untitled7

May 7, 2024

[3]:
```
import nltk
from nltk import pos_tag
nltk.download('averaged_perceptron_tagger')
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
import pandas as pd
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
```

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

[4]:
```
sample_document = """Text analytics involves analyzing unstructured text data␣
  ↪to extract meaningful insights.
It includes preprocessing steps such as tokenization, POS tagging, stop words␣
  ↪removal,
stemming, and lemmatization. Text analytics techniques are widely used in␣
  ↪natural language
processing (NLP), sentiment analysis, information retrieval, and text␣
  ↪classification.
The goal of text analytics is to transform text data into a structured format␣
  ↪that can be
```

```
used for further analysis and modeling. This process typically involves␣
 ↪cleaning and
preprocessing the text data, extracting features, and applying machine learning␣
 ↪algorithms.
Some common text analytics tasks include document classification, topic␣
 ↪modeling, named
entity recognition, and text summarization. With the increasing availability of␣
 ↪textual
data from sources such as social media, websites, and documents, text analytics␣
 ↪has become
an essential tool for businesses, researchers, and data scientists."""
```

```
[5]: tokens = word_tokenize(sample_document)
     tokens
```

```
[5]: ['Text',
      'analytics',
      'involves',
      'analyzing',
      'unstructured',
      'text',
      'data',
      'to',
      'extract',
      'meaningful',
      'insights',
      '.',
      'It',
      'includes',
      'preprocessing',
      'steps',
      'such',
      'as',
      'tokenization',
      ',',
      'POS',
      'tagging',
      ',',
      'stop',
      'words',
      'removal',
      ',',
      'stemming',
      ',',
      'and',
      'lemmatization',
      '.',
      'Text',
      'analytics',
      'techniques',
      'are',
      'widely',
      'used',
      'in',
      'natural',
      'language',
      'processing',
      '(',
      'NLP',
      ')',
      ',',
      'sentiment',
      'analysis',
      ',',
      'information',
      'retrieval',
      ',',
      'and',
      'text',
      'classification',
      '.',
      'The',
      'goal',
      'of',
      'text',
      'analytics',
      'is',
      'to',
      'transform',
      'text',
      'data',
      'into',
      'a',
      'structured',
      'format',
      'that',
      'can',
      'be',
      'used',
      'for',
      'further',
      'analysis',
      'and',
      'modeling',
      '.',
      'This',
      'process',
      'typically',
      'involves',
      'cleaning',
      'and',
      'preprocessing',
      'the',
      'text',
      'data',
      ',',
      'extracting',
      'features',
      ',',
      'and',
      'applying',
      'machine',
      'learning',
      'algorithms',
      '.',
      'Some',
      'common',
      'text',
      'analytics',
      'tasks',
      'include',
      'document',
      'classification',
      ',',
      'topic',
      'modeling',
      ',',
      'named',
      'entity',
      'recognition',
      ',',
      'and',
      'text',
      'summarization',
      '.',
      'With',
      'the',
      'increasing',
      'availability',
      'of',
      'textual',
       'data',
       'from',
       'sources',
       'such',
       'as',
       'social',
       'media',
       ',',
       'websites',
       ',',
       'and',
       'documents',
       ',',
       'text',
       'analytics',
       'has',
       'become',
       'an',
       'essential',
       'tool',
       'for',
       'businesses',
       ',',
       'researchers',
       ',',
       'and',
       'data',
       'scientists',
       '.']
```

