text_analysis

May 11, 2022

0.0.1 Text Analytics

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

[14]:
```
import nltk
import pandas as pd
import numpy as np
```

[13]:
```
#Sentence Tokenization
#Sentence tokenizer breaks text paragraph into sentences.

from nltk.tokenize import sent_tokenize
text="""Supervised Regression
In this case, the problem definition is rather similar to the previous example;␣
 ↪the difference
relies on the response. In a regression problem, the response y   , this means␣
 ↪the response is
real valued. For example, we can develop a model to predict the hourly salary␣
 ↪of individuals
given the corpus of their CV.
Unsupervised Learning
Management is often thirsty for new insights. Segmentation models can provide␣
 ↪this insight in
order for the marketing department to develop products for different segments.␣
 ↪A good
approach for developing a segmentation model, rather than thinking of␣
 ↪algorithms, is to select
features that are relevant to the segmentation that is desired.
For example, in a telecommunications company, it is interesting to segment␣
 ↪clients by their cell
phone usage. This would involve disregarding features that have nothing to do␣
 ↪with the
segmentation objective and including only those that do. In this case, this␣
 ↪would be selecting
```

```
features as the number of SMS used in a month, the number of inbound and␣
 ↪outbound minutes,
etc.
Big Data Analytics -Data Collection:
Data collection plays the most important role in the Big Data cycle. The␣
 ↪Internet provides
almost unlimited sources of data for a variety of topics. The importance of␣
 ↪this area depends on
the type of business, but traditional industries can acquire a diverse source␣
 ↪of external data and
combine those with their transactional data.
For example, let‘s assume we would like to build a system that recommends␣
 ↪restaurants. The
first step would be to gather data, in this case, reviews of restaurants from␣
 ↪different websites and
store them in a database. As we are interested in raw text, and would use that␣
 ↪for analytics, it is
not that relevant where the data for developing the model would be stored. This␣
 ↪may sound
contradictory with the big data main technologies, but in order to implement a␣
 ↪big data
application, we simply need to make it work in real time"""
```

[14]:
```
tokenized_text=sent_tokenize(text)
print(tokenized_text)
```

```
['Supervised Regression\nIn this case, the problem definition is rather similar
to the previous example; the difference \nrelies on the response.', 'In a
regression problem, the response y    , this means the response is \nreal
valued.', 'For example, we can develop a model to predict the hourly salary of
individuals \ngiven the corpus of their CV.', 'Unsupervised Learning\nManagement
is often thirsty for new insights.', 'Segmentation models can provide this
insight in \norder for the marketing department to develop products for
different segments.', 'A good \napproach for developing a segmentation model,
rather than thinking of algorithms, is to select \nfeatures that are relevant to
the segmentation that is desired.', 'For example, in a telecommunications
company, it is interesting to segment clients by their cell \nphone usage.',
'This would involve disregarding features that have nothing to do with the
\nsegmentation objective and including only those that do.', 'In this case, this
would be selecting \nfeatures as the number of SMS used in a month, the number
of inbound and outbound minutes, \netc.', 'Big Data Analytics -Data
Collection:\nData collection plays the most important role in the Big Data
cycle.', 'The Internet provides \nalmost unlimited sources of data for a variety
of topics.', 'The importance of this area depends on \nthe type of business, but
traditional industries can acquire a diverse source of external data and
\ncombine those with their transactional data.', 'For example, let‘s assume we
would like to build a system that recommends restaurants.', 'The \nfirst step
```

would be to gather data, in this case, reviews of restaurants from different websites and \nstore them in a database.', 'As we are interested in raw text, and would use that for analytics, it is \nnot that relevant where the data for developing the model would be stored.', 'This may sound \ncontradictory with the big data main technologies, but in order to implement a big data \napplication, we simply need to make it work in real time']

