MIS780 – Advanced AI for Business

Week 2

Text Analytics I

Dr Anagi Gamachchi

Discipline of Information Systems and Business Analytics,
Deakin Business School

What are the sources of text data?

Why should businesses care about textual data?

Outline

- **Text Analytics Framework**
- Text Classification
- Sentiment Analysis
  - Lexicon-based Sentiment Analysis
  - Aspect-level Sentiment Analysis

Text analytics framework

1 – Data Collection or Assembly:

- *Obtain* or *build corpus* (eg. Emails, customer's feedback, company's financial reports.)

Clothing ID	Age	Title	Review Text
767	33		Absolutely wonderful - silky and sex
1080	34		Love this dress! it's sooo pretty. i h
1077	60	Some ma	I had such high hopes for this dress
1049	50	My favori	I love, love, love this jumpsuit. it's fu
847	47	Flattering	This shirt is very flattering to all due

Can we apply any statistical or machine learning techniques to this data?

2 – Text Preprocessing:

- Perform the preparation tasks on the raw text corpus consists of a number of steps, which generally fall under the broad categories of ***tokenization***, ***normalization***, ***noise removal,*** *and* ***text representation***

Processed Textual Data

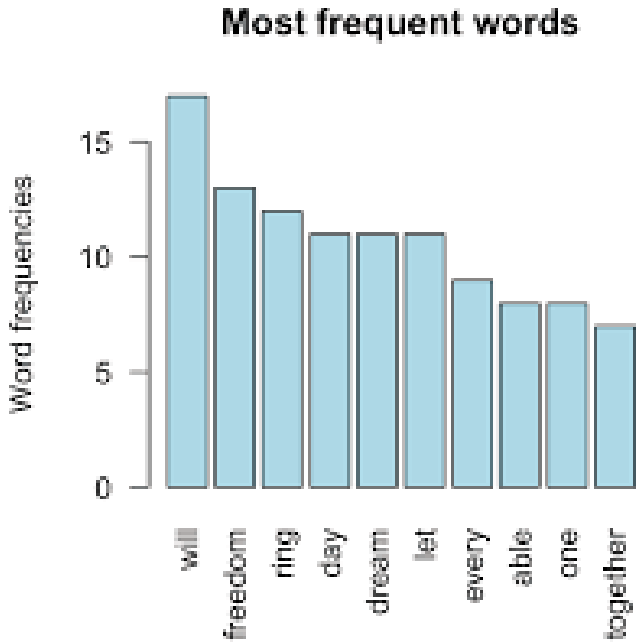
advanc	analysi	analyt	book
0	1	0	1
1	0	1	1

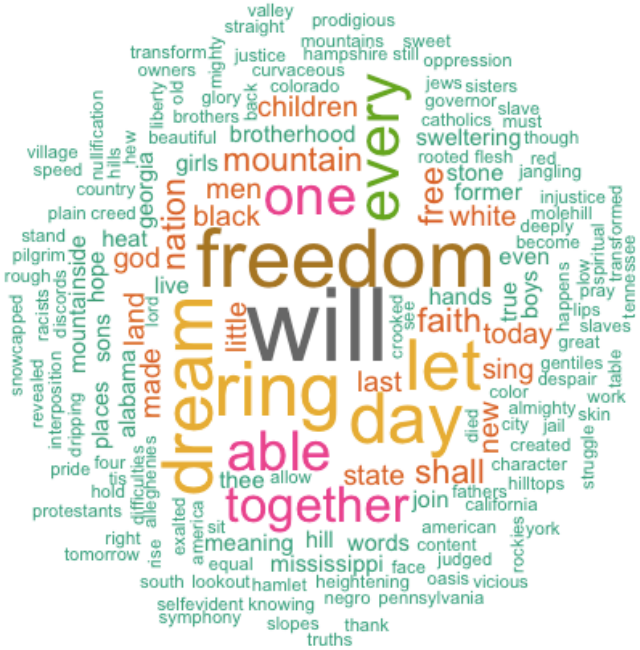
https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html

Text analytics framework (cont.)

3 – Exploration & Visualization:

- Exploring and visualizing it is an essential step in gaining insight (regardless of what our data is -- **text** or not)
- Common tasks: visualizing *word counts*, *distributions*, *word clouds*, etc.





http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-you-should-know

Text analytics framework (cont.)

4 – Model Building:

- Feature selection & engineering
- Training and testing models:
  - ***Language models***.
  - ***Machine learning classifiers***
  - ***Sequence models***

5 – Model Evaluation :

- Did the model perform as expected?
- Metrics will vary dependent on what type of text mining or NLP task

