms

TF . IDF – TF IDF formula gives the relative importance of a term in a corpus (list of documents), given by the following formula below.

TF.IDF= $\mathrm{tf} * \mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$

Tf-Idf Example

Consider a document containing 100 words wherein the word 'Cauvery' appears 3 times.

The term frequency (tf) for 'Cauvery' is then TF = (3 / 100) = 0.03.

Now, assume we have 10 million documents and the word 'Cauvery' appears in 1000 of these. Then, the inverse document frequency (idf) is calculated as IDF = log(10,000,000 / 1,000) = 4.

Thus, the Tf-idf weight is the product of these quantities

TF-IDF = 0.03 * 4 = 0.12

Create Document Term Matrix

```
Doc.
 ID   signed similar since situation six slightly small sold soon source south special speech spending spokesman ....
  1   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 0 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 0 0 0 0
  4   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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  5   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  6   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  7   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 0 0
  8   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  9   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 0 0 0
 10   0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Counts the number of time a word is appearing in each document.

It is a sparse Matrix with most values as 0. Standardize the counts if desired

Exclude words which are appearing too less number of times

Exclude words which are appearing too many number of times

Use this to create classifier model

Doc. ID	Label	signed similar since situation six slightly small sold soon source south special speech spending spokesman ....
1	Money	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2	Money	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3	Crude	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 0 0 0
4	Money	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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5	Money	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6	Money	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7	Crude	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 0 0
8	Crude	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 0 0 0 0
9	Money	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10	Crude	0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Use KNN as a Classifier Model for Topic (Money/Crude Oil)
Use SVM as a Classifier Model for Topic (Money/Crude Oil)
Use Naïve Bayes as a Classifier Model

Term Frequency Matrix

```
##        Docs
  Terms    1 2 3 4
   blue    1 0 0 0
   bright  0 1 1 1
   can     0 0 0 1
   see     0 0 0 1
  shining  0 0 0 1
  sky      1 0 1 0
  sun      0 1 1 2
  today    0 1 0 0
```

Idf

```
( idf <- log( ncol(tf) / ( 1 + rowSums(tf != 0) ) ) )
     blue           bright     can            see        shining     sky
 0.6931472 0.0000000 0.6931472 0.6931472 0.6931472 0.2876821
Sun                today
0.0000000     0.6931472
```

Tf-idf

```
Docs     blue              bright     can      see   shining     sky      sun
  1      0.6931472      0 0.0000000 0.0000000 0.0000000 0.2876821   0
  2      0.0000000      0 0.0000000 0.0000000 0.0000000 0.0000000   0
  3      0.0000000     0 0.0000000 0.0000000 0.0000000 0.2876821    0
  4       0.0000000      0 0.6931472 0.6931472 0.6931472 0.0000000  0
```

Normalization of Matrix

```
# Note that normalization is computed "row-wise"

tf_idf / sqrt( rowSums( tf_idf^2 ) )

