Assignment 7 - Text Analytics

Kaustubh Shrikant Kabra

ERP Number :- 38

TE Comp 1

Text Analytics

1. Extract Sample document and apply following document preprocessing methods: Tokenization, POS Tagging, stop words removal, Stemming and Lemmatization.
2. Create representation of document by calculating Term Frequency and Inverse Document Frequency

In [1]:

```
import pandas as pd
```

In [2]:

```
text = '''It was a Thursday, but it felt like a Monday to John. And John loved Mondays.

I should probably get another latte. I’ve just been sitting here with this empty cup. B

John was always impatient on the weekends; he missed the formal structure of the busine

Jesus, I’ve written another loser. '''
```

Tokenization of text

In [3]:

```
text_split = text.split()
```

In [4]:

```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\asus\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\asus\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\asus\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

Out[4]:

```
True
```

In [5]:

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
stop_words = stopwords.words('english')
```

In [6]:

```
#stop_words
```

In [7]:

```
tokenized = sent_tokenize(text)
for i in tokenized:

    wordsList = nltk.word_tokenize(i)

    # removing stop words from wordList
    wordsList = [w for w in wordsList if not w in stop_words]

    # Using a Tagger. Which is part-of-speech
    # tagger or POS-tagger.
    tagged = nltk.pos_tag(wordsList)

    print(tagged)
```

```
[('It', 'PRP'), ('Thursday', 'NNP'), (',', ','), ('felt', 'VBD'), ('like', 'IN'), ('Mond
ay', 'NNP'), ('John', 'NNP'), ('.', '.')]
[('And', 'CC'), ('John', 'NNP'), ('loved', 'VBD'), ('Mondays', 'NNP'), ('.', '.')]
[('He', 'PRP'), ('thrived', 'VBD'), ('work', 'NN'), ('.', '.')]
[('He', 'PRP'), ('dismissed', 'VBD'), ('old', 'JJ'), ('cliché', 'NN'), ('dreading', 'VB
G'), ('Monday', 'NNP'), ('mornings', 'NNS'), ('refused', 'VBD'), ('engage', 'JJ'), ('wat
er-cooler', 'JJ'), ('complaints', 'NNS'), ('“', 'JJ'), ('grind', 'VBP'), ('”', 'JJ'),
('empty', 'JJ'), ('conversations', 'NNS'), ('included', 'VBD'), ('familiar', 'JJ'), ('pa
rry', 'NN'), ('“', 'NNP'), ('How', 'NNP'), ('weekend', 'NN'), ('?', '.'), ('”', 'JJ'),
('“', 'NNP'), ('Too', 'NNP'), ('short', 'JJ'), ('!', '.'), ('”', 'NN'), ('.', '.')]
[('Yes', 'UH'), (',', ','), ('John', 'NNP'), ('liked', 'VBD'), ('work', 'NN'), ('unasham
ed', 'NN'), ('.', '.')]
[('I', 'PRP'), ('probably', 'RB'), ('get', 'VB'), ('another', 'DT'), ('latte', 'NN'),
('.', '.')]
[('I', 'PRP'), ('’', 'VBP'), ('sitting', 'VBG'), ('empty', 'JJ'), ('cup', 'NN'), ('.',
'.')]
[('But', 'CC'), ('I', 'PRP'), ('’', 'VBP'), ('start', 'JJ'), ('get', 'VB'), ('jittery',
'NN'), ('.', '.')]
[('I', 'PRP'), ('’', 'VBP'), ('get', 'VB'), ('decaf', 'NN'), ('.', '.')]
[('No', 'DT'), (',', ','), ('’', 'FW'), ('stupid', 'JJ'), (',', ','), ('feels', 'JJ'),
('stupid', 'JJ'), ('pay', 'NN'), ('decaf', 'NN'), ('.', '.')]
[('I', 'PRP'), ('’', 'VBP'), ('justify', 'NN'), ('.', '.')]
[('John', 'NNP'), ('always', 'RB'), ('impatient', 'JJ'), ('weekends', 'NNS'), (';',
':'), ('missed', 'VBN'), ('formal', 'JJ'), ('structure', 'NN'), ('business', 'NN'), ('we
ek', 'NN'), ('.', '.')]
[('When', 'WRB'), ('younger', 'JJR'), ('used', 'VBD'), ('stay', 'NN'), ('late', 'JJ'),
('school', 'NN'), ('Fridays', 'NNP'), ('come', 'VBP'), ('early', 'JJ'), ('Mondays', 'NN
P'), (',', ','), ('pattern', 'NN'), ('mother', 'NN'), ('referred', 'VBD'), ('equal', 'J
J'), ('parts', 'NNS'), ('admiration', 'NN'), ('disdain', 'VBP'), ('“', 'JJ'), ('studyin
g', 'VBG'), ('overtime.', 'JJ'), ('”', 'NNP'), ('Jesus', 'NNP'), (',', ','), ('I', 'PR
P'), ('’', 'VBP'), ('written', 'VBN'), ('another', 'DT'), ('loser', 'NN'), ('.', '.')]
```