[15]:
```
#Word Tokenization
#Word tokenizer breaks text paragraph into words.

from nltk.tokenize import word_tokenize
tokenized_word=word_tokenize(text)
print(tokenized_word)
```

```
['Supervised', 'Regression', 'In', 'this', 'case', ',', 'the', 'problem',
'definition', 'is', 'rather', 'similar', 'to', 'the', 'previous', 'example',
';', 'the', 'difference', 'relies', 'on', 'the', 'response', '.', 'In', 'a',
'regression', 'problem', ',', 'the', 'response', 'y', ' ', ' ', ',', 'this',
'means', 'the', 'response', 'is', 'real', 'valued', '.', 'For', 'example', ',',
'we', 'can', 'develop', 'a', 'model', 'to', 'predict', 'the', 'hourly',
'salary', 'of', 'individuals', 'given', 'the', 'corpus', 'of', 'their', 'CV',
'.', 'Unsupervised', 'Learning', 'Management', 'is', 'often', 'thirsty', 'for',
'new', 'insights', '.', 'Segmentation', 'models', 'can', 'provide', 'this',
'insight', 'in', 'order', 'for', 'the', 'marketing', 'department', 'to',
'develop', 'products', 'for', 'different', 'segments', '.', 'A', 'good',
'approach', 'for', 'developing', 'a', 'segmentation', 'model', ',', 'rather',
'than', 'thinking', 'of', 'algorithms', ',', 'is', 'to', 'select', 'features',
'that', 'are', 'relevant', 'to', 'the', 'segmentation', 'that', 'is', 'desired',
'.', 'For', 'example', ',', 'in', 'a', 'telecommunications', 'company', ',',
'it', 'is', 'interesting', 'to', 'segment', 'clients', 'by', 'their', 'cell',
'phone', 'usage', '.', 'This', 'would', 'involve', 'disregarding', 'features',
'that', 'have', 'nothing', 'to', 'do', 'with', 'the', 'segmentation',
'objective', 'and', 'including', 'only', 'those', 'that', 'do', '.', 'In',
'this', 'case', ',', 'this', 'would', 'be', 'selecting', 'features', 'as',
'the', 'number', 'of', 'SMS', 'used', 'in', 'a', 'month', ',', 'the', 'number',
'of', 'inbound', 'and', 'outbound', 'minutes', ',', 'etc', '.', 'Big', 'Data',
'Analytics', '-Data', 'Collection', ':', 'Data', 'collection', 'plays', 'the',
'most', 'important', 'role', 'in', 'the', 'Big', 'Data', 'cycle', '.', 'The',
'Internet', 'provides', 'almost', 'unlimited', 'sources', 'of', 'data', 'for',
'a', 'variety', 'of', 'topics', '.', 'The', 'importance', 'of', 'this', 'area',
'depends', 'on', 'the', 'type', 'of', 'business', ',', 'but', 'traditional',
'industries', 'can', 'acquire', 'a', 'diverse', 'source', 'of', 'external',
'data', 'and', 'combine', 'those', 'with', 'their', 'transactional', 'data',
'.', 'For', 'example', ',', 'let', '‘', 's', 'assume', 'we', 'would', 'like',
'to', 'build', 'a', 'system', 'that', 'recommends', 'restaurants', '.', 'The',
'first', 'step', 'would', 'be', 'to', 'gather', 'data', ',', 'in', 'this',
'case', ',', 'reviews', 'of', 'restaurants', 'from', 'different', 'websites',
'and', 'store', 'them', 'in', 'a', 'database', '.', 'As', 'we', 'are',
```

```
'interested', 'in', 'raw', 'text', ',', 'and', 'would', 'use', 'that', 'for',
'analytics', ',', 'it', 'is', 'not', 'that', 'relevant', 'where', 'the', 'data',
'for', 'developing', 'the', 'model', 'would', 'be', 'stored', '.', 'This',
'may', 'sound', 'contradictory', 'with', 'the', 'big', 'data', 'main',
'technologies', ',', 'but', 'in', 'order', 'to', 'implement', 'a', 'big',
'data', 'application', ',', 'we', 'simply', 'need', 'to', 'make', 'it', 'work',
'in', 'real', 'time']
```