[6]:
```
posTagWords = pos_tag(tokens)
posTagWords
```

[6]:
```
[('Text', 'JJ'),
 ('analytics', 'NNS'),
 ('involves', 'VBZ'),
 ('analyzing', 'VBG'),
 ('unstructured', 'JJ'),
 ('text', 'NN'),
 ('data', 'NNS'),
 ('to', 'TO'),
 ('extract', 'VB'),
 ('meaningful', 'JJ'),
 ('insights', 'NNS'),
 ('.', '.'),
 ('It', 'PRP'),
 ('includes', 'VBZ'),
('preprocessing', 'VBG'),
('steps', 'NNS'),
('such', 'JJ'),
('as', 'IN'),
('tokenization', 'NN'),
(',', ','),
('POS', 'NNP'),
('tagging', 'NN'),
(',', ','),
('stop', 'VB'),
('words', 'NNS'),
('removal', 'JJ'),
(',', ','),
('stemming', 'VBG'),
(',', ','),
('and', 'CC'),
('lemmatization', 'NN'),
('.', '.'),
('Text', 'NNP'),
('analytics', 'NNS'),
('techniques', 'NNS'),
('are', 'VBP'),
('widely', 'RB'),
('used', 'VBN'),
('in', 'IN'),
('natural', 'JJ'),
('language', 'NN'),
('processing', 'NN'),
('(', '('),
('NLP', 'NNP'),
(')', ')'),
(',', ','),
('sentiment', 'JJ'),
('analysis', 'NN'),
(',', ','),
('information', 'NN'),
('retrieval', 'NN'),
(',', ','),
('and', 'CC'),
('text', 'JJ'),
('classification', 'NN'),
('.', '.'),
('The', 'DT'),
('goal', 'NN'),
('of', 'IN'),
('text', 'JJ'),
('analytics', 'NNS'),
('is', 'VBZ'),
('to', 'TO'),
('transform', 'VB'),
('text', 'NN'),
('data', 'NNS'),
('into', 'IN'),
('a', 'DT'),
('structured', 'JJ'),
('format', 'NN'),
('that', 'WDT'),
('can', 'MD'),
('be', 'VB'),
('used', 'VBN'),
('for', 'IN'),
('further', 'JJ'),
('analysis', 'NN'),
('and', 'CC'),
('modeling', 'NN'),
('.', '.'),
('This', 'DT'),
('process', 'NN'),
('typically', 'RB'),
('involves', 'VBZ'),
('cleaning', 'NN'),
('and', 'CC'),
('preprocessing', 'VBG'),
('the', 'DT'),
('text', 'NN'),
('data', 'NNS'),
(',', ','),
('extracting', 'VBG'),
('features', 'NNS'),
(',', ','),
('and', 'CC'),
('applying', 'VBG'),
('machine', 'NN'),
('learning', 'NN'),
('algorithms', 'NN'),
('.', '.'),
('Some', 'DT'),
('common', 'JJ'),
('text', 'NN'),
('analytics', 'NNS'),
('tasks', 'NNS'),
('include', 'VBP'),
('document', 'JJ'),
('classification', 'NN'),
(',', ','),
('topic', 'NN'),
('modeling', 'NN'),
(',', ','),
('named', 'VBN'),
('entity', 'NN'),
('recognition', 'NN'),
(',', ','),
('and', 'CC'),
('text', 'JJ'),
('summarization', 'NN'),
('.', '.'),
('With', 'IN'),
('the', 'DT'),
('increasing', 'VBG'),
('availability', 'NN'),
('of', 'IN'),
('textual', 'JJ'),
('data', 'NNS'),
('from', 'IN'),
('sources', 'NNS'),
('such', 'JJ'),
('as', 'IN'),
('social', 'JJ'),
('media', 'NNS'),
(',', ','),
('websites', 'NNS'),
(',', ','),
('and', 'CC'),
('documents', 'NNS'),
(',', ','),
('text', 'NN'),
('analytics', 'NNS'),
('has', 'VBZ'),
('become', 'VBN'),
('an', 'DT'),
('essential', 'JJ'),
('tool', 'NN'),
('for', 'IN'),
('businesses', 'NNS'),
(',', ','),
('researchers', 'NNS'),
(',', ','),
('and', 'CC'),
('data', 'NNS'),
('scientists', 'NNS'),
('.', '.')]
```

[7]:

```
stop_words = set(stopwords.words('english'))
tokenized_words = [word for word in tokens if word.lower() not in stop_words]
tokenized_words
```