http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-you-should-know

Text Processing Operations

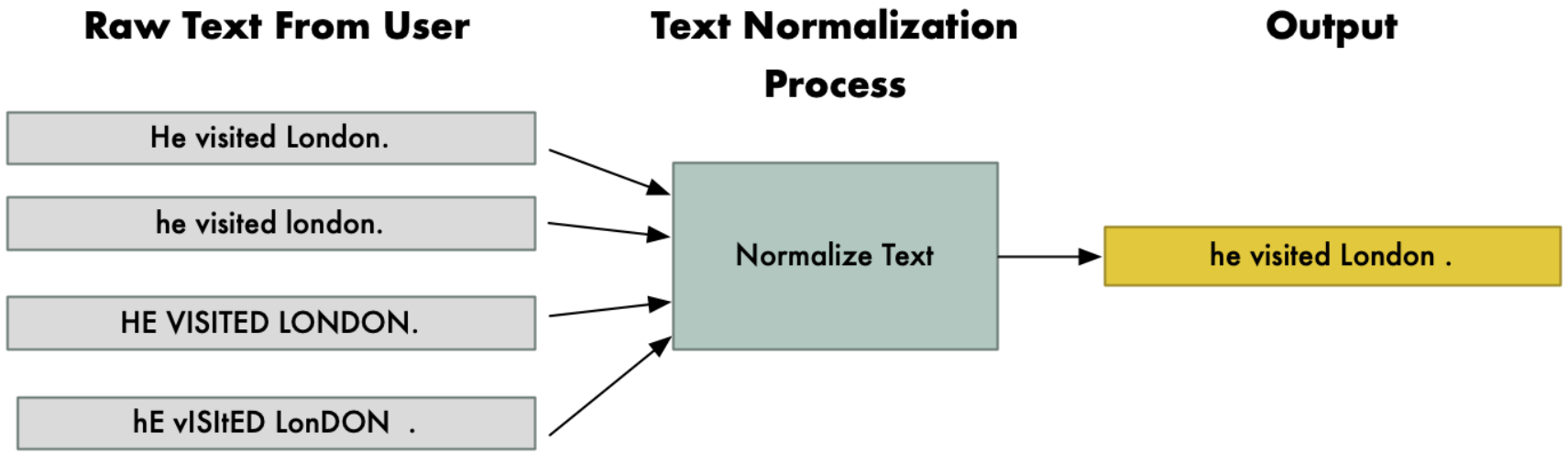
- **Tokenization & Segmentation:** splits longer strings of text into smaller pieces, or tokens.

```
raw = '''Online activities such as articles, website text, blog posts, social
media posts are generating unstructured textual data.'''


from nltk.tokenize import word_tokenize

tokens = word_tokenize(raw)
print(tokens)
```

```
['Online', 'activities', 'such', 'as', 'articles', ',', 'website', 'text',
',', 'blog', 'posts', ',', 'social', 'media', 'posts', 'are', 'generating',
'unstructured', 'textual', 'data', '.']
```

Text Processing Operations (cont.)

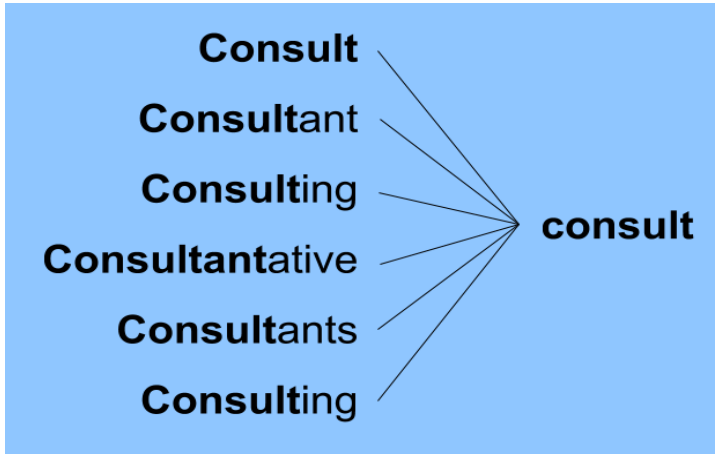
- **Normalization:** a series of related tasks meant to put all text on a same level , such as converting all text to the *same case* (*upper* or *lower*), *removing punctuation*, *converting numbers* to their *word equivalents*, and so on..



https://devopedia.org/text-normalization

Text Processing Operations (cont.)

- **Stemming:** eliminating affixes (suffixed, prefixes, infixes, etc.) from a word in order to obtain a word stem




```
['Online', 'activities', 'such', 'as', 'articles',
',', 'website', 'text', ',', 'blog', 'posts', ',',
'social', 'media', 'posts', 'are', 'generating',
'unstructured', 'textual', 'data', '.']
```

```
from nltk import PorterStemmer

porter = PorterStemmer()
[porter.stem(t) for t in tokens]
```

['onlin', 'activ', 'such', 'as', 'articl', ',', 'websit', 'text',
',', 'blog', 'post', ',', 'social', 'media', 'post', 'are',
'gener', 'unstructur', 'textual', 'data', '.']

https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html

Text Processing Operations (cont.)

- **Stop Words Removal:** such as "*the*" "*and*" and "*a*" don't generally contribute greatly to one's understanding of content.

```
from nltk.corpus import stopwords
stop_words=set(stopwords.words("english"))
print(stop_words)


filtered_sent=[]
for w in tokens:
    if w not in stop_words:
        filtered_sent.append(w)
print("Tokenized Sentence:",tokens)
print("Filterd Sentence:",filtered_sent)
```

```
Tokenized Sentence: ['Online', 'activities',
'such', 'as', 'articles', ',', 'website', 'text',
',', 'blog', 'posts', ',', 'social', 'media',
'posts', 'are', 'generating', 'unstructured',
'textual', 'data', '.', 'Corporate', 'and',
'business', 'need', 'to', 'analyze', 'textual',
'data'.']
```



```
Filterd Sentence: ['Online', 'activities',
'articles', ',', 'website', 'text', ',', 'blog',
'posts', ',', 'social', 'media', 'posts',
'generating', 'unstructured', 'textual', 'data',
'.',
```


https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html

Text Processing Operations (cont.)

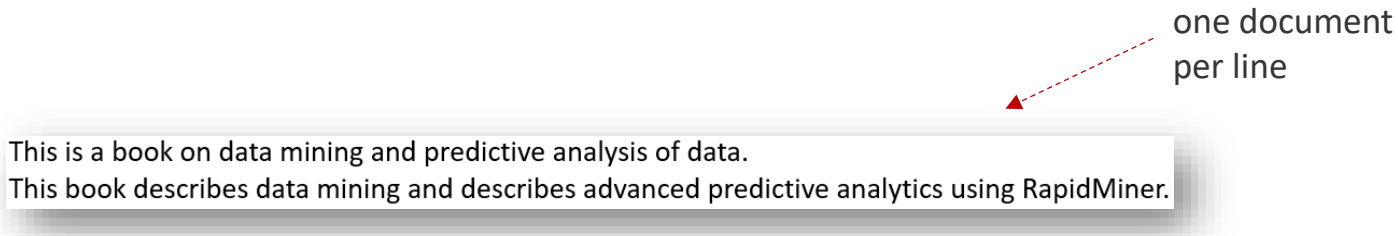
- **Noise Removal:** The operation is *depending on the specific text analytics problem*.
  - Assume that we obtained a text corpus from the world wide web, and that it is housed in a raw web format.
  - Text could be wrapped in HTML or XML tags.
  - Noise removal operation may include:
    - remove *text file headers*, *footers*
    - remove HTML, XML, etc. *markup* and *metadata*
    - extract valuable *data from other formats* (eg. JSON)

	raw_word	cleaned_word
0	..trouble..	trouble
1	trouble<	trouble
2	trouble!	trouble
3	<a>trouble	trouble
4	1.trouble	trouble

```
token = RegexpTokenizer(r'[a-zA-Z0-9]+')
```

https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html

Text Representation



term occurrence

advanc	analysi	analyt	book	data	describ	predict	rapidmin	use
0	1	0	1	2	0	1	0	0
1	0	1	1	1	2	1	1	1