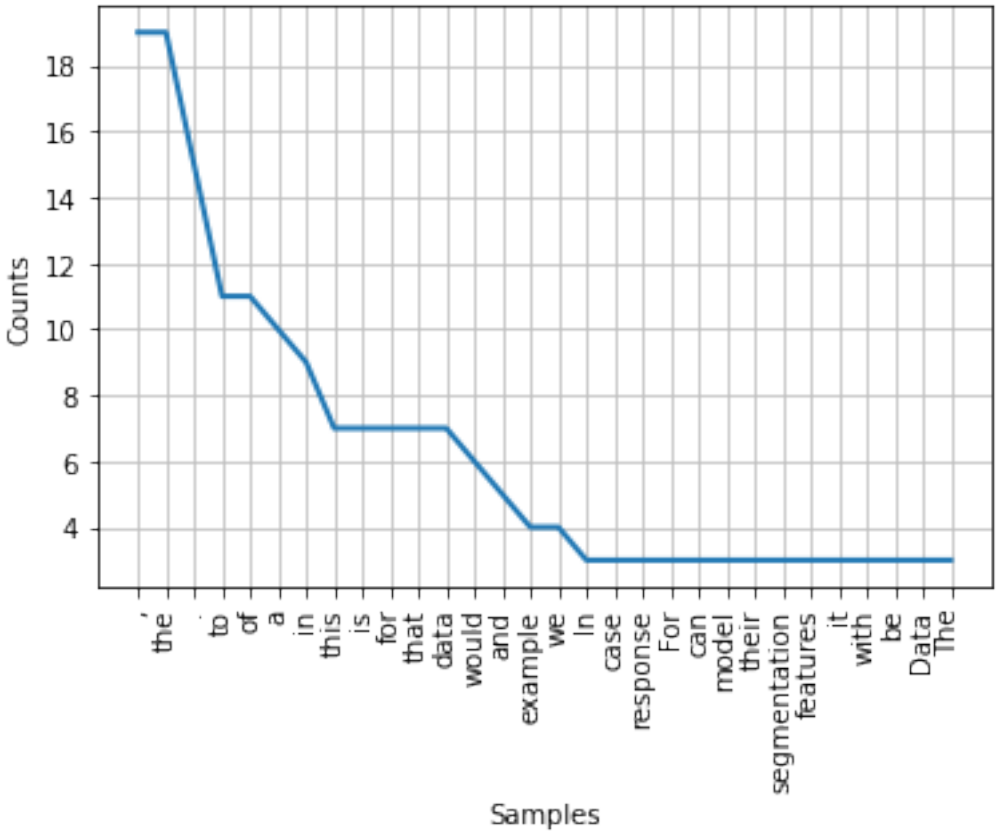
[16]:
```
#Frequency Distribution

from nltk.probability import FreqDist
fdist = FreqDist(tokenized_word)
print(fdist)
```

```
<FreqDist with 185 samples and 363 outcomes>
```

[17]:
```
fdist.most_common(2)
```

[17]:
```
[(',', 19), ('the', 19)]
```

[18]:
```
# Frequency Distribution Plot
import matplotlib.pyplot as plt
fdist.plot(30,cumulative=False)
plt.show()
```



[21]:
```
#Stopwords
#Stopwords considered as noise in the text.
#Text may contain stop words such as is, am, are, this, a, an, the, etc.

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

```
{'out', "that'll", 'hasn', 'ma', 'that', 'ourselves', 'too', 'myself', "she's",
'am', 'about', "you'll", 'be', 'before', 'and', 'her', 'does', 'my', 'mightn',
"haven't", 'above', 'just', 'his', "you've", 'he', 'during', 'couldn', 'with',
'all', 'o', "mustn't", 'herself', "wasn't", "wouldn't", 'hadn', 'ain',
"weren't", 'has', 'themselves', 'same', 'by', 'didn', 'most', 'not', 'who',
'wouldn', 'do', 'ours', 'yours', 'its', 'through', "aren't", "didn't", "hadn't",
'once', 'them', 'then', "needn't", 'other', 'shan', "doesn't", 'yourself', 'or',
'y', 'are', 'where', 'me', 'd', 'nor', 'some', 'll', 'himself', 'for', 'needn',
'so', 'she', 'further', 'm', 'your', 'we', 'him', 'weren', 'an', "you'd",
'into', 'now', 'after', 'but', 'mustn', 'off', 'haven', 'were', 'i', 'those',
'each', 'under', 't', 'our', 'the', "shan't", 'over', 'shouldn', 'have', 'than',
'will', 'had', 'won', 'been', 'isn', 'having', 'this', 'itself', 'against',
'don', "mightn't", 'you', 'what', 'how', 'was', 'should', 'only', "couldn't",
```

