I081 Aniruddh Kulkarni NLP Exp5

May 28, 2023

1 Name: Aniruddh Kulkarni

2 Roll no: I081

3 Stream: CS (AI)

4 Division: I

5 Semester: 5th Semester

6 Batch: I-3

7 Subject: NLP

8 Assignment-5

[1]:
```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
import nltk
```

[2]:
```
corpus = ["NLP drives computer programs that translate text from one language␣
  ↪to another, respond to spoken commands, and summarize large volumes of text␣
  ↪rapidly-even in real time. There’s a good chance you’ve interacted with NLP␣
  ↪in the form of voice-operated GPS systems, digital assistants,␣
  ↪speech-to-text dictation software, customer service chatbots, and other␣
  ↪consumer conveniences. But NLP also plays a growing role in enterprise␣
  ↪solutions that help streamline business operations, increase employee␣
  ↪productivity, and simplify mission-critical business processes."]

#corpus = corpus.split()
```

[3]:
```
from nltk.corpus import stopwords
nltk.download('stopwords')
nltk.download('punkt')

#initialize stopwords
stop_words_nltk = set(stopwords.words('english'))
```

```
print(stop_words_nltk)

#stopword removal
corpus2 = [i for i in corpus if not i in stop_words_nltk]
print("Tokenized corpus without stopwords:",corpus2)
```

```
{'is', 'into', "she's", 'does', 'all', 'theirs', 't', 'as', 'about', 'them',
'their', 'while', 'having', 'i', 'an', "you'll", 'only', 'd', 'am', 'why',
'his', 'each', 'out', 'yourself', 'down', 'by', 'don', 'shan', 'until',
'during', 'couldn', 'such', "you're", 'but', "hasn't", 'your', 'too', 'mustn',
'between', 'will', 'him', 'yours', "it's", 'mightn', 'where', 'won', "won't",
"aren't", "shan't", 'do', 'these', "weren't", 'when', 'because', 'has', 'with',
'below', 'most', 's', 'ourselves', 'after', 'at', 'itself', 'isn', 'we',
'being', 'y', 'have', 'o', 'wasn', 'they', 'under', "isn't", "that'll", 'off',
'any', 'that', 'up', 'further', "shouldn't", 'who', "doesn't", 'll', 'from',
'not', 'whom', 'own', 'yourselves', 'doesn', 're', 'himself', 'was', 'the',
'no', 'shouldn', 'just', 'she', 'needn', 'it', 'my', 'you', 'wouldn', 'before',
'themselves', 'both', 'how', 'didn', 'to', 'those', 'been', 'now', 'hadn', 'ma',
'then', 'he', 'once', 'over', "wouldn't", 'its', 'so', "wasn't", 'again', 'if',
'can', 'hers', 'm', 'were', 'had', 'of', 'which', "you'd", "needn't", 'nor',
'me', 'our', 'very', 'there', "mustn't", 'in', 'doing', 'haven', 'a', 'through',
"should've", 'same', 'hasn', 'more', "haven't", 'ain', "didn't", 'weren',
"you've", 'myself', 'than', 'above', 'did', 'this', 'some', 'ours', 'here',
'on', 'her', 'are', "couldn't", "mightn't", "don't", 'herself', 'and', 'for',
'aren', "hadn't", 'or', 'be', 'against', 'what', 'other', 've', 'should', 'few'}
Tokenized corpus without stopwords: ['NLP drives computer programs that
translate text from one language to another, respond to spoken commands, and
summarize large volumes of text rapidly—even in real time. There's a good chance
you've interacted with NLP in the form of voice-operated GPS systems, digital
assistants, speech-to-text dictation software, customer service chatbots, and
other consumer conveniences. But NLP also plays a growing role in enterprise
solutions that help streamline business operations, increase employee
productivity, and simplify mission-critical business processes.']
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/pushpakulkarni/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/pushpakulkarni/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
```

[4]:

```
print("Len of Corpus before stopword removal: ",len(corpus))
print("Len of Corpus after stopword removal: ",len(corpus2))
```

```
Len of Corpus before stopword removal:  1
Len of Corpus after stopword removal:  1
```