##

## Docs      blue    bright     can       see       shining       sky     sun  today

##    1 0.9236103      0 0.0000000 0.0000000 0.0000000 0.3833329   0     0

##    2 0.0000000      0 0.0000000 0.0000000 0.0000000 0.0000000   0     1

##    3 0.0000000      0 0.0000000 0.0000000 0.0000000 1.0000000   0     0

##    4 0.0000000      0 0.5773503 0.5773503 0.5773503 0.0000000   0     0
```

What is Naïve Bayes Algorithm

Supervised Classification Technique

Based on Bayes Theorem

Assumptions – Each features in a class is independent to each other

Basic Example :-

A fruit may be considered as apple if its red, round and diameter is 3.5 inches approx.

Why naïve :-

Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’

Advantages

Its simple and easy to understand and fast computaion

Easy to build and very useful on large dataset

Outperform even highly sophisticated classifications methods

Bayes Theorem

The probability of an event, based on prior knowledge of conditions that might be related to the event.

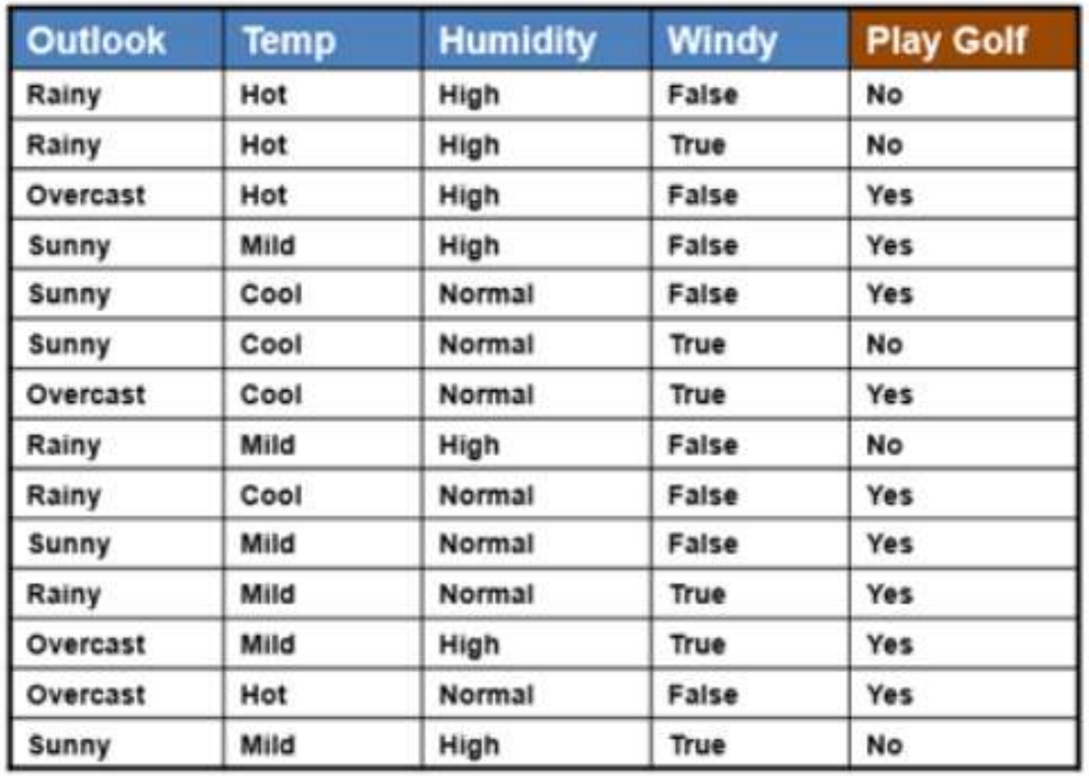