```
'until', 'up', "isn't", 'their', 'it', 'theirs', 'both', 'own', 'there',
'between', 'which', 'here', 'while', 'below', 'down', 'to', 'these', "don't",
'from', 'doing', 've', 'a', 'no', "shouldn't", 'in', 'at', "it's", 'on', 'aren',
'such', "you're", 'more', 'why', 'again', 'if', "hasn't", 'being', 'they',
'can', 'doesn', 'very', 'of', "should've", 'hers', 's', 'yourselves', 'when',
'wasn', 'is', 're', 'did', 'few', 'whom', 'any', 'as', "won't", 'because'}
```

[23]:
```
#Removing Stopwords
#In NLTK for removing stopwords, you need to create a list of stopwords
#and filter out your list of tokens from these words.

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

```
Tokenized Sentence: ['Supervised', 'Regression', 'In', 'this', 'case', ',',
'the', 'problem', 'definition', 'is', 'rather', 'similar', 'to', 'the',
'previous', 'example', ';', 'the', 'difference', 'relies', 'on', 'the',
'response', '.', 'In', 'a', 'regression', 'problem', ',', 'the', 'response',
'y', ' ', ' ', ',', 'this', 'means', 'the', 'response', 'is', 'real', 'valued',
'.', 'For', 'example', ',', 'we', 'can', 'develop', 'a', 'model', 'to',
'predict', 'the', 'hourly', 'salary', 'of', 'individuals', 'given', 'the',
'corpus', 'of', 'their', 'CV', '.', 'Unsupervised', 'Learning', 'Management',
'is', 'often', 'thirsty', 'for', 'new', 'insights', '.', 'Segmentation',
'models', 'can', 'provide', 'this', 'insight', 'in', 'order', 'for', 'the',
'marketing', 'department', 'to', 'develop', 'products', 'for', 'different',
'segments', '.', 'A', 'good', 'approach', 'for', 'developing', 'a',
'segmentation', 'model', ',', 'rather', 'than', 'thinking', 'of', 'algorithms',
',', 'is', 'to', 'select', 'features', 'that', 'are', 'relevant', 'to', 'the',
'segmentation', 'that', 'is', 'desired', '.', 'For', 'example', ',', 'in', 'a',
'telecommunications', 'company', ',', 'it', 'is', 'interesting', 'to',
'segment', 'clients', 'by', 'their', 'cell', 'phone', 'usage', '.', 'This',
'would', 'involve', 'disregarding', 'features', 'that', 'have', 'nothing', 'to',
'do', 'with', 'the', 'segmentation', 'objective', 'and', 'including', 'only',
'those', 'that', 'do', '.', 'In', 'this', 'case', ',', 'this', 'would', 'be',
'selecting', 'features', 'as', 'the', 'number', 'of', 'SMS', 'used', 'in', 'a',
'month', ',', 'the', 'number', 'of', 'inbound', 'and', 'outbound', 'minutes',
',', 'etc', '.', 'Big', 'Data', 'Analytics', '-Data', 'Collection', ':', 'Data',
'collection', 'plays', 'the', 'most', 'important', 'role', 'in', 'the', 'Big',
'Data', 'cycle', '.', 'The', 'Internet', 'provides', 'almost', 'unlimited',
'sources', 'of', 'data', 'for', 'a', 'variety', 'of', 'topics', '.', 'The',
'importance', 'of', 'this', 'area', 'depends', 'on', 'the', 'type', 'of',
'business', ',', 'but', 'traditional', 'industries', 'can', 'acquire', 'a',
'diverse', 'source', 'of', 'external', 'data', 'and', 'combine', 'those',
```

```
'with', 'their', 'transactional', 'data', '.', 'For', 'example', ',', 'let',
'‘', 's', 'assume', 'we', 'would', 'like', 'to', 'build', 'a', 'system', 'that',