[5]:

```
corpusn = "NLP drives computer programs that translate text from one language␣
 ↪to another, respond to spoken commands, and summarize large volumes of text␣
 ↪rapidly-even in real time. There’s a good chance you’ve interacted with NLP␣
 ↪in the form of voice-operated GPS systems, digital assistants,␣
 ↪speech-to-text dictation software, customer service chatbots, and other␣
 ↪consumer conveniences. But NLP also plays a growing role in enterprise␣
 ↪solutions that help streamline business operations, increase employee␣
 ↪productivity, and simplify mission-critical business processes."
values = corpusn.split()
values = [i for i in values if not i in stop_words_nltk]

values
```

[5]:

```
['NLP',
 'drives',
 'computer',
 'programs',
 'translate',
 'text',
 'one',
 'language',
 'another,',
 'respond',
 'spoken',
 'commands,',
 'summarize',
 'large',
 'volumes',
 'text',
 'rapidly-even',
 'real',
 'time.',
 'There’s',
 'good',
 'chance',
 'you’ve',
 'interacted',
 'NLP',
 'form',
 'voice-operated',
 'GPS',
 'systems,',
 'digital',
 'assistants,',
 'speech-to-text',
 'dictation',
 'software,',
 'customer',
 'service',
 'chatbots,',
 'consumer',
 'conveniences.',
 'But',
 'NLP',
 'also',
 'plays',
 'growing',
 'role',
 'enterprise',
 'solutions',
 'help',
 'streamline',
 'business',
 'operations,',
 'increase',
 'employee',
 'productivity,',
 'simplify',
 'mission-critical',
 'business',
 'processes.']
```

[6]:

```
corpus2
```

[6]:

```
['NLP drives computer programs that translate text from one language to another,
respond to spoken commands, and summarize large volumes of text rapidly-even in
real time. There’s a good chance you’ve interacted with NLP in the form of
voice-operated GPS systems, digital assistants, speech-to-text dictation
software, customer service chatbots, and other consumer conveniences. But NLP
also plays a growing role in enterprise solutions that help streamline business
operations, increase employee productivity, and simplify mission-critical
business processes.']
```

[7]:

```
#Label Encoding
label_encoder = LabelEncoder()
integer_encoded = label_encoder.fit_transform(values)
print("Label Encoded:",integer_encoded)




#One-Hot Encoding
onehot_encoder = OneHotEncoder()
onehot_encoded = onehot_encoder.fit_transform([corpus2]).toarray()
print("Onehot Encoded Matrix:\n",onehot_encoded)
```

```
Label Encoded: [ 2 17 11 34 50 48 29 26  5 37 44 10 46 27 52 48 35 36 49  3 21
8 53 25
  2 20 51  1 47 16  6 43 15 41 14 39  9 12 13  0  2  4 31 22 38 19 42 23
 45  7 30 24 18 33 40 28  7 32]
Onehot Encoded Matrix:
 [[1.]]
```

[8]:

```
#BOW
processed_docs = [doc.lower().replace(".","") for doc in corpus2]
processed_docs
```

[8]:

```
['nlp drives computer programs that translate text from one language to another,
respond to spoken commands, and summarize large volumes of text rapidly-even in
real time there’s a good chance you’ve interacted with nlp in the form of voice-
operated gps systems, digital assistants, speech-to-text dictation software,
customer service chatbots, and other consumer conveniences but nlp also plays a
growing role in enterprise solutions that help streamline business operations,
increase employee productivity, and simplify mission-critical business
processes']
```