Bag-of-words Representation

```
from sklearn.feature_extraction.text import CountVectorizer

CountVectorizer(lowercase=True,stop_words='english',ngram_range = (1,1),tokenizer = token.tokenize)
```


https://www.sketchengine.eu/pos-tags/

Discussion

What text processing operations were applied?

Review: "I recently purchased this laptop, and I am extremely satisfied with its performance. The processing speed is impressive, and it handles multitasking effortlessly.“

Processed Review: ["recent", "purchas", "laptop", "extrem", "satisfi", "perform", "process", "speed", "impress", "handl", "multitask", "effortless"]

https://www.sketchengine.eu/pos-tags/

Outline

- Text Analytics Framework
- **Text Classification**
- Sentiment Analysis
  - Lexicon-based Sentiment Analysis
  - Aspect-level Sentiment Analysis

Data Set for Text Classification

We will use "*Womens Clothing E-Commerce Reviews data set*", for demonstration of Text Classification steps.

Business Problem:
Evaluate overall opinion of customer toward women clothing products.

Analytic Goal:
Construct a model to automatically summarize customer opinions reflected in their review comments on various platforms.

Problem:
Opinion is estimated from textual data.

Approach:
Construct A Text Classification Model

```
import pandas as pd

df = pd.read_csv('ClothingReviews.csv')
df.head()
```

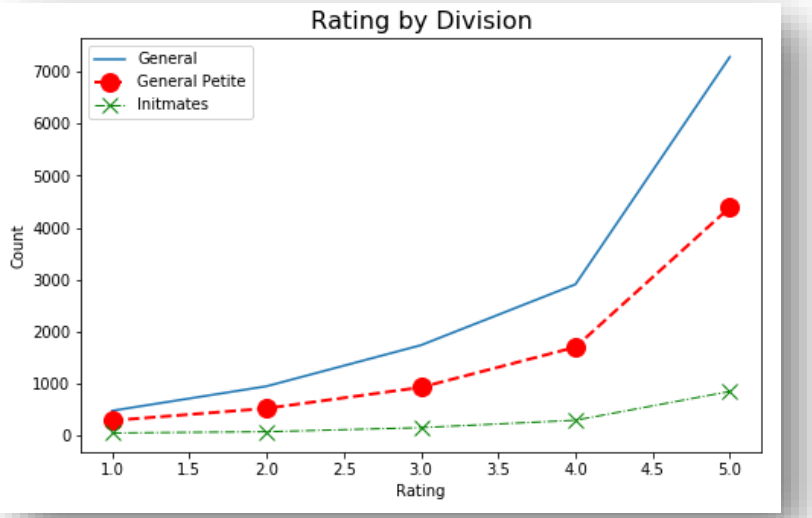
Label

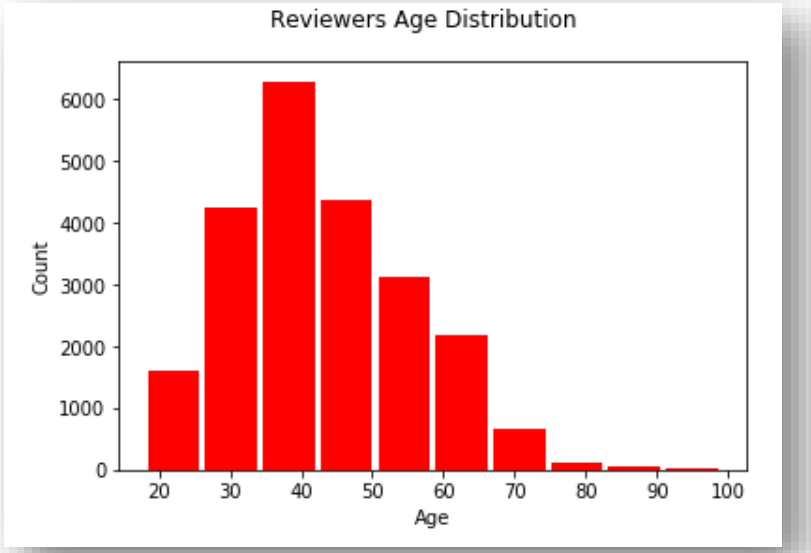
	Unnamed: 0	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
0	0	767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	Intimates
1	1	1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	Dresses
2	2	1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	Dresses	Dresses

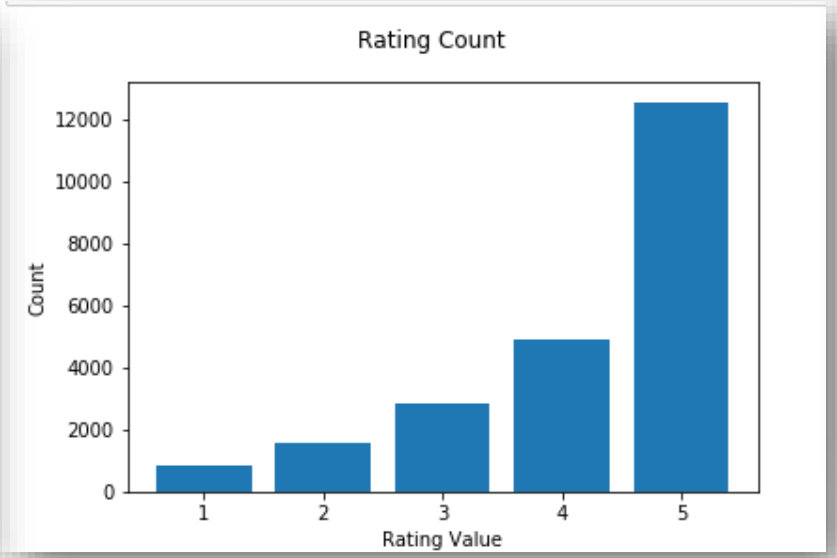
https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews

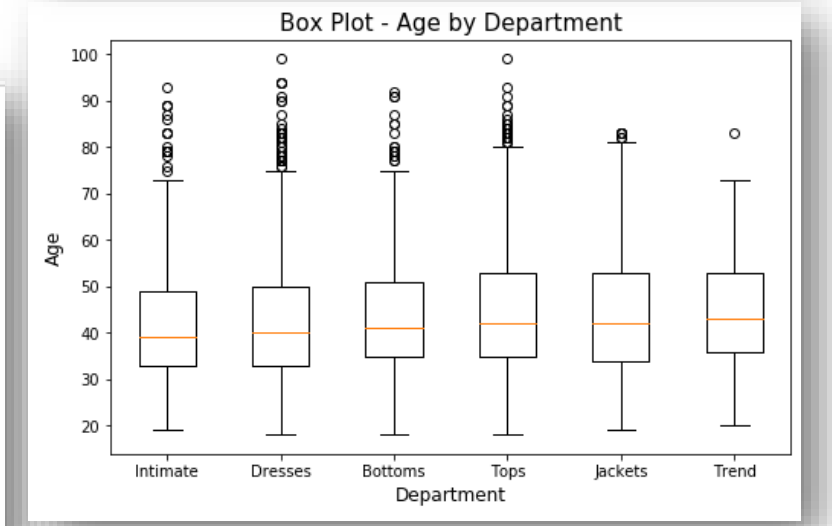
Exploration and Visualization

Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	Intimates
1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	Dresses









Text Features

Bag-of-words model (BoW) is the simplest way of extracting features from the text, which converts text into the matrix of occurrence of words within a document.

	about	bird	heard	is	the	word	you
About the bird, the bird, bird bird bird	1	5	0	0	2	0	0
You heard about the bird	1	1	1	0	1	0	1
The bird is the word	0	1	0	1	2	1	0

BoW Example
(Term Frequency)

```
from sklearn.feature_extraction.text import CountVectorizer
from nltk.tokenize import RegexpTokenizer

#tokenizer to remove unwanted elements from out data like symbols and
token = RegexpTokenizer(r'[a-zA-Z0-9]+')
cv = CountVectorizer(lowercase=True,stop_words='english',
                     ngram_range = (1,1),tokenizer = token.tokenize)

text_counts = cv.fit_transform(df['Review Text'])
print('Bag of Word Matrix size: ', text_counts.shape)
print('Data in the first row:\n', text_counts[1,:])
```

```
Bag of Word Matrix size:  (22641, 13875)
Data in the first row:
   (0, 12695)    1
  (0, 7679)      1
  (0, 12693)     1
  (0, 3538)      1
  (0, 6760)      1
  (0, 7144)      1
```

sparse matrix is represented as a Coordinate List (COO)
(see next slide)

Representation of Sparse Matrices

Full Matrix Format
(5x6 = 30 values)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix}$$

Sparse Matrix Format
(7x3 = 21 values)

Rows	Columns	Values
5	6	6
0	4	9
1	1	8
2	0	4
2	3	2
3	5	5
4	2	2

```
Bag of Word Matrix size:   (22641, 13875)
Data in the first row:
   (0, 12695)    1
  (0, 7679)      1
  (0, 12693)     1
  (0, 3538)      1
  (0, 6760)      1
  (0, 7144)      1
```

row column value

Sparse Matrix format is useful when the matrix has most of the elements are zero (to save memory storage)

Coordinate List (COO) only stores the row index, column index and the value of each non-zero element

Model Construction

Split the data for hold-out test:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    text_counts, df['Recommended IND'], test_size=0.3, random_state=1)
```

Train and evaluate Naive Bayes classifier:

```
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics

# Model Generation Using Multinomial Naive Bayes
clf = MultinomialNB().fit(X_train, y_train)
predicted= clf.predict(X_test)

print("MultinomialNB Accuracy:", round(metrics.accuracy_score(y_test, predicted),3))
print("Confusion Matrix:\n",metrics.confusion_matrix(y_test, predicted))
```

```
MultinomialNB Accuracy: 0.892
Confusion Matrix:
 [[ 768  438]
 [ 299 5288]]
```

	Predict 0 (Not Recommend)	Predict 1 (Recommend)
Actual 0 (Not Recommend)	768 (TN) (72%)	438 (FP) (8%)
Actual 1 (Recommend)	299 (FN) (28%)	5288 (TP) (92%)

Discussion Question

BoW Example
(Term Frequency)

	about	bird	heard	is	the	word	you
About the bird, the bird, bird bird bird	1	5	0	0	2	0	0
You heard about the bird	1	1	1	0	1	0	1
The bird is the word	0	1	0	1	2	1	0

```
Bag of Word Matrix size:  (22641, 13875)
```

Can we replace MultinominalNB by other classifiers (e.g., Logistic Regression, Decision Tree) ?

Dimensionality Reduction

The bag of word representations produces tables with high dimension (e.g., *not working well with most traditional machine learning techniques, logistic regression, decision tree*). This problem can be addressed by dimensionality reduction.

Univariate Selection:

- select features having the strongest relationships with the output variable based on statistical test

Recursive Feature Elimination:

- recursively removing attributes and building a model on remaining attributes.
- uses accuracy metric to rank the feature according to their importance

Principle Component Analysis:

- uses linear algebra to transform the dataset into a compressed form

Original Data set