'recommends', 'restaurants', '.', 'The', 'first', 'step', 'would', 'be', 'to',
'gather', 'data', ',', 'in', 'this', 'case', ',', 'reviews', 'of',
'restaurants', 'from', 'different', 'websites', 'and', 'store', 'them', 'in',
'a', 'database', '.', 'As', 'we', 'are', 'interested', 'in', 'raw', 'text', ',',
'and', 'would', 'use', 'that', 'for', 'analytics', ',', 'it', 'is', 'not',
'that', 'relevant', 'where', 'the', 'data', 'for', 'developing', 'the', 'model',
'would', 'be', 'stored', '.', 'This', 'may', 'sound', 'contradictory', 'with',
'the', 'big', 'data', 'main', 'technologies', ',', 'but', 'in', 'order', 'to',
'implement', 'a', 'big', 'data', 'application', ',', 'we', 'simply', 'need',
'to', 'make', 'it', 'work', 'in', 'real', 'time']

====================================

Filterd Sentence: ['Supervised', 'Regression', 'In', 'case', ',', 'problem',
'definition', 'rather', 'similar', 'previous', 'example', ';', 'difference',
'relies', 'response', '.', 'In', 'regression', 'problem', ',', 'response', ' ',
' ', ',', 'means', 'response', 'real', 'valued', '.', 'For', 'example', ',',
'develop', 'model', 'predict', 'hourly', 'salary', 'individuals', 'given',
'corpus', 'CV', '.', 'Unsupervised', 'Learning', 'Management', 'often',
'thirsty', 'new', 'insights', '.', 'Segmentation', 'models', 'provide',
'insight', 'order', 'marketing', 'department', 'develop', 'products',
'different', 'segments', '.', 'A', 'good', 'approach', 'developing',
'segmentation', 'model', ',', 'rather', 'thinking', 'algorithms', ',', 'select',
'features', 'relevant', 'segmentation', 'desired', '.', 'For', 'example', ',',
'telecommunications', 'company', ',', 'interesting', 'segment', 'clients',
'cell', 'phone', 'usage', '.', 'This', 'would', 'involve', 'disregarding',
'features', 'nothing', 'segmentation', 'objective', 'including', '.', 'In',
'case', ',', 'would', 'selecting', 'features', 'number', 'SMS', 'used', 'month',
',', 'number', 'inbound', 'outbound', 'minutes', ',', 'etc', '.', 'Big', 'Data',
'Analytics', '-Data', 'Collection', ':', 'Data', 'collection', 'plays',
'important', 'role', 'Big', 'Data', 'cycle', '.', 'The', 'Internet', 'provides',
'almost', 'unlimited', 'sources', 'data', 'variety', 'topics', '.', 'The',
'importance', 'area', 'depends', 'type', 'business', ',', 'traditional',
'industries', 'acquire', 'diverse', 'source', 'external', 'data', 'combine',
'transactional', 'data', '.', 'For', 'example', ',', 'let', '‘', 'assume',
'would', 'like', 'build', 'system', 'recommends', 'restaurants', '.', 'The',
'first', 'step', 'would', 'gather', 'data', ',', 'case', ',', 'reviews',
'restaurants', 'different', 'websites', 'store', 'database', '.', 'As',
'interested', 'raw', 'text', ',', 'would', 'use', 'analytics', ',', 'relevant',
'data', 'developing', 'model', 'would', 'stored', '.', 'This', 'may', 'sound',
'contradictory', 'big', 'data', 'main', 'technologies', ',', 'order',
'implement', 'big', 'data', 'application', ',', 'simply', 'need', 'make',
'work', 'real', 'time']
```