[9]:

```
from sklearn.feature_extraction.text import CountVectorizer
#look at the documents list
print("Our corpus: ", corpus2)
count_vect = CountVectorizer()
#Build a BOW representation for the corpus
bow_rep = count_vect.fit_transform(processed_docs)

#Look at the vocabulary mapping
print("Our vocabulary: ", count_vect.vocabulary_)
```

```
Our corpus:  ['NLP drives computer programs that translate text from one
language to another, respond to spoken commands, and summarize large volumes of
text rapidly-even in real time. There’s a good chance you’ve interacted with NLP
in the form of voice-operated GPS systems, digital assistants, speech-to-text
dictation software, customer service chatbots, and other consumer conveniences.
But NLP also plays a growing role in enterprise solutions that help streamline
business operations, increase employee productivity, and simplify mission-
critical business processes.']
Our vocabulary:  {'nlp': 32, 'drives': 16, 'computer': 9, 'programs': 41,
'that': 56, 'translate': 61, 'text': 55, 'from': 21, 'one': 34, 'language': 29,
'to': 60, 'another': 2, 'respond': 44, 'spoken': 51, 'commands': 8, 'and': 1,
'summarize': 53, 'large': 30, 'volumes': 64, 'of': 33, 'rapidly': 42, 'even':
19, 'in': 26, 'real': 43, 'time': 59, 'there': 58, 'good': 22, 'chance': 6,
'you': 66, 've': 62, 'interacted': 28, 'with': 65, 'the': 57, 'form': 20,
'voice': 63, 'operated': 35, 'gps': 23, 'systems': 54, 'digital': 15,
'assistants': 3, 'speech': 50, 'dictation': 14, 'software': 48, 'customer': 13,
'service': 46, 'chatbots': 7, 'other': 37, 'consumer': 10, 'conveniences': 11,
'but': 5, 'also': 0, 'plays': 38, 'growing': 24, 'role': 45, 'enterprise': 18,
'solutions': 49, 'help': 25, 'streamline': 52, 'business': 4, 'operations': 36,
'increase': 27, 'employee': 17, 'productivity': 40, 'simplify': 47, 'mission':
31, 'critical': 12, 'processes': 39}
```

[10]:
```
#see the BOW rep for the document
print("BoW representation : ", bow_rep[0].toarray())
```

```
BoW representation :  [[1 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1
1 1 1 1 3 2 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 2 1 1 1 3 1 1 1 1 1 1]]
```

[11]:
```
#Get the representation using this vocabulary, for a new text
temp = count_vect.transform(["NLP helps in language translation"])
print("Bow representation for 'NLP helps in language translation':", temp.
 ↪toarray())
```

```
Bow representation for 'NLP helps in language translation': [[0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 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]]
```