$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- $P(c|x)$ is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

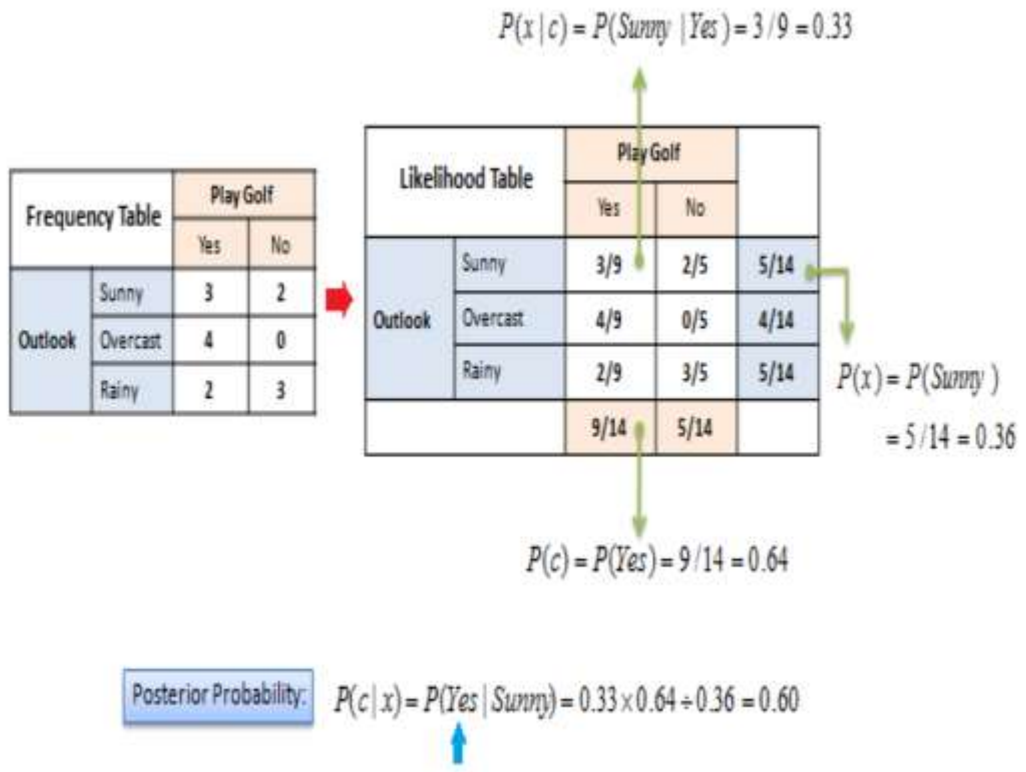
Example

We use the same simple Weather dataset here.



Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Example Contd…



Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

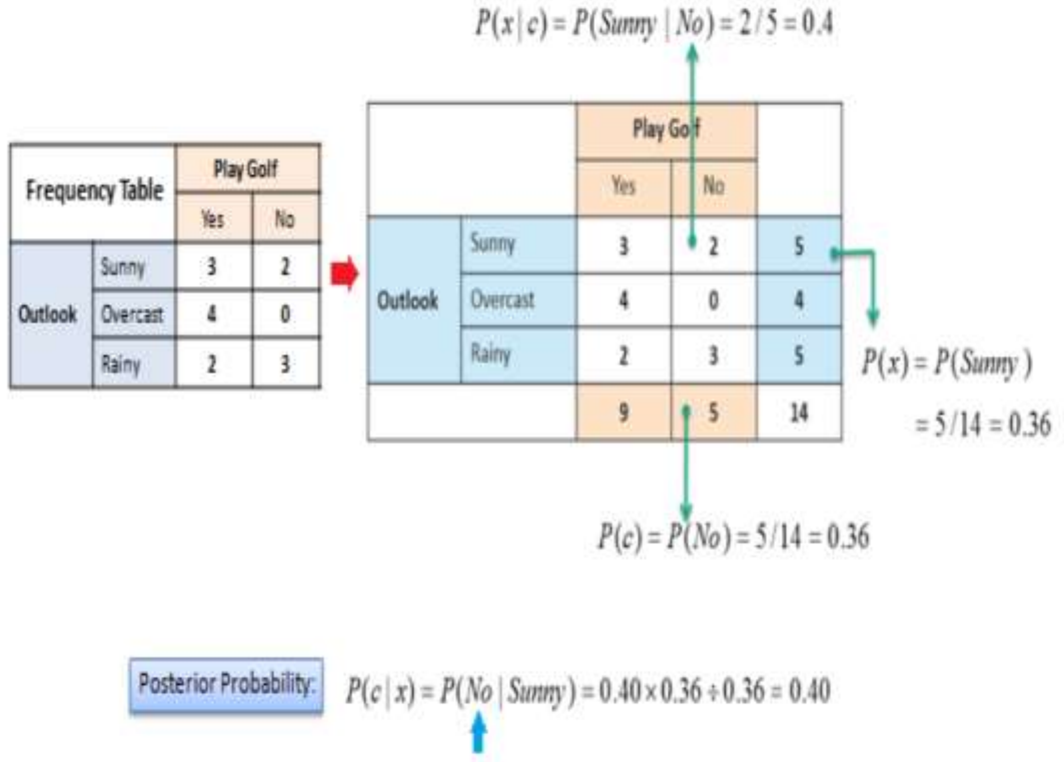
Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	3/9	2/5	5/14
	Overcast	4/9	0/5	4/14
	Rainy	2/9	3/5	5/14
		9/14	5/14	

$P(x \mid c) = P(Sunny \mid Yes) = 3/9 = 0.33$

$P(x) = P(Sunny)$
$= 5/14 = 0.36$

$P(c) = P(Yes) = 9/14 = 0.64$

Posterior Probability: $P(c \mid x) = P(Yes \mid Sunny) = 0.33 \times 0.64 \div 0.36 = 0.60$



Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
		9	5	14

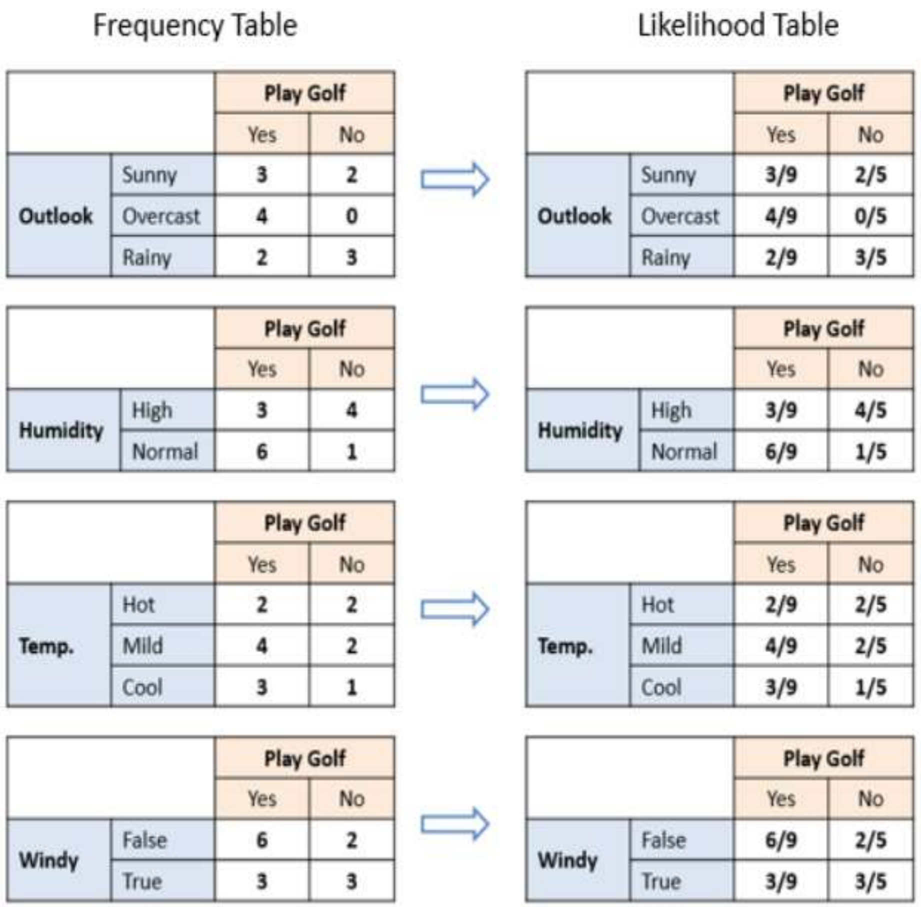
$P(x \mid c) = P(Sunny \mid No) = 2/5 = 0.4$

$P(x) = P(Sunny)$
$= 5/14 = 0.36$

$P(c) = P(No) = 5/14 = 0.36$

Posterior Probability: $P(c \mid x) = P(No \mid Sunny) = 0.40 \times 0.36 \div 0.36 = 0.40$

Naïve Bayes Theorem



Frequency Table