In [8]:

```
tokenized
```

Out[8]:

```
['It was a Thursday, but it felt like a Monday to John.',
 'And John loved Mondays.',
 'He thrived at work.',
 'He dismissed the old cliché of dreading Monday mornings and refused to engage in water
-cooler complaints about “the grind” and empty conversations that included the familiar
parry “How was your weekend?” “Too short!”.',
 'Yes, John liked his work and was unashamed.',
 'I should probably get another latte.',
 'I’ve just been sitting here with this empty cup.',
 'But then I’ll start to get jittery.',
 'I’ll get a decaf.',
 'No, that’s stupid, it feels stupid to pay for a decaf.',
 'I can’t justify that.',
 'John was always impatient on the weekends; he missed the formal structure of the busin
ess week.',
 'When he was younger he used to stay late after school on Fridays and come in early on
Mondays, a pattern his mother referred to with equal parts admiration and disdain as “st
udying overtime.”\n\nJesus, I’ve written another loser.']
```

Stemming and Lemmatization

In [9]:

```python
from nltk.stem.porter import PorterStemmer
```

In [10]:

```python
porter_stemmer = PorterStemmer()
```

In [11]:

```python
nltk_token = nltk.word_tokenize(text)
```

In [12]:

```python
for w in nltk_token:
  print("Actual : %s , Stem: %s" %(w, porter_stemmer.stem(w)))
```

```
Actual : It , Stem: it
Actual : was , Stem: wa
Actual : a , Stem: a
Actual : Thursday , Stem: thursday
Actual : , , Stem: ,
Actual : but , Stem: but
Actual : it , Stem: it
Actual : felt , Stem: felt
Actual : like , Stem: like
Actual : a , Stem: a
Actual : Monday , Stem: monday
Actual : to , Stem: to
Actual : John , Stem: john
Actual : . , Stem: .
Actual : And , Stem: and
Actual : John , Stem: john
Actual : loved , Stem: love
Actual : Mondays , Stem: monday
Actual : . , Stem: .
Actual : He , Stem: he
Actual : thrived , Stem: thrive
Actual : at , Stem: at
Actual : work , Stem: work
Actual : . , Stem: .
Actual : He , Stem: he
Actual : dismissed , Stem: dismiss
Actual : the , Stem: the
Actual : old , Stem: old
Actual : cliché , Stem: cliché
Actual : of , Stem: of
Actual : dreading , Stem: dread
Actual : Monday , Stem: monday
Actual : mornings , Stem: morn
Actual : and , Stem: and
Actual : refused , Stem: refus
Actual : to , Stem: to
Actual : engage , Stem: engag
Actual : in , Stem: in
Actual : water-cooler , Stem: water-cool
Actual : complaints , Stem: complaint
Actual : about , Stem: about
Actual : “ , Stem: “
Actual : the , Stem: the
Actual : grind , Stem: grind
Actual : ” , Stem: ”
Actual : and , Stem: and
Actual : empty , Stem: empti
Actual : conversations , Stem: convers
Actual : that , Stem: that
Actual : included , Stem: includ
Actual : the , Stem: the
Actual : familiar , Stem: familiar
Actual : parry , Stem: parri
Actual : “ , Stem: “
Actual : How , Stem: how
Actual : was , Stem: wa
Actual : your , Stem: your
Actual : weekend , Stem: weekend
Actual : ? , Stem: ?
Actual : ” , Stem: ”
Actual : “ , Stem: “
Actual : Too , Stem: too
Actual : short , Stem: short
Actual : ! , Stem: !
Actual : ” , Stem: ”
Actual : . , Stem: .
Actual : Yes , Stem: ye
Actual : , , Stem: ,
Actual : John , Stem: john
Actual : liked , Stem: like
Actual : his , Stem: hi
Actual : work , Stem: work
Actual : and , Stem: and
Actual : was , Stem: wa
Actual : unashamed , Stem: unasham
Actual : . , Stem: .
Actual : I , Stem: i
Actual : should , Stem: should