[12]:
```
from sklearn.feature_extraction.text import CountVectorizer

#Ngram vectorization example with count vectorizer and uni, bi, trigrams
count_vect = CountVectorizer(ngram_range=(1,3))

#Build a BOW representation for the corpus
bow_rep = count_vect.fit_transform(processed_docs)

#Look at the vocabulary mapping
print("Our vocabulary: ", count_vect.vocabulary_)

#see the BOW rep for first 2 documents
print("BoW representation: ", bow_rep[0].toarray())

#Get the representation using this vocabulary, for a new text
temp = count_vect.transform(["NLP helps in language translation"])

print("Bow representation for 'NLP helps in language translation':", temp.
 ↪toarray())
```

```
Our vocabulary:  {'nlp': 105, 'drives': 53, 'computer': 32, 'programs': 136,
'that': 185, 'translate': 206, 'text': 178, 'from': 68, 'one': 117, 'language':
96, 'to': 199, 'another': 10, 'respond': 145, 'spoken': 166, 'commands': 29,
'and': 3, 'summarize': 172, 'large': 99, 'volumes': 215, 'of': 112, 'rapidly':
139, 'even': 62, 'in': 83, 'real': 142, 'time': 196, 'there': 193, 'good': 71,
'chance': 23, 'you': 221, 've': 209, 'interacted': 93, 'with': 218, 'the': 190,
'form': 65, 'voice': 212, 'operated': 120, 'gps': 74, 'systems': 175, 'digital':
50, 'assistants': 13, 'speech': 163, 'dictation': 47, 'software': 157,
'customer': 44, 'service': 151, 'chatbots': 26, 'other': 126, 'consumer': 35,
'conveniences': 38, 'but': 20, 'also': 0, 'plays': 129, 'growing': 77, 'role':
148, 'enterprise': 59, 'solutions': 160, 'help': 80, 'streamline': 169,
'business': 16, 'operations': 123, 'increase': 90, 'employee': 56,
'productivity': 133, 'simplify': 154, 'mission': 102, 'critical': 41,
'processes': 132, 'nlp drives': 108, 'drives computer': 54, 'computer programs':
33, 'programs that': 137, 'that translate': 188, 'translate text': 207, 'text
from': 181, 'from one': 69, 'one language': 118, 'language to': 97, 'to
another': 200, 'another respond': 11, 'respond to': 146, 'to spoken': 202,
'spoken commands': 167, 'commands and': 30, 'and summarize': 8, 'summarize
large': 173, 'large volumes': 100, 'volumes of': 216, 'of text': 113, 'text
rapidly': 183, 'rapidly even': 140, 'even in': 63, 'in real': 86, 'real time':
143, 'time there': 197, 'there good': 194, 'good chance': 72, 'chance you': 24,
'you ve': 222, 've interacted': 210, 'interacted with': 94, 'with nlp': 219,
'nlp in': 110, 'in the': 88, 'the form': 191, 'form of': 66, 'of voice': 115,
'voice operated': 213, 'operated gps': 121, 'gps systems': 75, 'systems
digital': 176, 'digital assistants': 51, 'assistants speech': 14, 'speech to':
164, 'to text': 204, 'text dictation': 179, 'dictation software': 48, 'software
customer': 158, 'customer service': 45, 'service chatbots': 152, 'chatbots and':
27, 'and other': 4, 'other consumer': 127, 'consumer conveniences': 36,
'conveniences but': 39, 'but nlp': 21, 'nlp also': 106, 'also plays': 1, 'plays
growing': 130, 'growing role': 78, 'role in': 149, 'in enterprise': 84,
'enterprise solutions': 60, 'solutions that': 161, 'that help': 186, 'help
streamline': 81, 'streamline business': 170, 'business operations': 17,
'operations increase': 124, 'increase employee': 91, 'employee productivity':
57, 'productivity and': 134, 'and simplify': 6, 'simplify mission': 155,
'mission critical': 103, 'critical business': 42, 'business processes': 19, 'nlp
drives computer': 109, 'drives computer programs': 55, 'computer programs that':
34, 'programs that translate': 138, 'that translate text': 189, 'translate text
from': 208, 'text from one': 182, 'from one language': 70, 'one language to':
119, 'language to another': 98, 'to another respond': 201, 'another respond to':
12, 'respond to spoken': 147, 'to spoken commands': 203, 'spoken commands and':
168, 'commands and summarize': 31, 'and summarize large': 9, 'summarize large
volumes': 174, 'large volumes of': 101, 'volumes of text': 217, 'of text
rapidly': 114, 'text rapidly even': 184, 'rapidly even in': 141, 'even in real':
64, 'in real time': 87, 'real time there': 144, 'time there good': 198, 'there
good chance': 195, 'good chance you': 73, 'chance you ve': 25, 'you ve
interacted': 223, 've interacted with': 211, 'interacted with nlp': 95, 'with
nlp in': 220, 'nlp in the': 111, 'in the form': 89, 'the form of': 192, 'form of
voice': 67, 'of voice operated': 116, 'voice operated gps': 214, 'operated gps
systems': 122, 'gps systems digital': 76, 'systems digital assistants': 177,
'digital assistants speech': 52, 'assistants speech to': 15, 'speech to text':
165, 'to text dictation': 205, 'text dictation software': 180, 'dictation
software customer': 49, 'software customer service': 159, 'customer service
chatbots': 46, 'service chatbots and': 153, 'chatbots and other': 28, 'and other
consumer': 5, 'other consumer conveniences': 128, 'consumer conveniences but':
37, 'conveniences but nlp': 40, 'but nlp also': 22, 'nlp also plays': 107, 'also
plays growing': 2, 'plays growing role': 131, 'growing role in': 79, 'role in
```

```
enterprise': 150, 'in enterprise solutions': 85, 'enterprise solutions that':
61, 'solutions that help': 162, 'that help streamline': 187, 'help streamline
business': 82, 'streamline business operations': 171, 'business operations
increase': 18, 'operations increase employee': 125, 'increase employee
productivity': 92, 'employee productivity and': 58, 'productivity and simplify':
135, 'and simplify mission': 7, 'simplify mission critical': 156, 'mission
critical business': 104, 'critical business processes': 43}
BoW representation:  [[1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1
  1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1
  1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
  1 1 1 1 1 1 1 1]]
Bow representation for 'NLP helps in language translation': [[0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 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 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0]]
```