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

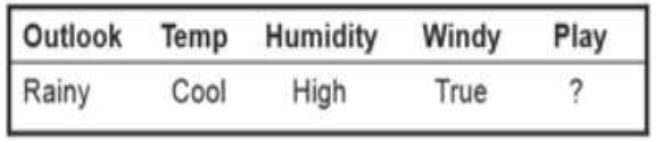
Likelihood Table

		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5



Outlook	Temp	Humidity	Windy	Play
Rainy	Cool	High	True	?

$$P(Yes \mid X) = P(Rainy \mid Yes) \times P(Cool \mid Yes) \times P(High \mid Yes) \times P(True \mid Yes) \times P(Yes)$$

$$P(Yes \mid X) = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.00529 \rightarrow 0.2 = \frac{0.00529}{0.02057 + 0.00529}$$

$$P(No \mid X) = P(Rainy \mid No) \times P(Cool \mid No) \times P(High \mid No) \times P(True \mid No) \times P(No)$$

$$P(No \mid X) = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.02057 \rightarrow 0.8 = \frac{0.02057}{0.02057 + 0.00529}$$

What is the benefit of doing all these?

The Only objective to perform all these operations on text to identify or represent them in way where we can identify which all documents or tweets or sentences are similar in nature or representations.

We use cosine similarity for calculating the similarity. We usually calculate dot products of two vectors and its values lies between 0-1. higher the values more two vectors or sentences or documents are similar.

Business Challenge – Sentiment Analysis

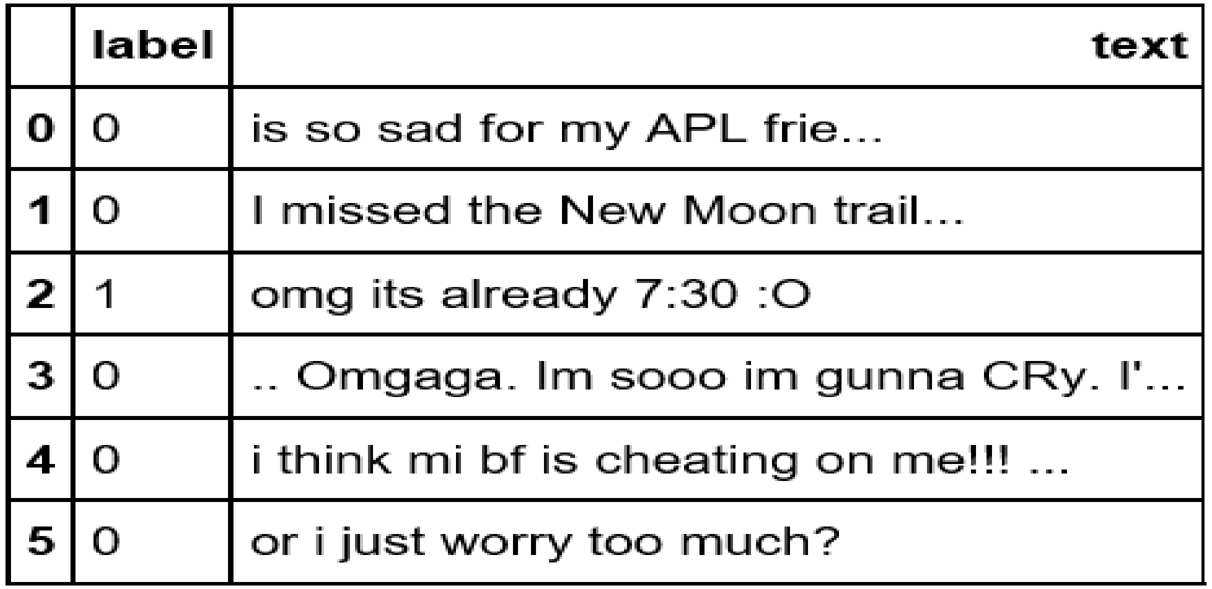
Twitter is popular microblogging site

Short text messages max 140 characters

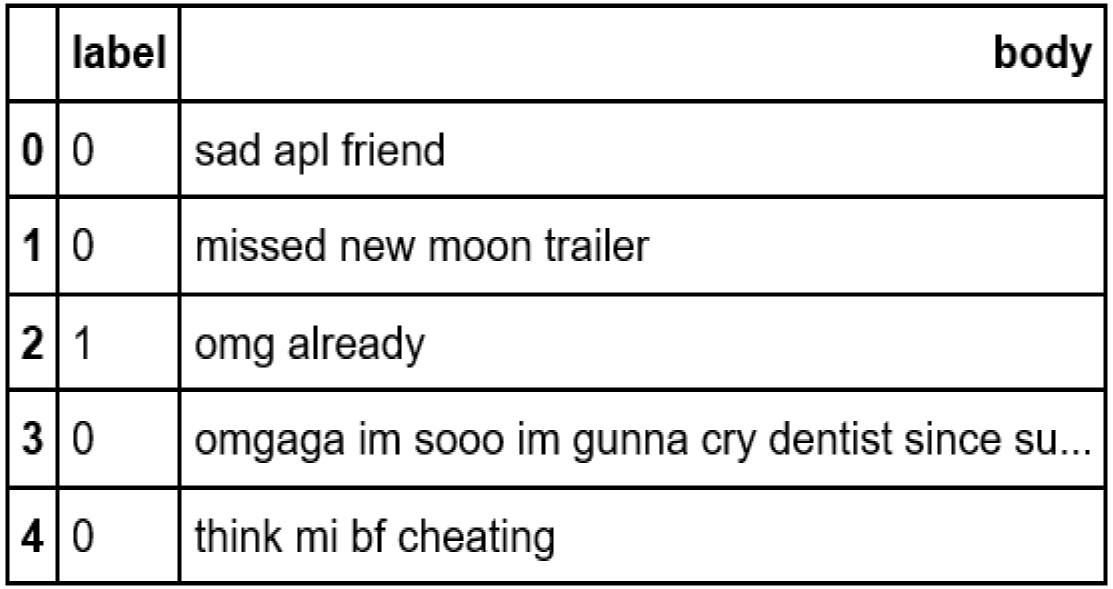
Twitter audiences varies from common man to celebrities

Tweets are not structures provide freedom to express your views in your language

Twitter



	label	text
0	0	is so sad for my APL frie...
1	0	I missed the New Moon trail...
2	1	omg its already 7:30 :O
3	0	.. Omgaga. Im sooo im gunna CRy. I'...
4	0	i think mi bf is cheating on me!!! ...
5	0	or i just worry too much?



	label	body
0	0	sad apl friend
1	0	missed new moon trailer
2	1	omg already
3	0	omgaga im sooo im gunna cry dentist since su...
4	0	think mi bf cheating

"   .. Omgaga. Im sooo im gunna CRy. I've been at this dentist since 11.. I was suposed 2 just get a crown put on (30mins)..."

'omgaga im sooo im gunna cry dentist since   suposed get crown put mins'