```
Bag of Word Matrix size:  (22641, 13875)
```

Reduced Data Set based on Univariate Selection

```
from sklearn.feature_selection import SelectKBest

#Get the target label
Target = df['Recommended IND']
#We will select the top 100 features
test = SelectKBest(k=100)
#Fit the function for ranking the features by score
fit = test.fit(text_counts, Target)
UnivariateFeatures = fit.transform(text_counts)
print('Reduced Data Set size:',UnivariateFeatures.shape)
```

```
Reduced Data Set size: (22641, 100)
```

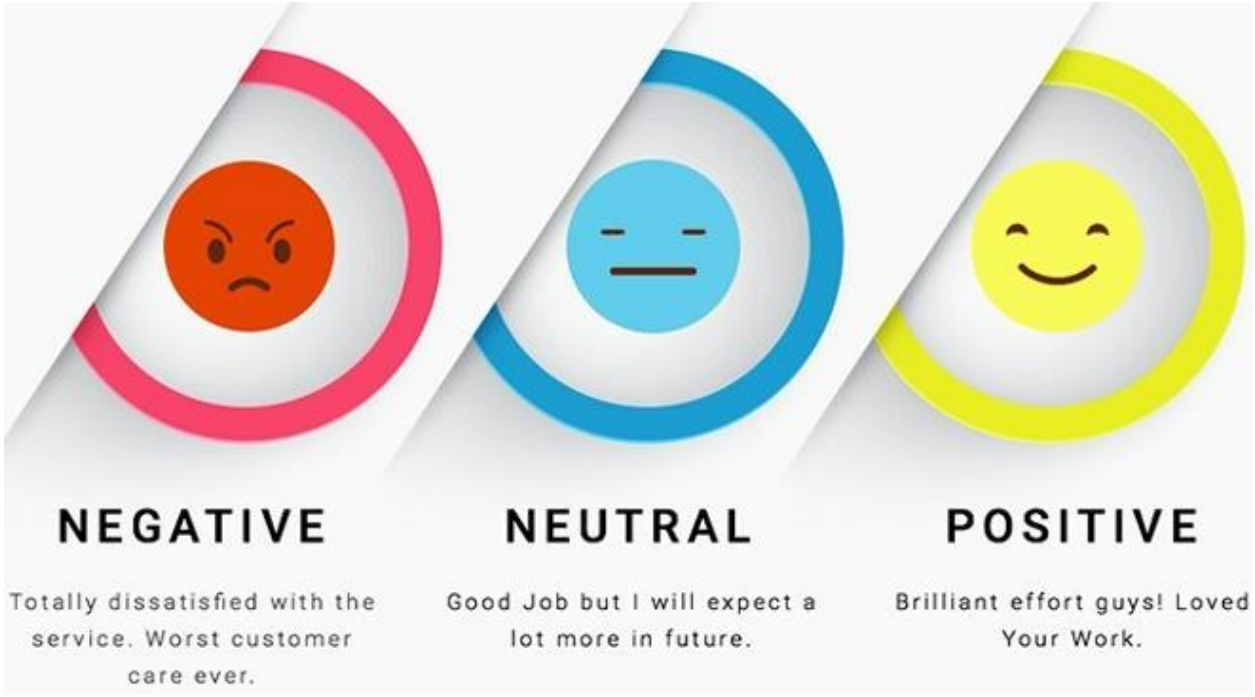
Outline

- Text Analytics Framework
- Text Classification
- **Sentiment Analysis**
  - **Lexicon-based Sentiment Analysis**
  - **Aspect-level Sentiment Analysis**

Sentiment Analysis

Sentiment analysis:

- Computational study of opinions, ***sentiments***, ***evaluations***, ***attitudes***, ***appraisal***, ***affects***, ***views***, ***emotions***, ***subjectivity***, etc., expressed in text.
  - Text = reviews, blogs, discussions, news, comments, feedback, etc.
- Sometimes referred to as **Opinion Mining**



https://www.kdnuggets.com/2018/03/5-things-sentiment-analysis-classification.html

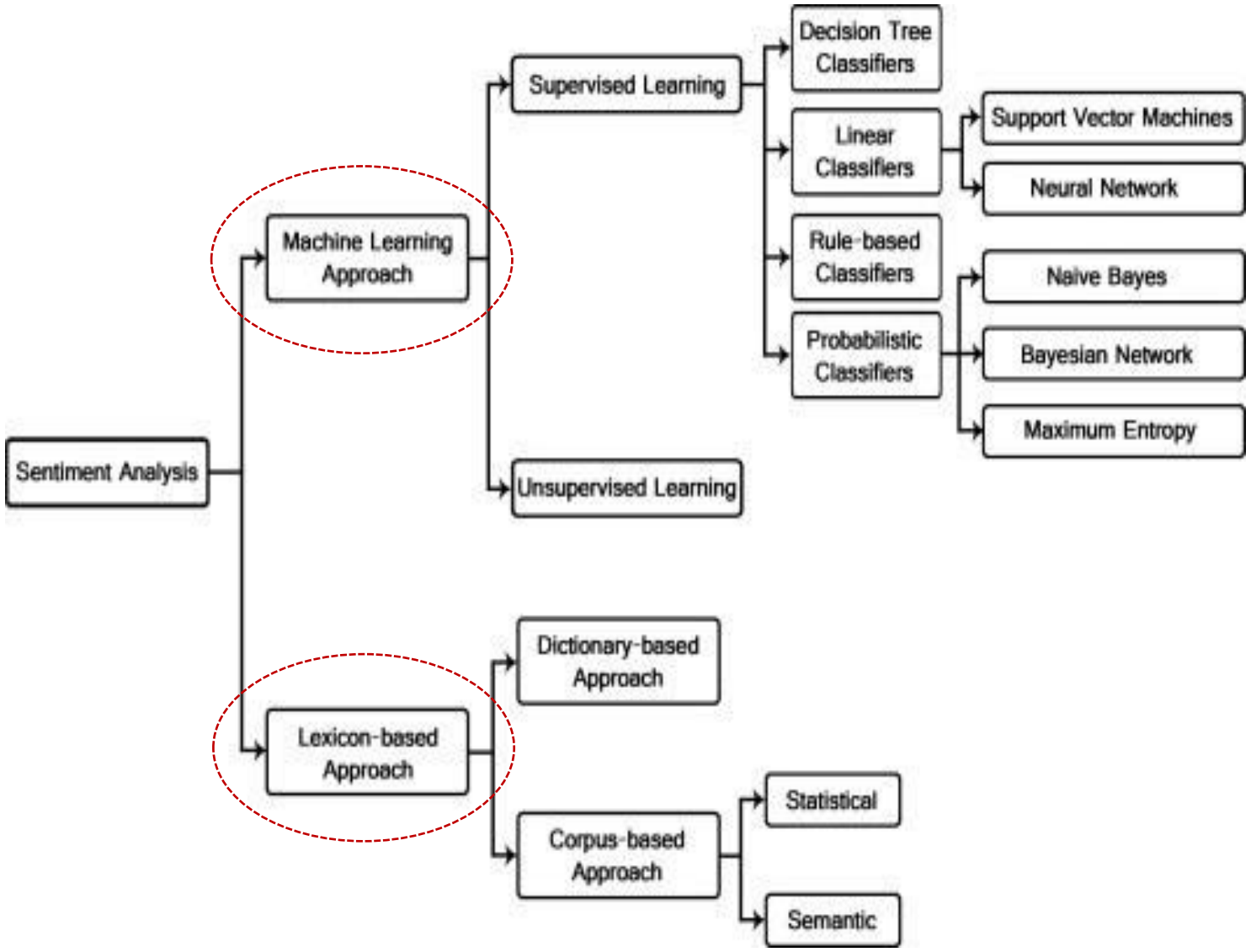
Discussion Question

Can the text classification task be treated as sentiment analysis?

Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	Intimates
1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	Dresses
1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	Dresses	Dresses

What if the label is not available?

Sentiment Analysis Techniques



Machine Learning:

- Employs a machine-learning technique and diverse features to construct a classifier that can identify text that expresses sentiment. (eg. **Text Classification** in the previous slides)

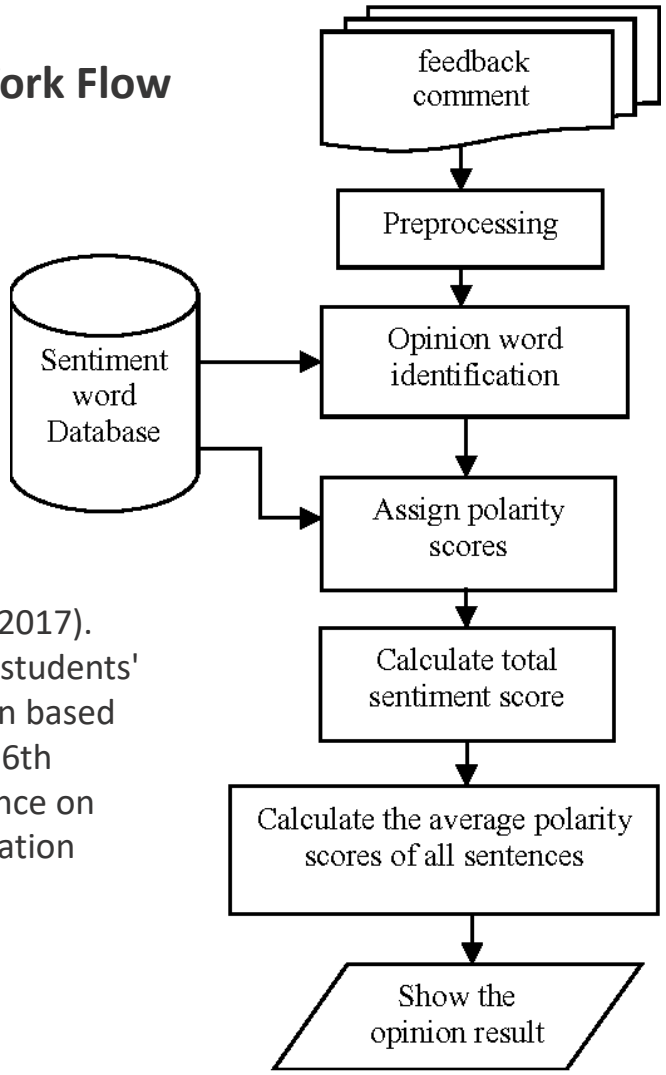
Lexicon-Based:

- uses a variety of words annotated by polarity score, to decide the general assessment score of a given content
- ***Pros***: does not require any training
- ***Cons***: large number of words and expressions are not included in sentiment lexicons data

https://www.kdnuggets.com/2018/03/5-things-sentiment-analysis-classification.html

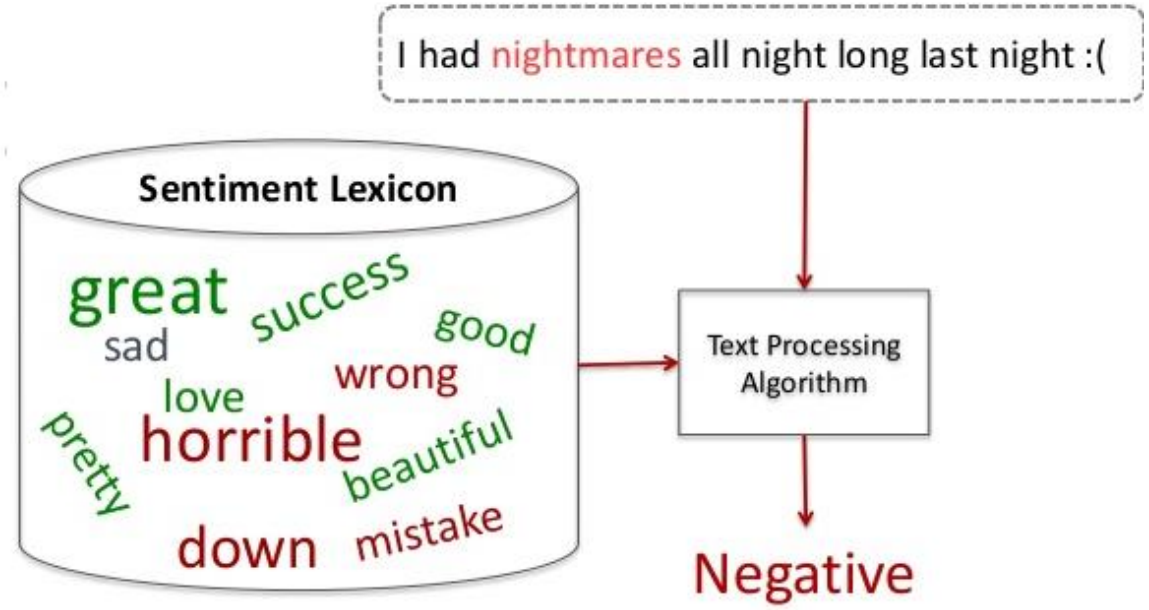
Lexicon-based Sentiment Analysis

Overall Work Flow



Aung, K.Z., My, N. N. (2017). Sentiment analysis of students' comment using lexicon based approach. IEEE/ACIS 16th International Conference on Computer and Information Science .

A Simple Example



https://www.slideshare.net/Staano/senticircles-for-contextual-and-conceptual-semantic-sentiment-analysis-of-twitter

Demo

Lexicon-based Sentiment Analysis (cont.)

The **nltk.sentiment** library provides **SentimentIntensityAnalyzer** module, which allows for estimating sentiment directly from natural text.

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
#Initialize an instance of SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()
message_text = df['Review Text'][1]
print('Review Comment:\n', message_text)
```

```
Review Comment:
 Love this dress!  it's sooo pretty.  i happened to find it in a store, and i'm glad i did bc i ne
ver would have ordered it online bc it's petite.  i bought a petite and am 5'8".  i love the lengt
h on me- hits just a little below the knee.  would definitely be a true midi on someone who is tru
ly petite.
```

```
#Estimate sentiment scores
scores = sid.polarity_scores(message_text)
for key in sorted(scores):
        print('{0}: {1} \n'.format(key, scores[key]), end='')
print('True Recommedation Label was: ', df['Recommended IND'][1])
```

```
compound: 0.9729
neg: 0.0
neu: 0.664
pos: 0.336
True Recommedation Label was:  1
```