Actual : probably , Stem: probabl
Actual : get , Stem: get
Actual : another , Stem: anoth
Actual : latte , Stem: latt
Actual : . , Stem: .
Actual : I , Stem: i
Actual : ’ , Stem: ’
Actual : ve , Stem: ve
Actual : just , Stem: just
Actual : been , Stem: been
Actual : sitting , Stem: sit
Actual : here , Stem: here
Actual : with , Stem: with
Actual : this , Stem: thi
Actual : empty , Stem: empti
Actual : cup , Stem: cup
Actual : . , Stem: .
Actual : But , Stem: but
Actual : then , Stem: then
Actual : I , Stem: i
Actual : ’ , Stem: ’
Actual : ll , Stem: ll
Actual : start , Stem: start
Actual : to , Stem: to
Actual : get , Stem: get
Actual : jittery , Stem: jitteri
Actual : . , Stem: .
Actual : I , Stem: i
Actual : ’ , Stem: ’
Actual : ll , Stem: ll
Actual : get , Stem: get
Actual : a , Stem: a
Actual : decaf , Stem: decaf
Actual : . , Stem: .
Actual : No , Stem: no
Actual : , , Stem: ,
Actual : that , Stem: that
Actual : ’ , Stem: ’
Actual : s , Stem: s
Actual : stupid , Stem: stupid
Actual : , , Stem: ,
Actual : it , Stem: it
Actual : feels , Stem: feel
Actual : stupid , Stem: stupid
Actual : to , Stem: to
Actual : pay , Stem: pay
Actual : for , Stem: for
Actual : a , Stem: a
Actual : decaf , Stem: decaf
Actual : . , Stem: .
Actual : I , Stem: i
Actual : can , Stem: can
Actual : ’ , Stem: ’
Actual : t , Stem: t
Actual : justify , Stem: justifi
Actual : that , Stem: that
Actual : . , Stem: .
Actual : John , Stem: john
Actual : was , Stem: wa
Actual : always , Stem: alway
Actual : impatient , Stem: impati
Actual : on , Stem: on
Actual : the , Stem: the
Actual : weekends , Stem: weekend
Actual : ; , Stem: ;
Actual : he , Stem: he
Actual : missed , Stem: miss
Actual : the , Stem: the
Actual : formal , Stem: formal
Actual : structure , Stem: structur
Actual : of , Stem: of
Actual : the , Stem: the
Actual : business , Stem: busi
Actual : week , Stem: week
Actual : . , Stem: .
Actual : When , Stem: when
Actual : he , Stem: he
Actual : was , Stem: wa
Actual : younger , Stem: younger
Actual : he , Stem: he
Actual : used , Stem: use
Actual : to , Stem: to
Actual : stay , Stem: stay
Actual : late , Stem: late
Actual : after , Stem: after
Actual : school , Stem: school
Actual : on , Stem: on
Actual : Fridays , Stem: friday
Actual : and , Stem: and
Actual : come , Stem: come
Actual : in , Stem: in
Actual : early , Stem: earli
Actual : on , Stem: on
Actual : Mondays , Stem: monday
Actual : , , Stem: ,
Actual : a , Stem: a
Actual : pattern , Stem: pattern
Actual : his , Stem: hi
Actual : mother , Stem: mother
Actual : referred , Stem: refer
Actual : to , Stem: to
Actual : with , Stem: with
Actual : equal , Stem: equal
Actual : parts , Stem: part
Actual : admiration , Stem: admir
Actual : and , Stem: and
Actual : disdain , Stem: disdain
Actual : as , Stem: as
Actual : “ , Stem: “
Actual : studying , Stem: studi
Actual : overtime. , Stem: overtime.
Actual : ” , Stem: ”
Actual : Jesus , Stem: jesu
Actual : , , Stem: ,
Actual : I , Stem: i
Actual : ’ , Stem: ’
Actual : ve , Stem: ve
Actual : written , Stem: written
Actual : another , Stem: anoth
Actual : loser , Stem: loser
Actual : . , Stem: .
```

Lemmatization

In [13]:

```
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
```

In [14]:

```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
```

```
[nltk_data]     C:\Users\ORIONORIGINAL\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\wordnet.zip.
```