```
1    56457
0    43532
Name: Sentiment, dtype: int64
```

Model Building

I have used two model Naïve Bayes Classifiers and SVM Classifiers to build the classifications model and prediction

Model	Naïve Bayes	RF
Overall	71.5%	69.44%
Class 0	60%	69.27%
Class1	81%	69.58%

Deep Learning Approach

If times permit

Challenges with traditional model

Hotel Booking in Mumbai on 28th June

Mumbai motel availability on 28th June

But you can know from this Motel and hotel are same in both but if we do word representations by one hot encoding and calculate dot product for these two words it will be 0.

This is the biggest obstacles while working on text analytics stuff.

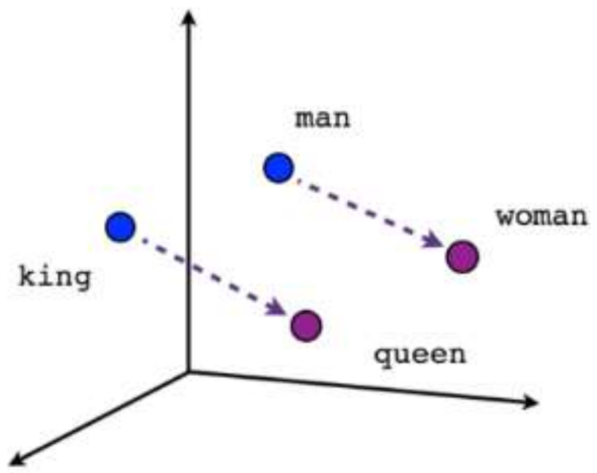
Deep Learning Approach – Word2Vec

Word2Vec is a word embedding model which converts words in sentences or documents to numeric on the basis of their representations in the sentence, documents and etc.

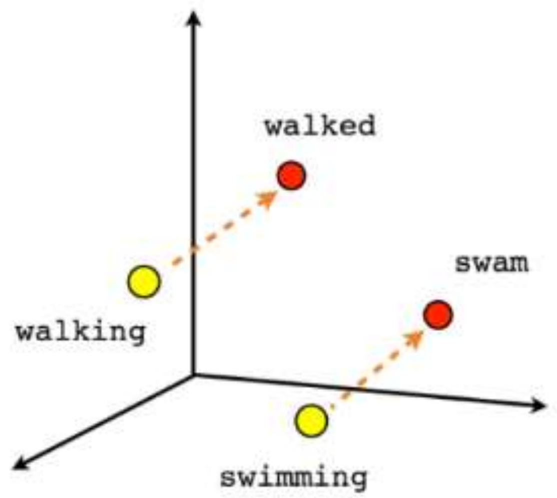
It uses shallow neural network with one hidden layer and input and output neurons are equal.