[7]:

```
['Text',
 'analytics',
 'involves',
 'analyzing',
 'unstructured',
 'text',
 'data',
 'extract',
 'meaningful',
 'insights',
 '.',
 'includes',
 'preprocessing',
 'steps',
 'tokenization',
 ',',
 'POS',
 'tagging',
 ',',
 'stop',
 'words',
 'removal',
 ',',
 'stemming',
 ',',
 'lemmatization',
 '.',
 'Text',
 'analytics',
 'techniques',
 'widely',
 'used',
 'natural',
 'language',
 'processing',
 '(',
 'NLP',
 ')',
 ',',
 'sentiment',
 'analysis',
 ',',
 'information',
 'retrieval',
 ',',
 'text',
 'classification',
 '.',
 'goal',
 'text',
 'analytics',
 'transform',
 'text',
 'data',
 'structured',
 'format',
 'used',
 'analysis',
 'modeling',
 '.',
 'process',
 'typically',
 'involves',
 'cleaning',
 'preprocessing',
 'text',
 'data',
 ',',
 'extracting',
 'features',
 ',',
 'applying',
 'machine',
 'learning',
 'algorithms',
 '.',
 'common',
 'text',
 'analytics',
 'tasks',
 'include',
 'document',
 'classification',
 ',',
 'topic',
 'modeling',
 ',',
 'named',
 'entity',
 'recognition',
 ',',
 'text',
 'summarization',
 '.',
 'increasing',
 'availability',
 'textual',
 'data',
 'sources',
 'social',
 'media',
 ',',
 'websites',
 ',',
 'documents',
 ',',
 'text',
 'analytics',
 'become',
 'essential',
 'tool',
 'businesses',
 ',',
 'researchers',
 ',',
 'data',
 'scientists',
 '.']
```

[8]:

```
stemmer = PorterStemmer()
```

[9]:

```
stemmed_words = [stemmer.stem(word) for word in tokenized_words]
stemmed_words
```

[9]:

```
['text',
 'analyt',
 'involv',
 'analyz',
 'unstructur',
 'text',
 'data',
 'extract',
 'meaning',
 'insight',
 '.',
 'includ',
 'preprocess',
'step',
'token',
',',
'po',
'tag',
',',
'stop',
'word',
'remov',
',',
'stem',
',',
'lemmat',
'.',
'text',
'analyt',
'techniqu',
'wide',
'use',
'natur',
'languag',
'process',
'(',
'nlp',
')',
',',
'sentiment',
'analysi',
',',
'inform',
'retriev',
',',
'text',
'classif',
'.',
'goal',
'text',
'analyt',
'transform',
'text',
'data',
'structur',
'format',
'use',
'analysi',
'model',
'.',
'process',
'typic',
'involv',
'clean',
'preprocess',
'text',
'data',
',',
'extract',
'featur',
',',
'appli',
'machin',
'learn',
'algorithm',
'.',
'common',
'text',
'analyt',
'task',
'includ',
'document',
'classif',
',',
'topic',
'model',
',',
'name',
'entiti',
'recognit',
',',
'text',
'summar',
'.',
'increas',
'avail',
'textual',
'data',
'sourc',
'social',
'media',
',',
'websit',
',',
'document',
',',
'text',
 'analyt',
 'becom',
 'essenti',
 'tool',
 'busi',
 ',',
 'research',
 ',',
 'data',
 'scientist',
 '.']
```

[15]:

```
lemmatizer=WordNetLemmatizer()
```

[30]:

```
lemmetized_words = [lemmetizer.lemmatize(word) for word in tokenized_words]

lemmatized_tokens = [lemmatizer.lemmatize(word, pos='v') if word != ',' else␣
 ↪word for word in tokenized_words]
lemmatized_tokens = [lemmatizer.lemmatize(word, pos='n') if word != ',' else␣
 ↪word for word in lemmatized_tokens]  # Nouns
lemmatized_tokens = [lemmatizer.lemmatize(word, pos='a') if word != ',' else␣
 ↪word for word in lemmatized_tokens]  # Adjectives
lemmatized_tokens = [lemmatizer.lemmatize(word, pos='r') if word != ',' else␣
 ↪word for word in lemmatized_tokens]  # Adverbs

lemmatized_tokens
```