[24]:
```
# Stemming
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize

ps = PorterStemmer()

stemmed_words=[]
for w in filtered_sent:
    stemmed_words.append(ps.stem(w))

print("Filtered Sentence:",filtered_sent)
print("\n========================================\n")
print("Stemmed Sentence:",stemmed_words)
```

```
Filtered Sentence: ['Supervised', 'Regression', 'In', 'case', ',', 'problem',
'definition', 'rather', 'similar', 'previous', 'example', ';', 'difference',
'relies', 'response', '.', 'In', 'regression', 'problem', ',', 'response', ' ',
' ', ',', 'means', 'response', 'real', 'valued', '.', 'For', 'example', ',',
'develop', 'model', 'predict', 'hourly', 'salary', 'individuals', 'given',
'corpus', 'CV', '.', 'Unsupervised', 'Learning', 'Management', 'often',
'thirsty', 'new', 'insights', '.', 'Segmentation', 'models', 'provide',
'insight', 'order', 'marketing', 'department', 'develop', 'products',
'different', 'segments', '.', 'A', 'good', 'approach', 'developing',
'segmentation', 'model', ',', 'rather', 'thinking', 'algorithms', ',', 'select',
'features', 'relevant', 'segmentation', 'desired', '.', 'For', 'example', ',',
'telecommunications', 'company', ',', 'interesting', 'segment', 'clients',
'cell', 'phone', 'usage', '.', 'This', 'would', 'involve', 'disregarding',
'features', 'nothing', 'segmentation', 'objective', 'including', '.', 'In',
'case', ',', 'would', 'selecting', 'features', 'number', 'SMS', 'used', 'month',
',', 'number', 'inbound', 'outbound', 'minutes', ',', 'etc', '.', 'Big', 'Data',
'Analytics', '-Data', 'Collection', ':', 'Data', 'collection', 'plays',
'important', 'role', 'Big', 'Data', 'cycle', '.', 'The', 'Internet', 'provides',
'almost', 'unlimited', 'sources', 'data', 'variety', 'topics', '.', 'The',
'importance', 'area', 'depends', 'type', 'business', ',', 'traditional',
'industries', 'acquire', 'diverse', 'source', 'external', 'data', 'combine',
'transactional', 'data', '.', 'For', 'example', ',', 'let', '‘', 'assume',
'would', 'like', 'build', 'system', 'recommends', 'restaurants', '.', 'The',
'first', 'step', 'would', 'gather', 'data', ',', 'case', ',', 'reviews',
'restaurants', 'different', 'websites', 'store', 'database', '.', 'As',
'interested', 'raw', 'text', ',', 'would', 'use', 'analytics', ',', 'relevant',
'data', 'developing', 'model', 'would', 'stored', '.', 'This', 'may', 'sound',
'contradictory', 'big', 'data', 'main', 'technologies', ',', 'order',
'implement', 'big', 'data', 'application', ',', 'simply', 'need', 'make',
'work', 'real', 'time']

========================================
```

```
Stemmed Sentence: ['supervis', 'regress', 'in', 'case', ',', 'problem',
'definit', 'rather', 'similar', 'previou', 'exampl', ';', 'differ', 'reli',
'respons', '.', 'in', 'regress', 'problem', ',', 'respons', ' ', ' ', ',',
'mean', 'respons', 'real', 'valu', '.', 'for', 'exampl', ',', 'develop',
'model', 'predict', 'hourli', 'salari', 'individu', 'given', 'corpu', 'cv', '.',
'unsupervis', 'learn', 'manag', 'often', 'thirsti', 'new', 'insight', '.',
'segment', 'model', 'provid', 'insight', 'order', 'market', 'depart', 'develop',
'product', 'differ', 'segment', '.', 'a', 'good', 'approach', 'develop',
'segment', 'model', ',', 'rather', 'think', 'algorithm', ',', 'select',
'featur', 'relev', 'segment', 'desir', '.', 'for', 'exampl', ',', 'telecommun',
'compani', ',', 'interest', 'segment', 'client', 'cell', 'phone', 'usag', '.',
'thi', 'would', 'involv', 'disregard', 'featur', 'noth', 'segment', 'object',
'includ', '.', 'in', 'case', ',', 'would', 'select', 'featur', 'number', 'sm',
'use', 'month', ',', 'number', 'inbound', 'outbound', 'minut', ',', 'etc', '.',
'big', 'data', 'analyt', '-data', 'collect', ':', 'data', 'collect', 'play',
'import', 'role', 'big', 'data', 'cycl', '.', 'the', 'internet', 'provid',
'almost', 'unlimit', 'sourc', 'data', 'varieti', 'topic', '.', 'the', 'import',
'area', 'depend', 'type', 'busi', ',', 'tradit', 'industri', 'acquir', 'divers',
'sourc', 'extern', 'data', 'combin', 'transact', 'data', '.', 'for', 'exampl',
',', 'let', '‘', 'assum', 'would', 'like', 'build', 'system', 'recommend',
'restaur', '.', 'the', 'first', 'step', 'would', 'gather', 'data', ',', 'case',
',', 'review', 'restaur', 'differ', 'websit', 'store', 'databas', '.', 'as',
'interest', 'raw', 'text', ',', 'would', 'use', 'analyt', ',', 'relev', 'data',
'develop', 'model', 'would', 'store', '.', 'thi', 'may', 'sound',
'contradictori', 'big', 'data', 'main', 'technolog', ',', 'order', 'implement',
'big', 'data', 'applic', ',', 'simpli', 'need', 'make', 'work', 'real', 'time']
```