```
model.similarity('woman','man')

0.73723527

model.doesnt_match('breakfast cereal dinner lunch';.split())

'cereal’

model.most_similar(positive=['woman','king'],negative=['man'],topn=1)

queen: 0.7508
```

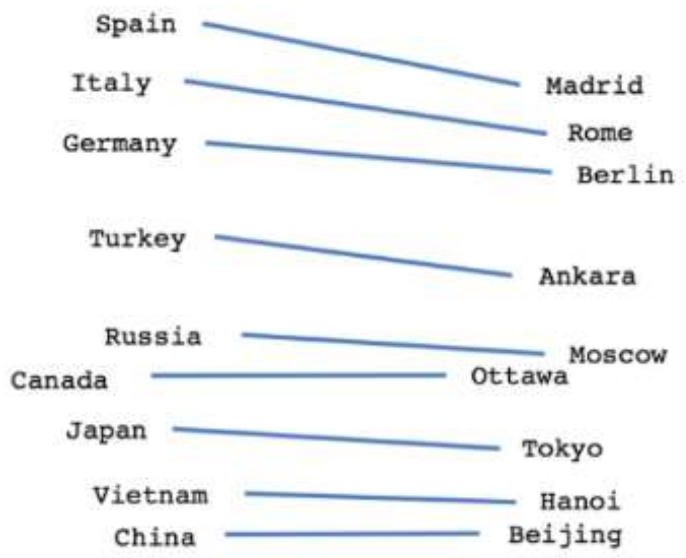
Word2Vec Visualization



Male-Female



Verb tense



Country-Capital

Word2Vec applications

Product Recommendation

Healthcare Applications

Thanks