Out[14]:

```
True
```

In [15]:

```
for w in nltk_token:
  print("Actual : %s , Lemme: %s" %(w, wordnet_lemmatizer.lemmatize(w)))
```

```
Actual : It , Lemme: It
Actual : was , Lemme: wa
Actual : a , Lemme: a
Actual : Thursday , Lemme: Thursday
Actual : , , Lemme: ,
Actual : but , Lemme: but
Actual : it , Lemme: it
Actual : felt , Lemme: felt
Actual : like , Lemme: like
Actual : a , Lemme: a
Actual : Monday , Lemme: Monday
Actual : to , Lemme: to
Actual : John , Lemme: John
Actual : . , Lemme: .
Actual : And , Lemme: And
Actual : John , Lemme: John
Actual : loved , Lemme: loved
Actual : Mondays , Lemme: Mondays
Actual : . , Lemme: .
Actual : He , Lemme: He
Actual : thrived , Lemme: thrived
Actual : at , Lemme: at
Actual : work , Lemme: work
Actual : . , Lemme: .
Actual : He , Lemme: He
Actual : dismissed , Lemme: dismissed
Actual : the , Lemme: the
Actual : old , Lemme: old
Actual : cliché , Lemme: cliché
Actual : of , Lemme: of
Actual : dreading , Lemme: dreading
Actual : Monday , Lemme: Monday
Actual : mornings , Lemme: morning
Actual : and , Lemme: and
Actual : refused , Lemme: refused
Actual : to , Lemme: to
Actual : engage , Lemme: engage
Actual : in , Lemme: in
Actual : water-cooler , Lemme: water-cooler
Actual : complaints , Lemme: complaint
Actual : about , Lemme: about
Actual : “ , Lemme: “
Actual : the , Lemme: the
Actual : grind , Lemme: grind
Actual : ” , Lemme: ”
Actual : and , Lemme: and
Actual : empty , Lemme: empty
Actual : conversations , Lemme: conversation
Actual : that , Lemme: that
Actual : included , Lemme: included
Actual : the , Lemme: the
Actual : familiar , Lemme: familiar
Actual : parry , Lemme: parry
Actual : “ , Lemme: “
Actual : How , Lemme: How
Actual : was , Lemme: wa
Actual : your , Lemme: your
Actual : weekend , Lemme: weekend
Actual : ? , Lemme: ?
Actual : ” , Lemme: ”
Actual : “ , Lemme: “
Actual : Too , Lemme: Too
Actual : short , Lemme: short
Actual : ! , Lemme: !
Actual : ” , Lemme: ”
Actual : . , Lemme: .
Actual : Yes , Lemme: Yes
Actual : , , Lemme: ,
Actual : John , Lemme: John
Actual : liked , Lemme: liked
Actual : his , Lemme: his
Actual : work , Lemme: work
Actual : and , Lemme: and
Actual : was , Lemme: wa
Actual : unashamed , Lemme: unashamed
Actual : . , Lemme: .
Actual : I , Lemme: I
Actual : should , Lemme: should
Actual : probably , Lemme: probably
Actual : get , Lemme: get
Actual : another , Lemme: another
Actual : latte , Lemme: latte
Actual : . , Lemme: .
Actual : I , Lemme: I
Actual : ’ , Lemme: ’
Actual : ve , Lemme: ve
Actual : just , Lemme: just
Actual : been , Lemme: been
Actual : sitting , Lemme: sitting
Actual : here , Lemme: here
Actual : with , Lemme: with
Actual : this , Lemme: this
Actual : empty , Lemme: empty
Actual : cup , Lemme: cup
Actual : . , Lemme: .
Actual : But , Lemme: But
Actual : then , Lemme: then
Actual : I , Lemme: I
Actual : ’ , Lemme: ’
Actual : ll , Lemme: ll
Actual : start , Lemme: start
Actual : to , Lemme: to
Actual : get , Lemme: get
Actual : jittery , Lemme: jittery
Actual : . , Lemme: .
Actual : I , Lemme: I
Actual : ’ , Lemme: ’
Actual : ll , Lemme: ll
Actual : get , Lemme: get
Actual : a , Lemme: a
Actual : decaf , Lemme: decaf
Actual : . , Lemme: .
Actual : No , Lemme: No
Actual : , , Lemme: ,
Actual : that , Lemme: that
Actual : ’ , Lemme: ’
Actual : s , Lemme: s
Actual : stupid , Lemme: stupid
Actual : , , Lemme: ,
Actual : it , Lemme: it
Actual : feels , Lemme: feel
Actual : stupid , Lemme: stupid
Actual : to , Lemme: to
Actual : pay , Lemme: pay
Actual : for , Lemme: for
Actual : a , Lemme: a
Actual : decaf , Lemme: decaf
Actual : . , Lemme: .
Actual : I , Lemme: I
Actual : can , Lemme: can
Actual : ’ , Lemme: ’
Actual : t , Lemme: t
Actual : justify , Lemme: justify
Actual : that , Lemme: that
Actual : . , Lemme: .
Actual : John , Lemme: John
Actual : was , Lemme: wa
Actual : always , Lemme: always
Actual : impatient , Lemme: impatient
Actual : on , Lemme: on
Actual : the , Lemme: the
Actual : weekends , Lemme: weekend
Actual : ; , Lemme: ;
Actual : he , Lemme: he
Actual : missed , Lemme: missed
Actual : the , Lemme: the
Actual : formal , Lemme: formal
Actual : structure , Lemme: structure
Actual : of , Lemme: of
Actual : the , Lemme: the
Actual : business , Lemme: business
Actual : week , Lemme: week
Actual : . , Lemme: .
Actual : When , Lemme: When
Actual : he , Lemme: he
Actual : was , Lemme: wa
Actual : younger , Lemme: younger
Actual : he , Lemme: he
Actual : used , Lemme: used
Actual : to , Lemme: to
Actual : stay , Lemme: stay
Actual : late , Lemme: late
Actual : after , Lemme: after
Actual : school , Lemme: school
Actual : on , Lemme: on
Actual : Fridays , Lemme: Fridays
Actual : and , Lemme: and
Actual : come , Lemme: come
Actual : in , Lemme: in
Actual : early , Lemme: early
Actual : on , Lemme: on
Actual : Mondays , Lemme: Mondays
Actual : , , Lemme: ,
Actual : a , Lemme: a
Actual : pattern , Lemme: pattern
Actual : his , Lemme: his
Actual : mother , Lemme: mother
Actual : referred , Lemme: referred
Actual : to , Lemme: to
Actual : with , Lemme: with
Actual : equal , Lemme: equal
Actual : parts , Lemme: part
Actual : admiration , Lemme: admiration
Actual : and , Lemme: and
Actual : disdain , Lemme: disdain
Actual : as , Lemme: a
Actual : “ , Lemme: “
Actual : studying , Lemme: studying
Actual : overtime. , Lemme: overtime.
Actual : ” , Lemme: ”
Actual : Jesus , Lemme: Jesus
Actual : , , Lemme: ,
Actual : I , Lemme: I
Actual : ’ , Lemme: ’
Actual : ve , Lemme: ve
Actual : written , Lemme: written
Actual : another , Lemme: another
Actual : loser , Lemme: loser
Actual : . , Lemme: .
```