[13]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer()
bow_rep_tfidf = tfidf.fit_transform(processed_docs)

#IDF for all words in the vocabulary
print("IDF for all words in the vocabulary",tfidf.idf_)
print("-"*10)
#All words in the vocabulary.
print("All words in the vocabulary",tfidf.get_feature_names())
print("-"*10)

#TFIDF representation for all documents in our corpus
print("TFIDF representation for  our corpus\n",bow_rep_tfidf.toarray())
print("-"*10)

temp = tfidf.transform(["NLP with ML helps in language processing"])
print("Tfidf representation for 'NLP with ML helps in language processing':\n",␣
 ↪temp.toarray())
```

```
IDF for all words in the vocabulary [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

```
----------
All words in the vocabulary ['also', 'and', 'another', 'assistants', 'business',
'but', 'chance', 'chatbots', 'commands', 'computer', 'consumer', 'conveniences',
'critical', 'customer', 'dictation', 'digital', 'drives', 'employee',
'enterprise', 'even', 'form', 'from', 'good', 'gps', 'growing', 'help', 'in',
'increase', 'interacted', 'language', 'large', 'mission', 'nlp', 'of', 'one',
'operated', 'operations', 'other', 'plays', 'processes', 'productivity',
'programs', 'rapidly', 'real', 'respond', 'role', 'service', 'simplify',
'software', 'solutions', 'speech', 'spoken', 'streamline', 'summarize',
'systems', 'text', 'that', 'the', 'there', 'time', 'to', 'translate', 've',
'voice', 'volumes', 'with', 'you']
----------
TFIDF representation for  our corpus
 [[0.09284767 0.27854301 0.09284767 0.09284767 0.18569534 0.09284767
  0.09284767 0.09284767 0.09284767 0.09284767 0.09284767 0.09284767
  0.09284767 0.09284767 0.09284767 0.09284767 0.09284767 0.09284767
  0.09284767 0.09284767 0.09284767 0.09284767 0.09284767 0.09284767
  0.09284767 0.09284767 0.27854301 0.09284767 0.09284767 0.09284767
  0.09284767 0.09284767 0.27854301 0.18569534 0.09284767 0.09284767
  0.09284767 0.09284767 0.09284767 0.09284767 0.09284767 0.09284767
  0.09284767 0.09284767 0.09284767 0.09284767 0.09284767 0.09284767
  0.09284767 0.09284767 0.09284767 0.09284767 0.09284767 0.09284767
  0.09284767 0.27854301 0.18569534 0.09284767 0.09284767 0.09284767
  0.27854301 0.09284767 0.09284767 0.09284767 0.09284767 0.09284767
  0.09284767]]
----------
Tfidf representation for 'NLP with ML helps in language processing':
 [[0.  0.  0.  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.  0.5 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.  0.5 0. ]]

/Users/pushpakulkarni/opt/anaconda3/lib/python3.9/site-
packages/sklearn/utils/deprecation.py:87: FutureWarning: Function
get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will
be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

9 Conclsuion

1) Thus we can infer that TF-IDF is more effective and better for ML models as it has less sparsity.

2) The reason for why TF-IDF is better is because BoW & Bag of n grams they might capture some context and be more readable, but their feature vector has huge sparsity. And for ML models we need least number of dimensions for efficient processing.

3) Even though text cleaning, pre processing, stop words removal is done, the feature vectors of

OHE, LE, BoW,Bag of n grams are sparse because it gives all words in a text equal importance. Whereas TF-IDF makes rare words more prominent and effectively ignores common words. It is closely related to frequency-based filters but much more mathematically elegant than placing hard cutoff thresholds.This makes the feature vector a bit less sparse and thus gives better performance while training the model.