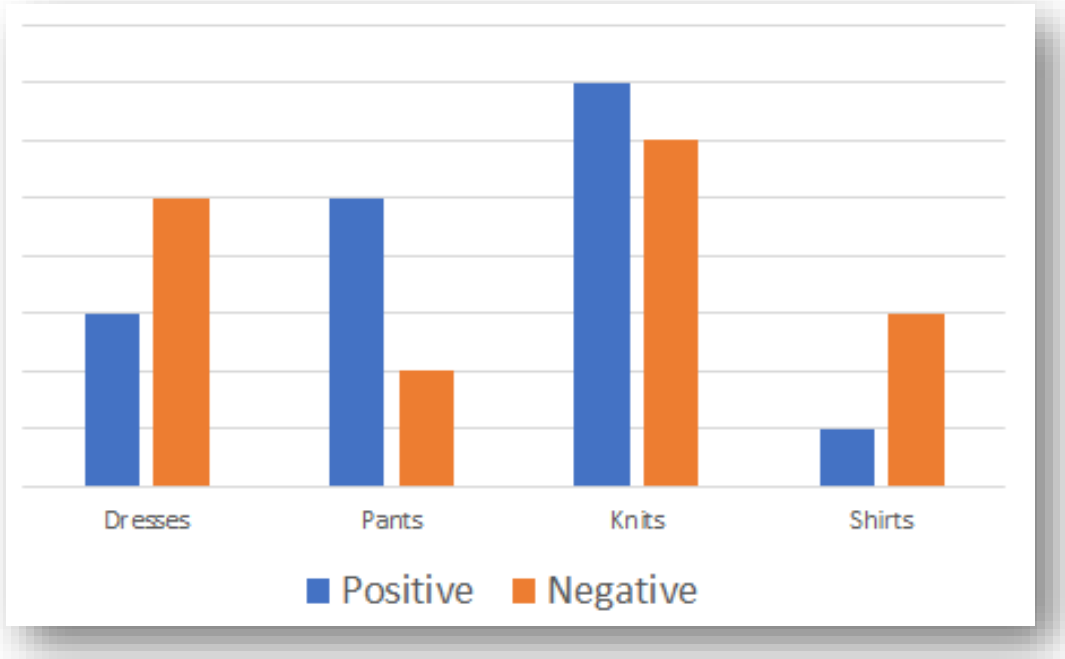
Estimated Sentiment based on Lexicon-Based approach is correct as Positive

Discussion Question

Now, we obtained sentiment labels (positive/negative) of products.
So what can business mangers do with this?

Review Text	Sentiment
Absolutely wonderful - silky and sexy and	1
Love this dress! it's sooo pretty. i happen	1
I had such high hopes for this dress and re	0
I love, love, love this jumpsuit. it's fun, flirt	1
This shirt is very flattering to all due to the	1
I love tracy reese dresses, but this one is n	0
I aded this in my basket at hte last mintue	1
I ordered this in carbon for store pick up, a	1
I love this dress. i usually get an xs but it ru	1
I'm 5"5' and 125 lbs. i ordered the s petite	1
Dress runs small esp where the zipper area	0

Assessing Customer Satisfaction of Products

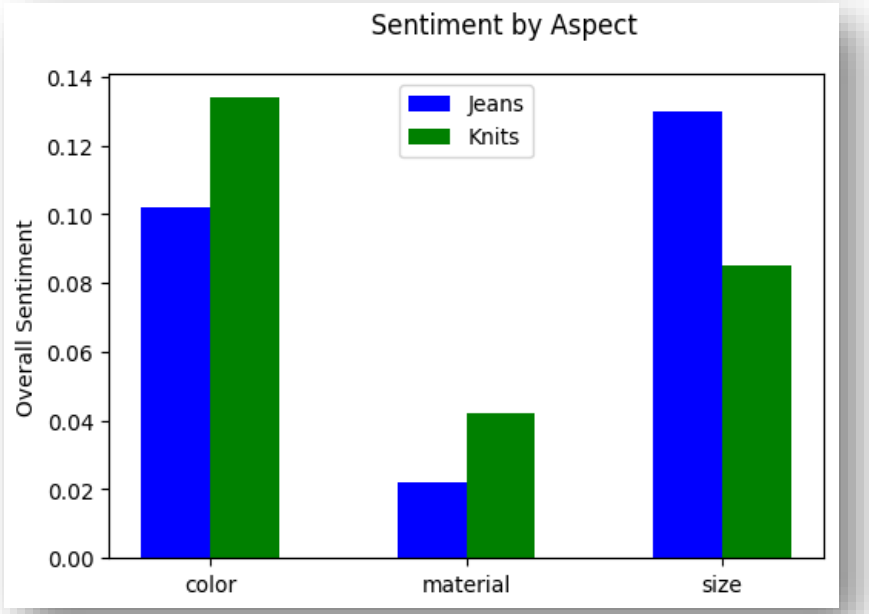


Discussion Question

In addition to the overall sentiment, what extra insights can we obtain from the review comments?

'the color is really nice charcoal with shimmer, and went well with pencil skirts, flare pants, etc. the material feels very cheap and disapointing. on it will cause it to rip the fabric. pretty disappointed as it was going to be my christmas dress this year! needless to say it will be going back. Material is bad, the size is verys mall and uncomfortable. with a leg opening the size of my waist and hem line above my ankle, and front pleats to make me fluffy, i think you can imagine that it is not a flattering look. In generall, Hate the material, Love the color '

Product Features (Aspects)



Aspect-Level Sentiment Analysis in Python

```
SAMPLE REVIEW:
 the color is really nice charcoal with shimmer, and went well with pencil skirts,flare pants, et
c. the material feels very cheap and disapointing. on it will cause it to rip the fabric. pretty d
isappointed as it was going to be my christmas dress this year! needless to say it will be going b
ack. Material is bad, the size is verys mall and uncomfortable. with a leg opening the size of my
waist and hem line above my ankle, and front pleats to make me fluffy,i think you can imagine that
it is not a flattering look. In generall, Hate the material, Love the color.
OVERALL SENTIMENT SCORE:
 {'neg': 0.126, 'neu': 0.77, 'pos': 0.104, 'compound': -0.3738}
```

Document-Level Sentiment Score

```
#Compute Sentiment Score by text trunk
from nltk.tokenize import sent_tokenize

ReviewComment=ReviewComment.replace(',','.')
sents =  sent_tokenize(ReviewComment)
scores = []
for s in range(len(sents)):
       scores.append(sid.polarity_scores(sents[s]))
[s for s in scores] #Show sentiment score of individual trunks
```

Split text into sentences and compute sentiment score

```
[{'neg': 0.0, 'neu': 0.694, 'pos': 0.306, 'compound': 0.4754},
 {'neg': 0.0, 'neu': 0.741, 'pos': 0.259, 'compound': 0.2732},
 {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0},
 {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0},
```

Sentence-Level Sentiment Score

Aspect-Level Sentiment Analysis in Python (cont.)

```
from statistics import mean
#Define function to compute sentiment score for aspect
def aspect_sentiment(aspect,sents,scores):
    AspSen = []
    for s in range(len(sents)):
        #Check if the aspect is mentioned in the text trunk
        Index = sents[s].find(aspect)
        if Index > 0:
            AspSen.append(scores[s])
    Sentiment = [AspSen[x]['compound'] for x in range(len(AspSen))]

    #Return average of sentiment scores of aspect
    return [round(mean(Sentiment),3)]

print('color: ', aspect_sentiment('color',sents,scores))
print('material: ', aspect_sentiment('material',sents,scores))
print('size: ', aspect_sentiment('size',sents,scores))
```

```
color:  [0.556]
material:  [-0.286]
size:  [-0.191]
```

Aspect-Level Sentiment Score

Aggregate sentiment score for all sentences with respect to individual features

Business Application:
Obtain an overview about customer sentiments toward specific product features for product improvement (negative features) or promotion (positive features).

Discussion Question

How to determine what aspects (e.g., color, size of a product) are available in the data set?

```
the color is really nice charcoal with
shimmer, and went well with pencil skirts
```

- Identify Nouns terms. (How?)
- Compute the popularity of noun terms.
- Select noun terms describing product aspects based on analysts understanding about the business domain.

Part of Speech Tagging

- **Part of Speech Tagging (POS) :** is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech

Tag set example

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	*+,%, &*
CD	Cardinal number	*one, two, three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WPS	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	*$*
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	*#*
PDT	Predeterminer	*all, both*	"	Left quote	(' or ")
POS	Possessive ending	*'s*	"	Right quote	(' or ")
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one's*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

```
text = nltk.word_tokenize("the color is really nice charcoal with shimmer, and went well with pencil skirts")
nltk.pos_tag(text)
```

```
[('the', 'DT'),
('color', 'NN'),
('is', 'VBZ'),
('really', 'RB'),
('nice', 'JJ'),
('charcoal', 'NN'),
('with', 'IN'),
('shimmer', 'NN'),
('and', 'CC'),
('went', 'VBD'),
('well', 'RB'),
('with', 'IN'),
('pencil', 'NN'),
('skirts', 'NNS')]
```

https://www.sketchengine.eu/pos-tags/

In this lecture, we have covered:

- Overall view Text Analytic Framework and Tex Processing Operations
- Text classification Procedure in Python
- Sentiment Analysis based on Lexical Approach
- Sentiment Analysis at Aspect Level
- Sentiment Summarization

Summary