[30]:

```
['Text',
 'analytics',
 'involve',
 'analyze',
 'unstructured',
 'text',
 'data',
 'extract',
 'meaningful',
 'insight',
 '.',
 'include',
 'preprocessing',
 'step',
 'tokenization',
 ',',
 'POS',
 'tag',
 ',',
 'stop',
'word',
'removal',
',',
'stem',
',',
'lemmatization',
'.',
'Text',
'analytics',
'technique',
'widely',
'use',
'natural',
'language',
'process',
'(',
'NLP',
')',
',',
'sentiment',
'analysis',
',',
'information',
'retrieval',
',',
'text',
'classification',
'.',
'goal',
'text',
'analytics',
'transform',
'text',
'data',
'structure',
'format',
'use',
'analysis',
'model',
'.',
'process',
'typically',
'involve',
'clean',
'preprocessing',
'text',
'data',
',',
'extract',
'feature',
',',
'apply',
'machine',
'learn',
'algorithm',
'.',
'common',
'text',
'analytics',
'task',
'include',
'document',
'classification',
',',
'topic',
'model',
',',
'name',
'entity',
'recognition',
',',
'text',
'summarization',
'.',
'increase',
'availability',
'textual',
'data',
'source',
'social',
'medium',
',',
'website',
',',
'document',
',',
'text',
'analytics',
'become',
'essential',
'tool',
'business',
',',
'researcher',
 ',',
 'data',
 'scientist',
 '.']
```

```
[26]: tfid_vectorizer = TfidfVectorizer()
      tfid_matrix = tfid_vectorizer.fit_transform([sample_document])
      tfid_matrix
```

```
[26]: <1x91 sparse matrix of type '<class 'numpy.float64'>'
              with 91 stored elements in Compressed Sparse Row format>
```

```
[28]: tfid_df = pd.DataFrame(tfid_matrix.toarray(),columns =tfid_vectorizer.
       ↪get_feature_names_out())
      tfid_df
      tfid_matrix.toarray()
```

```
[28]: array([[0.05407381, 0.05407381, 0.10814761, 0.27036904, 0.05407381,
              0.43259046, 0.05407381, 0.05407381, 0.10814761, 0.05407381,
              0.05407381, 0.05407381, 0.05407381, 0.05407381, 0.10814761,
              0.05407381, 0.05407381, 0.27036904, 0.05407381, 0.05407381,
              0.05407381, 0.05407381, 0.05407381, 0.05407381, 0.05407381,
              0.10814761, 0.05407381, 0.05407381, 0.05407381, 0.05407381,
              0.05407381, 0.05407381, 0.05407381, 0.05407381, 0.05407381,
              0.05407381, 0.05407381, 0.05407381, 0.10814761, 0.05407381,
              0.05407381, 0.05407381, 0.05407381, 0.05407381, 0.05407381,
              0.05407381, 0.05407381, 0.10814761, 0.05407381, 0.05407381,
              0.05407381, 0.10814761, 0.05407381, 0.10814761, 0.05407381,
              0.05407381, 0.05407381, 0.05407381, 0.05407381, 0.05407381,
              0.05407381, 0.05407381, 0.05407381, 0.05407381, 0.05407381,
              0.05407381, 0.05407381, 0.05407381, 0.05407381, 0.10814761,
              0.05407381, 0.05407381, 0.05407381, 0.05407381, 0.54073807,
              0.05407381, 0.05407381, 0.16222142, 0.05407381, 0.10814761,
              0.05407381, 0.05407381, 0.05407381, 0.05407381, 0.05407381,
              0.05407381, 0.10814761, 0.05407381, 0.05407381, 0.05407381,
              0.05407381]])
```

```
[ ]:
```