[2]:
```
#Lexicon Normalization
#performing stemming and Lemmatization

from nltk.stem.wordnet import WordNetLemmatizer
lem = WordNetLemmatizer()

from nltk.stem.porter import PorterStemmer
stem = PorterStemmer()

word = "went"
print("Lemmatized Word:",lem.lemmatize(word,"v"))
print("Stemmed Word:",stem.stem(word))
```

```
Lemmatized Word: go
Stemmed Word: went
```

[35]:
```
#POS Tagging

sent = "Tagging is a kind of classification that may be defined as the␣
 ↪automatic assignment of description to the tokens."
```

```
tokens=nltk.word_tokenize(sent)
print(tokens)
```

```
['Tagging', 'is', 'a', 'kind', 'of', 'classification', 'that', 'may', 'be',
'defined', 'as', 'the', 'automatic', 'assignment', 'of', 'description', 'to',
'the', 'tokens', '.']
```

[38]:
```
nltk.pos_tag(tokens)
```

[38]:
```
[('Tagging', 'NN'),
 ('is', 'VBZ'),
 ('a', 'DT'),
 ('kind', 'NN'),
 ('of', 'IN'),
 ('classification', 'NN'),
 ('that', 'WDT'),
 ('may', 'MD'),
 ('be', 'VB'),
 ('defined', 'VBN'),
 ('as', 'IN'),
 ('the', 'DT'),
 ('automatic', 'JJ'),
 ('assignment', 'NN'),
 ('of', 'IN'),
 ('description', 'NN'),
 ('to', 'TO'),
 ('the', 'DT'),
 ('tokens', 'NNS'),
 ('.', '.')]
```

2. Create representation of document by calculating Term Frequency and Inverse Document Frequency.

[6]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
corpus = ['data science is one of the most important fields of science',
          'this is one of the best data science courses',
          'data scientists analyze data' ]
```

[7]:
```
tr_idf_model  = TfidfVectorizer()
tf_idf_vector = tr_idf_model.fit_transform(corpus)
```

[10]:
```
tf_idf_vector
```

[10]:
```
<3x14 sparse matrix of type '<class 'numpy.float64'>'
        with 21 stored elements in Compressed Sparse Row format>
```

[11]:
```
tf_idf_array = tf_idf_vector.toarray()

print(tf_idf_array)
```

```
[[0.         0.         0.         0.18952581 0.32089509 0.32089509
  0.24404899 0.32089509 0.48809797 0.24404899 0.48809797 0.
  0.24404899 0.        ]
 [0.         0.40029393 0.40029393 0.23642005 0.         0.
  0.30443385 0.         0.30443385 0.30443385 0.30443385 0.
  0.30443385 0.40029393]
 [0.54270061 0.         0.         0.64105545 0.         0.
  0.         0.         0.         0.         0.         0.54270061
  0.         0.        ]]
```

[12]:
```
words_set = tr_idf_model.get_feature_names()

print(words_set)
```

```
['analyze', 'best', 'courses', 'data', 'fields', 'important', 'is', 'most',
'of', 'one', 'science', 'scientists', 'the', 'this']
```

[15]:
```
df_tf_idf = pd.DataFrame(tf_idf_array, columns = words_set)

df_tf_idf
```

[15]:

	analyze	best	courses	data	fields	important	is	most	of	one	science	scientists	the	this
0	0.000000	0.000000	0.000000	0.189526	0.320895	0.320895	0.244049	0.320895	0.488098	0.244049	0.488098	0.000000	0.244049	0.000000
1	0.000000	0.400294	0.400294	0.236420	0.000000	0.000000	0.304434	0.000000	0.304434	0.304434	0.304434	0.000000	0.304434	0.400294
2	0.542701	0.000000	0.000000	0.641055	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.542701	0.000000	0.000000