2. Word count

Term Frequency (TF)

Formula: **tf(t,d)** = count of t in d / number of words in d

In [16]:

```
sentence1 = "Data Science is the best job of the 21st century"
sentence2 = "machine learning is the key for data science"
```

In [17]:

```
# Spliting both sentences
sentence1 = sentence1.split(" ")
sentence2 = sentence2.split(" ")
```

In [18]:

```
join = set(sentence1).union(set(sentence2))
```

In [19]:

```
join
```

Out[19]:

```
{'21st',
 'Data',
 'Science',
 'best',
 'century',
 'data',
 'for',
 'is',
 'job',
 'key',
 'learning',
 'machine',
 'of',
 'science',
 'the'}
```

In [20]:

```
wordDict1 = dict.fromkeys(join, 0)
wordDict2 = dict.fromkeys(join, 0)

for word in sentence1:
  wordDict1[word] += 1


for word in sentence2:
  wordDict2[word] += 1
```

In [21]:

```
pd.DataFrame([wordDict1, wordDict2])
```

Out[21]:

	the	is	learning	for	century	Science	job	key	of	data	science	machine	21st	best	Data
0	2	1	0	0	1	1	1	0	1	0	0	0	1	1	1
1	1	1	1	1	0	0	0	1	0	1	1	1	0	0	0

In [22]:

```
def getTF(wordDict, data):
  res = {}
  corpusCount = len(data)
  for word, count in wordDict.items():
    res[word] = count/float(corpusCount)
  return res

tf1 = getTF(wordDict1, sentence1)
tf2 = getTF(wordDict2, sentence2)
```

In [23]:

```
tf2
```

Out[23]:

```
{'the': 0.125,
 'is': 0.125,
 'learning': 0.125,
 'for': 0.125,
 'century': 0.0,
 'Science': 0.0,
 'job': 0.0,
 'key': 0.125,
 'of': 0.0,
 'data': 0.125,
 'science': 0.125,
 'machine': 0.125,
 '21st': 0.0,
 'best': 0.0,
 'Data': 0.0}
```

In [24]:

```
tf = pd.DataFrame([tf1, tf2])
```

In [25]:

```
tf
```

Out[25]:

	the	is	learning	for	century	Science	job	key	of	data	science	machine	21st	best
0	0.200	0.100	0.000	0.000	0.1	0.1	0.1	0.000	0.1	0.000	0.000	0.000	0.1	0.1
1	0.125	0.125	0.125	0.125	0.0	0.0	0.0	0.125	0.0	0.125	0.125	0.125	0.0	0.0

In [26]:

```
filtered_sentence = [w for w in wordDict1 if w not in stop_words]
filtered_sentence
```

Out[26]:

```
['learning',
 'century',
 'Science',
 'job',
 'key',
 'data',
 'science',
 'machine',
 '21st',
 'best',
 'Data']
```

Inverse Document Frequency (IDF)

Formula: idf(t) = log(N/(df + 1))

In [27]:

```
import math
def getIDF(documents):
  n = len(documents)
  res = {}
  res = dict.fromkeys(documents[0].keys(), 0)

  for word, count in res.items():
    res[word] = math.log10(n / (float(count) + 1))
  return res
```

In [28]:

```
idfs = getIDF([wordDict1, wordDict2])
idfs
```

Out[28]:

```
{'the': 0.3010299956639812,
 'is': 0.3010299956639812,
 'learning': 0.3010299956639812,
 'for': 0.3010299956639812,
 'century': 0.3010299956639812,
 'Science': 0.3010299956639812,
 'job': 0.3010299956639812,
 'key': 0.3010299956639812,
 'of': 0.3010299956639812,
 'data': 0.3010299956639812,
 'science': 0.3010299956639812,
 'machine': 0.3010299956639812,
 '21st': 0.3010299956639812,
 'best': 0.3010299956639812,
 'Data': 0.3010299956639812}
```

In [29]:

```
def getTFIDF(tf, idf):
  tfidf = {}
  for word, count in tf.items():
    tfidf[word] = count*idf[word]
  return tfidf
```

In [30]:

```
tfidf1 = getTFIDF(tf1, idfs)
tfidf2 = getTFIDF(tf2, idfs)

pdTFIDF = pd.DataFrame([tfidf1, tfidf2])
pdTFIDF
```

Out[30]:

	the	is	learning	for	century	Science	job	key	of	data	
0	0.060206	0.030103	0.000000	0.000000	0.030103	0.030103	0.030103	0.000000	0.030103	0.000000	0.
1	0.037629	0.037629	0.037629	0.037629	0.000000	0.000000	0.000000	0.037629	0.000000	0.037629	0.

TFIDF using sklearn

In [31]:

```
from sklearn.feature_extraction.text import TfidfVectorizer


firstV= "Data Science is the sexiest job of the 21st century"
secondV= "machine learning is the key for data science"

vectorize= TfidfVectorizer()

response= vectorize.fit_transform([firstV, secondV])

# get idf values
print('\nIdf values:')
for ele1, ele2 in zip(vectorize.get_feature_names(), vectorize.idf_):
    print(ele1, ':', ele2)
```

```
Idf values:
21st : 1.4054651081081644
century : 1.4054651081081644
data : 1.0
for : 1.4054651081081644
is : 1.0
job : 1.4054651081081644
key : 1.4054651081081644
learning : 1.4054651081081644
machine : 1.4054651081081644
of : 1.4054651081081644
science : 1.0
```

```
sexiest : 1.4054651081081644
the : 1.0
```

In [32]:

```
print('\nTf-Idf values:')
print(response)
```

```
Tf-Idf values:
  (0, 1)	0.34211869506421816
  (0, 0)	0.34211869506421816
  (0, 9)	0.34211869506421816
  (0, 5)	0.34211869506421816
  (0, 11)	0.34211869506421816
  (0, 12)	0.48684053853849035
  (0, 4)	0.24342026926924518
  (0, 10)	0.24342026926924518
  (0, 2)	0.24342026926924518
  (1, 3)	0.40740123733358447
  (1, 6)	0.40740123733358447
  (1, 7)	0.40740123733358447
  (1, 8)	0.40740123733358447
  (1, 12)	0.28986933576883284
  (1, 4)	0.28986933576883284
  (1, 10)	0.28986933576883284
  (1, 2)	0.28986933576883284
```

In [ ]: