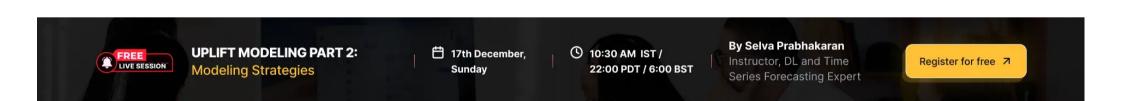


Courses ~

Search

Search

Login



101 NLP Exercises (using modern libraries)

by Shrivarsheni

Natural language processing is the technique by which AI understands human language. NLP tasks such as text classification, summarization, sentiment analysis, translation are widely used. This post aims to serve as a reference for basic and advanced NLP tasks.



101 NLP Exercises using modern libraries. Photo by Ana Justin Luebke.

1. Import nitk and download the 'stopwords' and 'punkt' packages

Difficulty Level: L1

Q. Import nltk and necessary packages

Show Solution >

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

#> [nltk_data] Downloading package punkt to
/root/nltk_data...
#> [nltk_data] Unzipping tokenizers/punkt.zip.
#> [nltk_data] Error loading stop: Package 'stop'
not found in index
#> [nltk_data] Downloading package stopwords to
/root/nltk_data...
#> [nltk_data] Unzipping corpora/stopwords.zip.
#> True
```

2. Import spacy and load the language model

Difficulty Level: L1

Q. Import spacy library and load 'en_core_web_sm' model for english language. Load 'xx_ent_wiki_sm' for multi language support.

Show Solution ✓

```
# Import and load model

import spacy
nlp=spacy.load("en_core_web_sm")
nlp
# More models here: https://spacy.io/models
```

```
#> <spacy.lang.en.English at 0x7facaf6cd0f0>
```

3. How to tokenize a given text?

Difficulty Level: L1

Q. Print the tokens of the given text document

Input:

```
text="Last week, the University of Cambridge shared its own
research that shows if everyone wears a mask outside
home, dreaded 'second wave' of the pandemic can be avoided."
```

Desired Output:

```
Last
week

the
University
of
Cambridge
shared
...(truncated)...
```

```
# Tokeniation with nltk
tokens=nltk.word_tokenize(text)
for token in tokens:
    print(token)
```

```
# Tokenization with spaCy
nlp=spacy.load("en_core_web_sm")
doc=nlp(text)
for token in doc:
    print(token.text)
```

4. How to get the sentences of a text document?

Difficulty Level: L1



Q. Print the sentences of the given text document

Input:

lace in the company of others. Within this context of physical threat, social

Desired Output:

The outbreak of coronavirus disease 2019 (COVID-19) has created Not only the rate of contagion and patterns of transmission three Within this context of physical threat, social and physical dist ... (truncated)...

```
# Tokenizing the text into sentences with spaCy
doc=nlp(text)
for sentence in doc.sents:
   print(sentence)
   print(' ')
```

- #> The outbreak of coronavirus disease 2019 (COVID-19) has created a global health crisis that has had a deep impact on the way we perceive our world and our everyday lives.
- #> Not only the rate of contagion and patterns of transmission threatens our sense of agency, but the safety measures put in place to contain the spread of the virus also require social distancing by refraining from doing what is inherently human, which is to find solace in the company of others.
- #> Within this context of physical threat, social and physical
 distancing, as well as public alarm, what has been (and can be)
- #> the role of the different mass media channels in our lives on individual, social and societal levels?
- #> Mass media have long been recognized as powerful forces shaping how we experience the world and ourselves.
- #> This recognition is accompanied by a growing volume of research, that closely follows the footsteps of technological transformations (e.g. radio, movies, television, the internet,

101 NLP Exercises (using modern libraries) - Machine Learning Plus

mobiles) and the zeitgeist (e.g. cold war, 9/11, climate change) in an attempt to map mass media major impacts on how we perceive ourselves, both as individuals and citizens.

#> Are media (broadcast and digital) still able to convey a
sense of unity reaching large audiences, or are messages lost
in the noisy crowd of mass self-communication?

Extracting sentences with nltk
nltk.sent tokenize(text)

#> ['The outbreak of coronavirus disease 2019 (COVID-19) has created a global health crisis that has had a deep impact on the way we perceive our world and our everyday lives.', #> 'Not only the rate of contagion and patterns of transmission threatens our sense of agency, but the safety measures put in place to contain the spread of the virus also require social distancing by refraining from doing what is inherently human, which is to find solace in the company of others.', #> 'Within this context of physical threat, social and physical distancing, as well as public alarm, what has been (and can be) the role of the different mass media channels in our lives on individual, social and societal levels?', #> 'Mass media have long been recognized as powerful forces shaping how we experience the world and ourselves.', #> 'This recognition is accompanied by a growing volume of research, that closely follows the footsteps of technological transformations (e.g.', #> 'radio, movies, television, the internet, mobiles) and the zeitgeist (e.g.', #> 'cold war, 9/11, climate change) in an attempt to map mass media major impacts on how we perceive ourselves, both as individuals and citizens.', #> 'Are media (broadcast and digital) still able to convey a sense of unity reaching large audiences, or are messages lost in the noisy crowd of mass self-communication?']

5. How to tokenize a text using the 'transformers' package?

Difficulty Level: L1

Q. Tokenize the given text in encoded form using the tokenizer of Huggingface's transformer package.

Input:

text="I love spring season. I go hiking with my friends"

Desired Output:

```
[101, 1045, 2293, 3500, 2161, 1012, 1045, 2175, 13039, 2007, 202]
[CLS] i love spring season. i go hiking with my friends [SEP]
```

Show Solution >

Import tokenizer from transfromers
!pip install transformers
from transformers import AutoTokenizer

```
# Initialize the tokenizer
tokenizer=AutoTokenizer.from_pretrained('bert-base-uncased')

# Encoding with the tokenizer
inputs=tokenizer.encode(text)
print(inputs)
tokenizer.decode(inputs)
```

```
#> [101, 1045, 2293, 3500, 2161, 1012, 1045, 2175, 13039, 2007,
2026, 2814, 102]
#> [CLS] i love spring season. i go hiking with my friends
[SEP]
```

6. How to tokenize text with stopwords as delimiters?

Difficulty Level: L2

Q. Tokenize the given text with stop words ("is","the","was") as delimiters. Tokenizing this way identifies meaningful phrases. Sometimes, useful for topic modeling

Input:

```
text = "Walter was feeling anxious. He was diagnosed today. He
probably is the best person I know.""
```

Expected Output:

```
['Walter',
  'feeling anxious',
  'He',
  'diagnosed today',
  'He probably',
  'best person I know']
```

```
# Solution
text = "Walter was feeling anxious. He was diagnosed today. He
probably is the best person I know."
stop_words_and_delims = ['was', 'is', 'the', '.', ',', '-',
'!', '?']
for r in stop words and delims:
    text = text.replace(r, 'DELIM')
words = [t.strip() for t in text.split('DELIM')]
words filtered = list(filter(lambda a: a not in [''], words))
words_filtered
#> ['Walter',
#> 'feeling anxious',
#> 'He',
#> 'diagnosed today',
#> 'He probably',
#> 'best person I know']
```

7. How to remove stop words in a text?

Difficulty Level: L1

Q. Remove all the stopwords ('a', 'the', 'was'...) from the text

Input:

text="""the outbreak of coronavirus disease 2019 (COVID-19) has created a global health crisis that has had a deep impact on the way we perceive our world and our everyday lives. Not only the rate of contagion and patterns of transmission threatens our sense of agency, but the safety measures put in place to contain the spread of the virus also require social distancing by refraining from doing what is inherently human, which is to find solace in the company of others. Within this context of physical threat, social and physical distancing, as well as public alarm, what has been (and can be) the role of the different mass media channels in our lives on individual, social and societal levels? Mass media have long been recognized as powerful forces shaping how we experience the world and ourselves. This recognition is accompanied by a growing volume of research, that closely follows the footsteps of technological transformations (e.g. radio, movies, television, the internet, mobiles) and the zeitgeist (e.g. cold war, 9/11, climate change) in an attempt to map mass media major impacts on how we perceive ourselves, both as individuals and citizens. Are media (broadcast and digital) still able to convey a sense of unity reaching large audiences, or are messages lost in the noisy crowd of mass self-communication?

Desired Output:

'outbreak coronavirus disease 2019 (COVID-19) created global health crisis deep impact way perceive world everyday lives . rate contagion patterns transmission threatens sense agency , safety measures place contain spread virus require social distancing refraining inherently human , find solace company . context physical threat , social physical distancing , public alarm , () role different mass media channels lives individual , social societal levels ? Mass media long recognized powerful forces shaping experience world . recognition accompanied growing volume research , closely follows footsteps technological transformations (e.g. radio, movies, television , internet , mobiles) zeitgeist (e.g. cold war , 9/11 , climate change) attempt map mass media major impacts perceive , individuals citizens . media (broadcast digital) able convey sense unity reaching large audiences , messages lost noisy crowd mass self - communication ?'

```
# Method 1
# Removing stopwords in nltk

from nltk.corpus import stopwords
my_stopwords=set(stopwords.words('english'))
new_tokens=[]

# Tokenization using word_tokenize()
all_tokens=nltk.word_tokenize(text)

for token in all_tokens:
   if token not in my_stopwords:
        new_tokens.append(token)
" ".join(new_tokens)
```

#> 'outbreak coronavirus disease 2019 (COVID-19) created global health crisis deep impact way perceive world everyday lives . Not rate contagion patterns transmission threatens sense agency , safety measures put place contain spread virus also require social distancing refraining inherently human , find solace company others . Within context physical threat , social physical distancing , well public alarm , () role different mass media channels lives individual , social societal levels ? Mass media long recognized powerful forces shaping experience world . This recognition accompanied growing volume research , closely follows footsteps technological transformations (e.g . radio , movies , television , internet, mobiles) zeitgeist (e.g . cold war , 9/11 , climate change) attempt map mass media major impacts perceive , individuals citizens . Are media (broadcast digital) still able convey sense unity reaching large audiences , messages lost noisy crowd mass self-communication ?'

```
# Method 2
# Removing stopwords in spaCy

doc=nlp(text)
new_tokens=[]

# Using is_stop attribute of each token to check if it's a
stopword
for token in doc:
   if token.is_stop==False:
        new_tokens.append(token.text)

" ".join(new_tokens)
```

#> 'outbreak coronavirus disease 2019 (COVID-19) created global health crisis deep impact way perceive world everyday lives . rate contagion patterns transmission threatens sense agency , safety measures place contain spread virus require social distancing refraining inherently human , find solace company . context physical threat , social physical distancing , public alarm , () role different mass media channels lives individual , social societal levels ? Mass media long recognized powerful forces shaping experience world . recognition accompanied growing volume research , closely follows footsteps technological transformations (e.g. radio , movies , television , internet , mobiles) zeitgeist (e.g. cold war , 9/11 , climate change) attempt map mass media major impacts perceive , individuals citizens . media (broadcast digital) able convey sense unity reaching large audiences , messages lost noisy crowd mass self - communication ?'

8. How to add custom stop words in spaCy?

Difficulty Level: L1

Q. Add the custom stopwords "NIL" and "JUNK" in spaCy and remove the stopwords in below text

Input:

```
text=" Jonas was a JUNK great guy NIL Adam was evil NIL Martha
JUNK was more of a fool "
```

Expected Output:

'Jonas great guy Adam evil Martha fool'

```
# list of custom stop words
customize_stop_words = ['NIL','JUNK']

# Adding these stop words
for w in customize_stop_words:
    nlp.vocab[w].is_stop = True
doc = nlp(text)
tokens = [token.text for token in doc if not token.is_stop]
" ".join(tokens)
```

```
#> ' Jonas great guy Adam evil Martha fool'
```

9. How to remove punctuations?

Difficulty Level: L1

Q. Remove all the punctuations in the given text

Input:

```
text="The match has concluded !!! India has won the match . Will we fin the finals too ? !"
```

Desired Output:

```
'The match has concluded India has won the match Will we fin the finals too'
```

```
# Removing punctuations in spaCy

doc=nlp(text)
new_tokens=[]
# Check if a token is a punctuation through is_punct attribute
for token in doc:
    if token.is_punct==False:
        new_tokens.append(token.text)

" ".join(new_tokens)
```

```
#> 'The match has concluded India has won the match Will we fin
the finals too'
```

```
# Method 2
# Removing punctuation in nltk with RegexpTokenizer

tokenizer=nltk.RegexpTokenizer(r"\w+")

tokens=tokenizer.tokenize(text)
" ".join(tokens)
```

#> 'The match has concluded India has won the match Will we fin the finals too'

10. How to perform stemming

Difficulty Level: L2

Q. Perform stemming/ convert each token to it's root form in the given text

Input:

text= "Dancing is an art. Students should be taught dance as a subject in schools . I danced in many of my school function. Some people are always hesitating to dance."

Desired Output:

text= 'danc is an art . student should be taught danc as a subject in school . I danc in mani of my school function . some peopl are alway hesit to danc .'

Show Solution >

```
# Stemming with nltk's PorterStemmer
from nltk.stem import PorterStemmer
stemmer=PorterStemmer()
stemmed_tokens=[]
for token in nltk.word_tokenize(text):
  stemmed_tokens.append(stemmer.stem(token))
" ".join(stemmed_tokens)
```

#> 'danc is an art . student should be taught danc as a subject in school . I danc in mani of my school function . some peopl are alway hesit to danc .'

11. How to lemmatize a given text?

Difficulty Level: L2

Q. Perform lemmatzation on the given text

Hint: Lemmatization Approaches

Input:

text= "Dancing is an art. Students should be taught dance as a subject in schools . I danced in many of my school function. Some people are always hesitating to dance."

Desired Output:

text= 'dancing be an art . student should be teach dance as a subject in school . -PRON- dance in many of -PRON- school function . some people be always hesitate to dance .'

Show Solution >

```
# Lemmatization using spacy's lemma_ attribute of token
nlp=spacy.load("en_core_web_sm")
doc=nlp(text)

lemmatized=[token.lemma_ for token in doc]
" ".join(lemmatized)
```

#> 'dancing be an art . student should be teach dance as a subject in school . -PRON- dance in many of -PRON- school function . some people be always hesitate to dance .'

12. How to extract usernames from emails?

Difficulty Level: L2

Q. Extract the usernames from the email addresses present in the text

Input:

text= "The new registrations are potter709@gmail.com ,
elixir101@gmail.com. If you find any disruptions, kindly
contact granger111@gamil.com or severus77@gamil.com "

Desired Output:

```
['potter709', 'elixir101', 'granger111', 'severus77']
```

```
# Using regular expression to extract usernames
import re

# \S matches any non-whitespace character
# @ for as in the Email
# + for Repeats a character one or more times
usernames= re.findall('(\S+)@', text)
print(usernames)
```

```
#> ['potter709', 'elixir101', 'granger111', 'severus77']
```

13. How to find the most common words in the text excluding stopwords

Difficulty Level: L2

Q. Extract the top 10 most common words in the given text excluding stopwords.

Input:

text="""Junkfood - Food that do no good to our body. And there's no need of them in our body but still we willingly eat them because they are great in taste and easy to cook or ready to eat. Junk foods have no or very less nutritional value and irrespective of the way they are marketed, they are not healthy to consume. The only reason of their gaining popularity and increased trend of consumption is

that they are ready to eat or easy to cook foods. People, of all age groups are moving towards Junkfood as it is hassle free and often ready to grab and eat. Cold drinks, chips, noodles, pizza, burgers, French fries etc. are few examples from the great variety of junk food available in the market.

Junkfood is the most dangerous food ever but it is pleasure in eating and it gives a great taste in mouth examples of Junkfood are kurkure and chips.. cold rings are also source of junk food... they shud nt be ate in high amounts as it results fatal to our body... it cn be eated in a limited extend ... in research its found tht ths junk foods r very dangerous fr our health

Junkfood is very harmful that is slowly eating away the health of the present generation. The term itself denotes how dangerous it is for our bodies. Most importantly, it tastes so good that people consume it on a daily basis. However, not much awareness is spread about the harmful effects of Junkfood. The problem is more serious than you think. Various studies show that Junkfood impacts our health negatively. They contain higher levels of calories, fats, and sugar. On the contrary, they have very low amounts of healthy nutrients and lack dietary fibers. Parents must discourage their children from consuming junk food because of the ill effects it has on one's health.

Junkfood is the easiest way to gain unhealthy weight. The amount of fats and sugar in the food makes you gain weight rapidly. However, this is not a healthy weight. It is more of fats and cholesterol which will have a harmful impact on your health. Junk food is also one of the main reasons for the increase in obesity nowadays.

This food only looks and tastes good, other than that, it has no positive points. The amount of calorie your body requires to stay fit is not fulfilled by this food. For instance, foods like French fries, burgers, candy, and cookies, all have high amounts of sugar and fats. Therefore, this can result in long-term illnesses like diabetes and high blood pressure. This may also result in kidney failure."""

Desired Output:

```
text= {Junkfood: 10,
  food: 8,
  good: 5,
  harmful: 3
  body: 1,
  need: 1,
  ...(truncated)
```

```
# Creating spacy doc of the text
nlp=spacy.load("en_core_web_sm")
doc=nlp(text)
```

```
# Removal of stop words and punctuations
words=[token for token in doc if token.is_stop==False and
token.is_punct==False]

freq_dict={}

# Calculating frequency count
for word in words:
    if word not in freq_dict:
        freq_dict[word]=1
    else:
        freq_dict[word]+=1

freq_dict
```

```
{Junkfood: 10,
  food: 8,
  good: 5,
  harmful : 3
  body: 1,
  need: 1,
  ...(truncated)
```

14. How to do spell correction in a given text

?

Difficulty Level: L2

Q. Correct the spelling errors in the following text

Input:

```
text="He is a gret person. He beleives in bod"
```

Desired Output:

```
text="He is a great person. He believes in god"
```

Show Solution >

```
# Import textblob
from textblob import TextBlob

# Using textblob's correct() function
text=TextBlob(text)
print(text.correct())
#> He is a great person. He believes in god
```

15. How to tokenize tweets?

Difficulty Level: L2

Q. Clean the following tweet and tokenize them

Input:

```
text=" Having lots of fun #goa #vaction #summervacation. Fancy
dinner @Beachbay restro :) "
```

Desired Output:

```
['Having',
  'lots',
  'of',
  'fun',
  'goa',
  'vaction',
  'summervacation',
  'Fancy',
  'dinner',
  'Beachbay',
  'restro']
```

Show Solution >

```
import re
# Cleaning the tweets
text=re.sub(r'[^\w]', ' ', text)

# Using nltk's TweetTokenizer
from nltk.tokenize import TweetTokenizer
tokenizer=TweetTokenizer()
tokenizer.tokenize(text)
```

```
#> ['Having',
#> 'lots',
#> 'of',
#> 'fun',
#> 'goa',
#> 'vaction',
#> 'summervacation',
#> 'Fancy',
#> 'dinner',
#> 'Beachbay',
#> 'restro']
```

16. How to extract all the nouns in a text?

Difficulty Level: L2

Q. Extract and print all the nouns present in the below text

Input:

```
text="James works at Microsoft. She lives in manchester and
likes to play the flute"
```

```
James
Microsoft
manchester
flute
```

```
# Coverting the text into a spacy Doc
nlp=spacy.load("en_core_web_sm")
doc=nlp(text)

# Using spacy's pos_ attribute to check for part of speech tags
for token in doc:
    if token.pos_=='NOUN' or token.pos_=='PROPN':
        print(token.text)

#> James
#> Microsoft
#> manchester
#> flute
```

17. How to extract all the pronouns in a text?

Difficulty Level: L2

Q. Extract and print all the pronouns in the text

Input:

Desired Output:

```
He
He
She
```

```
text="John is happy finally. He had landed his dream job
finally. He told his mom. She was elated "
```

Show Solution >

```
# Using spacy's pos_ attribute to check for part of speech tags
nlp=spacy.load("en_core_web_sm")
doc=nlp(text)

for token in doc:
   if token.pos_=='PRON':
        print(token.text)

#> He
#> He
#> She
```

18. How to find similarity between two words?

Find the similarity between any two words.

Input:

```
word1="amazing"
word2="terrible"
word3="excellent"
```

Desired Output:

```
#> similarity between amazing and terrible is 0.4618907134376460
#> similarity between amazing and excellent is 0.63882070867377
```

Show Solution >

```
# Convert words into spacy tokens
import spacy
!python -m spacy download en_core_web_lg
nlp=spacy.load('en core web lg')
token1=nlp(word1)
token2=nlp(word2)
token3=nlp(word3)
# Use similarity() function of tokens
print('similarity between', word1, 'and', word2, 'is'
, token1.similarity(token2))
print('similarity between', word1, 'and', word3, 'is'
, token1.similarity(token3))
#> similarity between amazing and terrible is
0.46189071343764604
#> similarity between amazing and excellent is
0.6388207086737778
```

19. How to find similarity between two documents?

Difficulty Level: L2

Q. Find the similarity between any two text documents

Input:

```
text1="John lives in Canada"
text2="James lives in America, though he's not from there"
```

Desired Output:

```
0.792817083631068
```

```
# Finding similarity using spaCy library

doc1=nlp(text1)
doc2=nlp(text2)
doc1.similarity(doc2)
```

```
#> 0.792817083631068
```

20. How to find the cosine similarity of two documents?

Difficulty Level: L3

Q. Find the cosine similarity between two given documents

Input

```
text1='Taj Mahal is a tourist place in India'
text2='Great Wall of China is a tourist place in china'
```

Desired Output:

```
[[1. 0.45584231]
[0.45584231 1. ]]
```

Show Solution >

To understand more about the maths behind this or for similar problems, check this post https://www.machinelearningplus.com/nlp/cosine-similarity/

21. How to find soft cosine similarity of documents?

Difficulty Level: L3

Q. Compute the soft cosine similarity of the given documents

Hint: Soft Cosine Similarity

Input:

```
doc soup = "Soup is a primarily liquid food, generally served
warm or hot (but may be cool or cold), that is made by
combining ingredients of meat or vegetables with stock, juice,
water, or another liquid. "
doc noodles = "Noodles are a staple food in many cultures. They
are made from unleavened dough which is stretched, extruded, or
rolled flat and cut into one of a variety of shapes."
doc dosa = "Dosa is a type of pancake from the Indian
subcontinent, made from a fermented batter. It is somewhat
similar to a crepe in appearance. Its main ingredients are rice
and black gram."
doc trump = "Mr. Trump became president after winning the
political election. Though he lost the support of some
republican friends, Trump is friends with President Putin"
doc election = "President Trump says Putin had no political
interference is the election outcome. He says it was a
witchhunt by political parties. He claimed President Putin is a
friend who had nothing to do with the election"
doc putin = "Post elections, Vladimir Putin became President of
Russia. President Putin had served as the Prime Minister
earlier in his political career"
```

Desired Output:

```
0.5842470477718544
```

Show Solution >

```
# Prepare a dictionary and a corpus.
dictionary = corpora.Dictionary([simple preprocess(doc) for doc
in documents])
# Prepare the similarity matrix
similarity matrix =
fasttext_model300.similarity_matrix(dictionary, tfidf=None,
threshold=0.0, exponent=2.0, nonzero_limit=100)
# Convert the sentences into bag-of-words vectors.
sent_1 = dictionary.doc2bow(simple_preprocess(doc_trump))
sent 2 = dictionary.doc2bow(simple preprocess(doc_election))
sent 3 = dictionary.doc2bow(simple preprocess(doc putin))
sent 4 = dictionary.doc2bow(simple preprocess(doc soup))
sent_5 = dictionary.doc2bow(simple_preprocess(doc_noodles))
sent 6 = dictionary.doc2bow(simple preprocess(doc dosa))
sentences = [sent_1, sent_2, sent_3, sent_4, sent_5, sent_6]
# Compute soft cosine similarity
print(softcossim(sent 1, sent 2, similarity matrix))
#> 0.5842470477718544
```

Difficulty Level: L2

22. How to find similar words using pretrained Word2Vec?

Q. Find all similiar words to "amazing" using Google news Word2Vec.

Desired Output:

```
#> [('incredible', 0.90),
#> ('awesome', 0.82),
#> ('unbelievable', 0.82),
#> ('fantastic', 0.77),
#> ('phenomenal', 0.76),
#> ('astounding', 0.73),
#> ('wonderful', 0.72),
#> ('unbelieveable', 0.71),
#> ('remarkable', 0.70),
#> ('marvelous', 0.70)]
```

Show Solution >

```
# Import gensim api
import gensim.downloader as api
# Load the pretrained google news word2vec model
word2vec_model300 = api.load('word2vec-google-news-300')
# Using most similar() function
word2vec model300.most similar('amazing')
#> [('incredible', 0.9054000973701477),
#> ('awesome', 0.8282865285873413),
#> ('unbelievable', 0.8201264142990112),
#> ('fantastic', 0.778986930847168),
#> ('phenomenal', 0.7642048001289368),
#> ('astounding', 0.7347068786621094),
#> ('wonderful', 0.7263179421424866),
#> ('unbelieveable', 0.7165080904960632),
#> ('remarkable', 0.7095627188682556),
#> ('marvelous', 0.7015583515167236)]
```

23. How to compute Word mover distance?

Difficulty Level: L3

Q. Compute the word mover distance between given two texts

Input:

```
sentence_orange = 'Oranges are my favorite fruit'
sent="apples are not my favorite"
```

Desired Output:

```
5.378
```

```
# Impting gensim 's Word2Vec model
import gensim

from gensim.models import Word2Vec
model=Word2Vec()

sentence_orange = 'Oranges are my favorite fruit'
sent="apples are not my favorite"

# Computing the word mover distance
distance = model.wmdistance(sent, sentence_orange)

#> 5.378
```

24. How to replace all the pronouns in a text with their respective object names

Difficulty Level: L2

Q. Replace the pronouns in below text by the respective object nmaes

Input:

```
text=" My sister has a dog and she loves him"
```

Desired Output:

```
[My sister, she]
[a dog , him ]
```

Show Solution >

```
# Import neural coref library
!pip install neuralcoref
import spacy
import neuralcoref

# Add it to the pipeline
nlp = spacy.load('en')
neuralcoref.add_to_pipe(nlp)

# Printing the coreferences
doc1 = nlp('My sister has a dog. She loves him.')
print(doc1._.coref_clusters)
```

spaCy also provides the feature of visualizing the coreferences. Check out this https://spacy.io/universe/project/neuralcoref-vizualizer/.

25. How to extract topic keywords using LSA?

Difficulty Level: L3

Q. Extract the topic keywords from the given texts using LSA(Latent Semantic Analysis)

Input:

texts= ["""It's all about travel. I travel a lot. those who do not travel read only a page." - said Saint Augustine. He was a great travel person. Travelling can teach you more than any university course. You learn about the culture of the country you visit. If you talk to locals, you will likely learn about their thinking, habits, traditions and history as well. If you travel, you will not only learn about foreign cultures, but about your own as well. You will notice the cultural differences, and will find out what makes your culture unique. After retrurning from a long journey, you will see your country with new eyes.""",

""" You can learn a lot about yourself through travelling. You can observe how you feel beeing far from your country. You will find out how you feel about your homeland. You should travel You will realise how you really feel about foreign people. You will find out how much you know/do not know about the world. You will be able to observe how you react in completely new situations. You will test your language, orientational and social skills. You will not be the same person after returning home. During travelling you will meet people that are very different from you. If you travel enough, you will learn to accept and appreciate these differences. Traveling makes you more open and accepting.""",

"""Some of my most cherished memories are from the times when I was travelling. If you travel, you can experience things that you could never experience at home. You may see beautiful places and landscapes that do not exist where you live. You may meet people that will change your life, and your thingking. You may try activities that you have never tried before. Travelling will inevitably make you more independent and confident. You will realise that you can cope with a lot of unexpected situations. You will realise that you can survive without all that help that is always available for you at home. You will likely find out that you are much stronger and braver than you have expected.""",

"""If you travel, you may learn a lot of useful things. These things can be anything from a new recepie, to a new, more effective solution to an ordinary problem or a new way of creating something. Even if you go to a country where they speak the same language as you, you may still learn some new words and expressions that are only used there. If you go to a country where they speak a different language, you will learn even more."",

"""After arriving home from a long journey, a lot of travellers experience that they are much more motivated than they were before they left. During your trip you may learn things that you will want to try at home as well. You may want to test your new skills and knowledge. Your experiences will give you a lot of energy. During travelling you may experience the craziest, most exciting things, that will eventually become great stories that you can tell others. When you grow old and look back at your life and all your travel experiences, you will realise how much you have done in your life and your life was not in vain. It can provide you with happiness and satisfaction for the rest of your life.""",

"""The benefits of travel are not just a one-time thing: travel changes you physically and psychologically. Having little time or money isn't a valid excuse. You can travel for cheap very easily. If you have a full-time job and a family, you can still travel on the weekends or holidays, even with a baby. travel more is likely to have a tremendous impact on your mental well-being, especially if you're no used to going out of your comfort zone. Trust me: travel more and your doctor will be happy. Be sure to get in touch with your physician, they might recommend some medication to accompany you in your travels, especially if you're heading to regions of the globe with potentially dangerous diseases.""",

"""Sure, you probably feel comfortable where you are, but that is just a fraction of the world! If you are a student, take advantage of programs such as Erasmus to get to know more people, experience and understand their culture. Dare traveling to regions you have a skeptical opinion about. I bet that you will change your mind and realize that everything is not so bad abroad.""",

""" So, travel makes you cherish life. Let's travel more . Share your travel diaries with us too"""

```
101 NLP Exercises (using modern libraries) - Machine Learning Plus
#> learn new life travelling country feel
#> Topic 1:
#> life cherish diaries let share experience
#> Topic 2:
#> feel know time people just regions
#> Topic 3:
#> time especially cherish diaries let share
.. (truncated)..
```

```
# Importing the Tf-idf vectorizer from sklearn
from sklearn.feature extraction.text import TfidfVectorizer
# Defining the vectorizer
vectorizer = TfidfVectorizer(stop words='english',
max_features= 1000, max_df = 0.5, smooth_idf=True)
# Transforming the tokens into the matrix form through
.fit transform()
matrix= vectorizer.fit transform(texts)
# SVD represent documents and terms in vectors
from sklearn.decomposition import TruncatedSVD
SVD model = TruncatedSVD(n_components=10,
algorithm='randomized', n_iter=100, random_state=122)
SVD model.fit(matrix)
# Getting the terms
terms = vectorizer.get_feature_names()
# Iterating through each topic
for i, comp in enumerate(SVD model.components):
   terms comp = zip(terms, comp)
    # sorting the 7 most important terms
    sorted terms = sorted(terms comp, key= lambda x:x[1],
reverse=True)[:7]
    print("Topic "+str(i)+": ")
    # printing the terms of a topic
    for t in sorted terms:
       print(t[0],end=' ')
    print(' ')
#> Topic 0:
#> learn new life travelling country feel
#> Topic 1:
#> life cherish diaries let share experience
#> Topic 2:
#> feel know time people just regions
#> Topic 3:
#> time especially cherish diaries let share
#> Topic 4:
#> cherish diaries let share makes feel
#> Topic 5:
#> culture augustine course cultural cultures eyes
#> Topic 6:
#> want experiences life things advantage bad
#> Topic 7:
#> observe feel experiences want skills test
```

26. How to extract topic Keywords using LDA

?

Difficulty Level: L3

Q. Extract the the topics from the given texts with the help of LDA(Latent dirichlet algorithm)

Input:

texts= ["""It's all about travel. I travel a lot. those who do not travel read only a page." - said Saint Augustine. He was a great travel person. Travelling can teach you more than any university course. You learn about the culture of the country you visit. If you talk to locals, you will likely learn about their thinking, habits, traditions and history as well. If you travel, you will not only learn about foreign cultures, but about your own as well. You will notice the cultural differences, and will find out what makes your culture unique. After retrurning from a long journey, you will see your country with new eyes.""",

""" You can learn a lot about yourself through travelling. You can observe how you feel beeing far from your country. You will find out how you feel about your homeland. You should travel You will realise how you really feel about foreign people. You will find out how much you know/do not know about the world. You will be able to observe how you react in completely new situations. You will test your language, orientational and social skills. You will not be the same person after returning home. During travelling you will meet people that are very different from you. If you travel enough, you will learn to accept and appreciate these differences. Traveling makes you more open and accepting.""",

"""Some of my most cherished memories are from the times when I was travelling. If you travel, you can experience things that you could never experience at home. You may see beautiful places and landscapes that do not exist where you live. You may meet people that will change your life, and your thingking. You may try activities that you have never tried before. Travelling will inevitably make you more independent and confident. You will realise that you can cope with a lot of unexpected situations. You will realise that you can survive without all that help that is always available for you at home. You will likely find out that you are much stronger and braver than you have expected.""",

"""If you travel, you may learn a lot of useful things. These things can be anything from a new recepie, to a new, more effective solution to an ordinary problem or a new way of creating something. Even if you go to a country where they speak the same language as you, you may still learn some new words and expressions that are only used there. If you go to a country where they speak a different language, you will learn even more."",

"""After arriving home from a long journey, a lot of travellers experience that they are much more motivated than they were before they left. During your trip you may learn things that you will want to try at home as well. You may want to test your new skills and knowledge. Your experiences will give you a lot of energy. During travelling you may experience the craziest, most exciting things, that will eventually become great stories that you can tell others. When you grow old and look back at your life and all your travel experiences, you will realise how much you have done in your life and your life was not in vain. It can provide you with happiness and satisfaction for the rest of your life.""",

"""The benefits of travel are not just a one-time thing: travel changes you physically and psychologically. Having little time or money isn't a valid excuse. You can travel for cheap very easily. If you have a full-time job and a family, you can still travel on the weekends or holidays, even with a baby. travel more is likely to have a tremendous impact on your mental well-being, especially if you're no used to going out of your comfort zone. Trust me: travel more and your doctor will be happy. Be sure to get in touch with your physician, they might recommend some medication to accompany you in your travels, especially if you're heading to regions of the globe with potentially dangerous diseases.""",

"""Sure, you probably feel comfortable where you are, but that is just a fraction of the world! If you are a student, take advantage of programs such as Erasmus to get to know more people, experience and understand their culture. Dare traveling to regions you have a skeptical opinion about. I bet that you will change your mind and realize that everything is not so bad abroad.""",

""" So, travel makes you cherish life. Let's travel more . Share your travel diaries with us too"""

```
[(0, '0.068*"travel" + 0.044*"learn" + 0.027*"country" + 0.027*"If" + 0.026*"find"'), (1, '0.054*"may" + 0.036*"realise" + 0.036*"home" + 0.036*"experience" + 0.036*"never"'), (2, '0.047*"may" + 0.044*"life" + 0.039*"new" + 0.036*"things" + 0.032*"learn"'), (3, '0.031*"If" + 0.031*"people" + 0.031*"I" +
```

```
101 NLP Exercises (using modern libraries) - Machine Learning Plus

0.031*"world" + 0.031*"know"'), (4, '0.085*"travel" +

0.042*"\'" + 0.042*"-" + 0.042*"time" + 0.028*"especially"')]
```

```
# Import gensim, nltk
import gensim
from gensim import models, corpora
import nltk
from nltk.corpus import stopwords
# Before topic extraction, we remove punctuations and
stopwords.
my stopwords=set(stopwords.words('english'))
punctuations=['.','!',',"You","I"]
# We prepare a list containing lists of tokens of each text
all tokens=[]
for text in texts:
  tokens=[]
  raw=nltk.wordpunct_tokenize(text)
  for token in raw:
    if token not in my_stopwords:
      if token not in punctuations:
        tokens.append(token)
        all_tokens.append(tokens)
# Creating a gensim dictionary and the matrix
dictionary = corpora.Dictionary(all_tokens)
doc_term_matrix = [dictionary.doc2bow(doc) for doc in
all_tokens]
# Building the model and training it with the matrix
from gensim.models.ldamodel import LdaModel
model = LdaModel(doc term matrix, num topics=5, id2word =
dictionary, passes=40)
print(model.print_topics(num_topics=6,num_words=5))
```

To understand more about how LDA works , check out our https://www.mach modeling-gensim-python/ .

27. How to extract topic keywords using NMF?

Difficulty Level: L3

Q. Extract the the topics from the given texts with the help of NMF(Non-negative Matrix Factorization method)

Input:

```
texts= ["""It's all about travel. I travel a lot. those who do not travel read only a page." - said Saint Augustine. He was a great travel person. Travelling can teach you more than any university course. You learn about the culture of the country you visit. If you talk to locals, you will likely learn about their thinking, habits, traditions and history as well. If you travel, you will not only learn about foreign cultures, but about your own as well. You will notice the cultural differences, and will find out what makes your culture unique. After retrurning from a long journey, you will see your country with new eyes."",

""" You can learn a lot about yourself through travelling. You can observe how you feel beeing far from your country. You will find out how you feel about your homeland. You
```

should travel You will realise how you really feel about foreign people. You will find out how much you know/do not know about the world. You will be able to observe how you react in completely new situations. You will test your language, orientational and social skills. You will not be the same person after returning home. During travelling you will meet people that are very different from you. If you travel enough, you will learn to accept and appreciate these differences. Traveling makes you more open and accepting.""",

"""Some of my most cherished memories are from the times when I was travelling. If you travel, you can experience things that you could never experience at home. You may see beautiful places and landscapes that do not exist where you live. You may meet people that will change your life, and your thingking. You may try activities that you have never tried before. Travelling will inevitably make you more independent and confident. You will realise that you can cope with a lot of unexpected situations. You will realise that you can survive without all that help that is always available for you at home. You will likely find out that you are much stronger and braver than you have expected.""",

"""If you travel, you may learn a lot of useful things. These things can be anything from a new recepie, to a new, more effective solution to an ordinary problem or a new way of creating something. Even if you go to a country where they speak the same language as you, you may still learn some new words and expressions that are only used there. If you go to a country where they speak a different language, you will learn even more."",

"""After arriving home from a long journey, a lot of travellers experience that they are much more motivated than they were before they left. During your trip you may learn things that you will want to try at home as well. You may want to test your new skills and knowledge. Your experiences will give you a lot of energy. During travelling you may experience the craziest, most exciting things, that will eventually become great stories that you can tell others. When you grow old and look back at your life and all your travel experiences, you will realise how much you have done in your life and your life was not in vain. It can provide you with happiness and satisfaction for the rest of your life.""",

"""The benefits of travel are not just a one-time thing: travel changes you physically and psychologically. Having little time or money isn't a valid excuse. You can travel for cheap very easily. If you have a full-time job and a family, you can still travel on the weekends or holidays, even with a baby. travel more is likely to have a tremendous impact on your mental well-being, especially if you're no used to going out of your comfort zone. Trust me: travel more and your doctor will be happy. Be sure to get in touch with your physician, they might recommend some medication to accompany you in your travels, especially if you're heading to regions of the globe with potentially dangerous diseases.""",

"""Sure, you probably feel comfortable where you are, but that is just a fraction of the world! If you are a student, take advantage of programs such as Erasmus to get to know more people, experience and understand their culture. Dare traveling to regions you have a skeptical opinion about. I bet that you will change your mind and realize that everything is not so bad abroad.""",

""" So, travel makes you cherish life. Let's travel more . Share your travel diaries with us too"""

```
Topic 0:
 [('new', 0.6329770846997606), ('learn', 0.49810389825931783),
('speak', 0.47477546214544547), ('language',
0.43443029670471806), ('country', 0.36653909845383115),
('things', 0.3433223730439043)]
 Topic 1:
 [('life', 0.34063551920788737), ('home', 0.31402014643240667),
('experience', 0.3025841622571281), ('realise',
0.24642870225288288), ('travelling', 0.2180915553025073),
('things', 0.2076347895889257)]
 Topic 2:
[('feel', 0.3462484013922396), ('know', 0.28400088182008115),
('people', 0.2431266883545085), ('world', 0.22169277349692076),
('traveling', 0.22169277349692076), ('bet',
0.18671974365540467)]
...(truncated)
```

```
from sklearn.feature extraction.text import TfidfVectorizer
# Defining the vectorizer
vectorizer = TfidfVectorizer(stop words='english',
max features= 1000, max df = 0.5, smooth idf=True)
# Transforming the tokens into the matrix form through
.fit transform()
nmf matrix= vectorizer.fit transform(texts)
from sklearn.decomposition import NMF
nmf model = NMF(n components=6)
nmf model.fit(nmf matrix)
# Function to print topics
def print topics nmf(model, vectorizer, top n=6):
    for idx, topic in enumerate(model.components):
        print("Topic %d:" % (idx))
       print([(vectorizer.get feature names()[i], topic[i])
                        for i in topic.argsort()[:-top n -
1:-1])
print topics nmf(nmf model, vectorizer)
```

```
#> Topic 0:
#> [('new', 0.6329770846997606), ('learn',
0.49810389825931783), ('speak', 0.47477546214544547),
('language', 0.43443029670471806), ('country',
0.36653909845383115), ('things', 0.3433223730439043)]
#> Topic 1:
#> [('life', 0.34063551920788737), ('home',
0.31402014643240667), ('experience', 0.3025841622571281),
('realise', 0.24642870225288288), ('travelling',
0.2180915553025073), ('things', 0.2076347895889257)]
#> Topic 2:
#> [('feel', 0.3462484013922396), ('know',
0.28400088182008115), ('people', 0.2431266883545085), ('world',
0.22169277349692076), ('traveling', 0.22169277349692076),
('bet', 0.18671974365540467)]
#> Topic 3:
#> [('time', 0.44163173193053806), ('especially',
0.2944211546203588), ('zone', 0.1472105773101794),
('dangerous', 0.1472105773101794), ('excuse',
0.1472105773101794), ('benefits', 0.1472105773101794)]
#> Topic 4:
#> [('cherish', 0.4703713910017504), ('diaries',
0.4703713910017504), ('share', 0.4703713910017504), ('let',
0.4703713910017504), ('life', 0.3536009233728117), ('makes',
0.3410286245344885)]
#> Topic 5:
#> [('learn', 0.2790596001102511), ('culture',
0.2285890660745815), ('country', 0.2086315311687136),
('locals', 0.140827224681607), ('eyes', 0.140827224681607),
('retrurning', 0.140827224681607)]
```

28. How to classify a text as positive/negative sentiment

Difficulty Level: L2

Q. Detect if a text is positive or negative sentiment

Input:

```
text="It was a very pleasant day"
```

Note that the magnitude of polarity represents the extent/intensity. If it the polarity is greater than 0, it represents positive sentiment and viceversa.

29. How to use the Word2Vec model for representing words?

Difficulty Level: L2

Q. Extract the word vector representation of the word using word2vec model

Input:

Desired Output:

```
array([ 2.94046826e-03, -1.31368915e-05, -3.43682081e-03,
-3.73885059e-03,
        2.49790819e-03, -1.23431312e-03, -9.60227044e-04,
2.31345627e-03,
       -4.97973803e-03, 2.09524506e-03, 2.00997619e-03,
-4.10459843e-03,
        8.42132606e-04, -2.70003616e-03, 3.12150107e-03,
1.23607670e-03,
        2.16376456e-03, 5.02903073e-04, -3.72780557e-03,
4.35266597e-03,
       -1.80016900e-03, 3.42973252e-03, -2.12087762e-03,
1.14531387e-04,
        3.03449039e-03, -8.75897415e-04, -3.50620854e-03,
5.10322629e-03,
       ...(truncated)
Positive
```

```
# We prepare a list containing lists of tokens of each text
tokens=[]
for text in texts:
```

```
101 NLP Exercises (using modern libraries) - Machine Learning Plus
    tokens=[]
    raw=nltk.wordpunct</em>tokenize(text)
    for token in raw:
        tokens.append(token)
        all_tokens.append(tokens)

# Import and fit the model with data
import gensim
from gensim.models import Word2Vec
model=Word2Vec(all_tokens)

# Getting the vector representation of a word
model['Photography']
```

```
array([ 2.94046826e-03, -1.31368915e-05, -3.43682081e-03,
-3.73885059e-03,
        2.49790819e-03, -1.23431312e-03, -9.60227044e-04,
2.31345627e-03,
       -4.97973803e-03, 2.09524506e-03, 2.00997619e-03,
-4.10459843e-03,
        8.42132606e-04, -2.70003616e-03, 3.12150107e-03,
1.23607670e-03,
        2.16376456e-03, 5.02903073e-04, -3.72780557e-03,
4.35266597e-03,
       -1.80016900e-03, 3.42973252e-03, -2.12087762e-03,
1.14531387e-04,
        3.03449039e-03, -8.75897415e-04, -3.50620854e-03,
5.10322629e-03,
        2.36228597e-03, 3.20315338e-03, -1.77754264e-03,
3.44046485e-03,
       -4.72177169e-04, 3.79201653e-03, 3.50930146e-03,
9.24638123e-04,
       -3.63159878e-03, 4.49452689e-03, -1.94674812e-03,
2.66790786e-03,
        3.57741816e-03, 4.08058614e-03, -4.22306563e-04,
3.21556488e-03,
        1.93726353e-03, -4.70201066e-03, -6.77402073e-04,
3.57477437e-03,
        2.40847061e-04, -3.06745851e-03, -3.21992044e-03,
-2.77571869e-03
        1.84161821e-03, -2.28599668e-03, 1.12327258e-03,
4.90778498e-03,
       -3.74632655e-03, 4.14755428e-03, -1.51176169e-03,
-2.46686535e-03,
       -2.91575165e-03, 1.66514842e-03, -2.64900009e-04,
4.17628558e-03,
       -1.15438248e-03, 3.30674206e-03, 3.89241078e-03,
1.07316789e-03,
       -3.56393168e-03, 4.21310542e-03, -3.83528182e-03,
4.87843808e-03,
        3.38425953e-03, 5.87464485e-04, 1.10692088e-03,
1.82324962e-03,
        3.44771869e-03, 2.54350528e-03, -3.22796614e-03,
4.83927201e-04,
       -4.45320550e-03, 4.85936319e-03, -3.69266351e-03,
-1.26241916e-03,
        4.05845884e-03, 2.44187587e-03, 1.55774585e-03,
-1.97902485e-03,
       -2.21285340e-03, 1.51218695e-03, -1.10817770e-03,
-1.91929389e-03,
       3.81433661e-03, -9.82026220e-04, -8.55478633e-04,
1.73925143e-03,
       -9.87094129e-04, 1.61158561e-03, 1.61566911e-03,
-6.77109638e-04],
      dtype=float32)
```

30. How to visualize the word embedding obtained from word2Vec model?

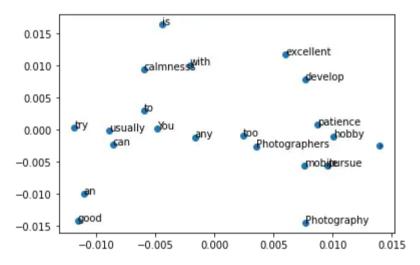
Difficulty Level: L4

Q. Implement Word embedding on the given texts and visualize it

" You can try Photography with any good mobile too"]

Show Solution >

```
# We prepare a list containing lists of tokens of each text
all tokens=[]
for text in texts:
  tokens=[]
  raw=nltk.wordpunct_tokenize(text)
  for token in raw:
    tokens.append(token)
    all_tokens.append(tokens)
# Import and fit the model with data
import gensim
from gensim.models import Word2Vec
model=Word2Vec(all_tokens)
# Visualizing the word embedding
from sklearn.decomposition import PCA
from matplotlib import pyplot
X = model[model.wv.vocab]
pca = PCA(n_components=2)
result = pca.fit transform(X)
# create a scatter plot of the projection
pyplot.scatter(result[:, 0], result[:, 1])
words = list(model.wv.vocab)
for i, word in enumerate(words):
        pyplot.annotate(word, xy=(result[i, 0], result[i, 1]))
pyplot.show()
```



Word2Vec Representation

31. How to represent the document using Doc2Vec model?

Difficulty Level: L2

Q. Represent a text document in the form a vector

Input:

```
array([ 2.6586275e-03, 3.2867077e-03, -2.0473711e-03, 6.0251489e-04,
```

```
# Importing the model
from gensim.models import Doc2Vec

# Preparing data in the format and fitting to the model
def tagged_document(list_of_list_of_words):
    for i, list_of_words in enumerate(list_of_list_of_words):
        yield gensim.models.doc2vec.TaggedDocument(list_of_words,
[i])
my_data = list(tagged_document(all_tokens))
model=Doc2Vec(my_data)

model.infer_vector(['photography','is','an',' excellent
','hobby ','to',' pursue '])
```

```
array([ 2.6586275e-03, 3.2867077e-03, -2.0473711e-03,
6.0251489e-04,
       -1.5340233e-03, 1.5060971e-03, 1.0988972e-03,
1.0712545e-03,
       -4.3745534e-03, -4.0448168e-03, -1.8953394e-04,
-2.0953947e-04,
       -3.3285557e-03, 1.0409033e-03, -8.5728493e-04,
4.5999791e-03,
       1.8428586e-03, 2.9749258e-03, 4.8927322e-04,
-4.1088923e-03,
       -1.2474873e-03, 4.5802444e-03, 2.4389643e-03,
-4.2193010e-04,
       -2.4726104e-03, 2.4501325e-03, 3.3282219e-03,
-3.0891516e-03,
        3.2441942e-03, -1.2857418e-03, -8.4910257e-04,
-1.0371304e-03,
       4.3518590e-03, 1.3085983e-03, 4.8915138e-03,
1.9108410e-03,
       -2.3149159e-03, -2.8708300e-03, 3.5418086e-03,
4.3390174e-03,
       2.7052627e-03, 4.1727605e-03, -3.7339646e-03,
4.4227624e-03,
        3.5092062e-03, 1.0140887e-03, -1.2085630e-03,
-1.5898966e-03,
       -1.0424303e-03, 2.5275371e-03, -4.4435970e-03,
2.9752296e-04,
        4.6713585e-03, 4.1678254e-03, -1.3408947e-03,
-4.1671298e-03,
       -5.3989125e-04, 2.3537579e-03, 4.9786703e-03,
-2.0938511e-03,
       -4.0806020e-03, -3.6052074e-03, 1.2903051e-03,
-4.2635379e-03
       -3.6390694e-03, -3.3433773e-03, 3.6569773e-03,
-1.8581208e-03,
       1.3781790e-04, -1.6561428e-03, -4.5162151e-03,
2.0534093e-04,
      -2.7264019e-03, -1.7743753e-03, -2.7915081e-03,
-1.1389129e-03,
        4.9526147e-03, 3.7630240e-03, -1.9377380e-03,
1.6532684e-04,
        4.9404724e-04, 3.4463860e-03, 2.6799906e-03,
1.6751935e-03,
       -6.6813978e-04, 3.6566416e-03, 2.5076446e-05,
1.9042364e-03,
       -1.0040828e-03, -8.4077887e-04, 3.3536348e-03,
-1.2608888e-03,
       -4.6293526e-03, 2.6570701e-03, -3.4919968e-03,
8.2246581e-04.
       6.5824442e-04, 1.2701214e-04, 3.8290059e-04,
-3.5389795e-03],
      dtype=float32)
```

To understand more about how to <code>gensim</code> library's features, check out our detailed post /https://www.machinelearningplus.com/nlp/gensim-tutorial/

32. How to extract the TF-IDF Matrix?

Difficulty Level: L3

Q. Extract the TF-IDF (Term Frequency -Inverse Document Frequency) Matrix for the given list of text documents

Input:

Desired Output:

```
(0, 13) 0.2511643891128359

(0, 12) 0.35300278529739293

(0, 8) 0.35300278529739293

(0, 5) 0.7060055705947859

(0, 6) 0.2511643891128359

(0, 7) 0.2511643891128359

...(truncated)..
```

```
# Method 1-Using gensim
from gensim import corpora
from gensim.utils import simple_preprocess
doc tokenized = [simple preprocess(text) for text in
text documents]
dictionary = corpora.Dictionary()
# Creating the Bag of Words from the docs
BoW_corpus = [dictionary.doc2bow(doc, allow_update=True) for
doc in doc tokenized]
for doc in BoW corpus:
  print([[dictionary[id], freq] for id, freq in doc])
import numpy as np
tfidf = models.TfidfModel(BoW_corpus)
#> [['for', 2], ['hobby', 1], ['is', 1], ['many', 1],
['painting', 1], ['passion', 1], ['some', 1]]
#> [['hobby', 1], ['is', 1], ['painting', 1], ['some', 1],
['and', 1], ['coin', 1], ['collectioni', 1], ['do', 1],
['every', 1], ['my', 1], ['now', 1], ['then', 1]]
```

33. How to create bigrams using Gensim's Phraser?

Difficulty Level: L3

Q. Create bigrams from the given texts using Gensim library's Phrases

Input:

```
documents = ["the mayor of new york was there", "new york mayor
was present"]
```

Desired Output:

```
['the', 'mayor', 'of', 'new york', 'was', 'there']
['new york', 'mayor', 'was', 'present']
```

Show Solution >

```
# Import Phraser from gensim
from gensim.models import Phrases
from gensim.models.phrases import Phraser

sentence_stream = [doc.split(" ") for doc in documents]

# Creating bigram phraser
bigram = Phrases(sentence_stream, min_count=1, threshold=2, delimiter=b' ')
bigram_phraser = Phraser(bigram)

for sent in sentence_stream:
    tokens_ = bigram_phraser[sent]
    print(tokens_)
```

34. How to create bigrams, trigrams using ngrams?

Difficulty Level: L3

Q. Extract all bigrams , trigrams using	ngrams	of	nltk	library
Input :				

Processing is how machines understand text "

world. Data science can do wonders . Natural Language

Sentences="Machine learning is a neccessary field in today's

Desired Output:

```
Bigrams are [('machine', 'learning'), ('learning', 'is'),
('is', 'a'), ('a', 'neccessary'), ('neccessary', 'field'),
('field', 'in'), ('in', "today's"), ("today's", 'world.'),
('world.', 'data'), ('data', 'science'), ('science', 'can'),
('can', 'do'), ('do', 'wonders'), ('wonders', '.'), ('.',
'natural'), ('natural', 'language'), ('language',
'processing'), ('processing', 'is'), ('is', 'how'), ('how',
'machines'), ('machines', 'understand'), ('understand',
'text')]
 Trigrams are [('machine', 'learning', 'is'), ('learning',
'is', 'a'), ('is', 'a', 'neccessary'), ('a', 'neccessary',
'field'), ('neccessary', 'field', 'in'), ('field', 'in',
"today's"), ('in', "today's", 'world.'), ("today's", 'world.',
'data'), ('world.', 'data', 'science'), ('data', 'science',
'can'), ('science', 'can', 'do'), ('can', 'do', 'wonders'),
('do', 'wonders', '.'), ('wonders', '.', 'natural'), ('.',
'natural', 'language'), ('natural', 'language', 'processing'),
('language', 'processing', 'is'), ('processing', 'is', 'how'),
('is', 'how', 'machines'), ('how', 'machines', 'understand'),
('machines', 'understand', 'text')]
```

Show Solution >

```
# Creating bigrams and trigrams
from nltk import ngrams
bigram=list(ngrams(Sentences.lower().split(),2))
trigram=list(ngrams(Sentences.lower().split(),3))

print(" Bigrams are",bigram)
print(" Trigrams are", trigram)
```

35. How to detect the language of entered text?

Difficulty Level : L1

Q. Find out the language of the given text

Input:

text="El agente imprime su pase de abordaje. Los oficiales de seguridad del aeropuerto pasan junto a él con un perro grande. El perro está olfateando alrededor del equipaje de las personas tratando de detectar drogas o explosivos."

Desired Output:

```
{'language': 'es', 'score': 0.9999963653206719}
El agente imprime su pase de abordaje. {'language': 'es',
'score': 0.9999969081229643}
```

Show Solution >

```
# Install spacy's languagedetect library
import spacy
!pip install spacy_langdetect
from spacy langdetect import LanguageDetector
nlp = spacy.load('en')
# Add the language detector to the processing pipeline
nlp.add pipe(LanguageDetector(), name='language detector',
last=True)
doc = nlp(text)
# document level language detection. Think of it like average
language of the document!
print(doc. .language)
# sentence level language detection
for sent in doc.sents:
   print(sent, sent._.language)
#> {'language': 'es', 'score': 0.9999963653206719}
#> El agente imprime su pase de abordaje. {'language': 'es',
'score': 0.9999969081229643}
#> Los oficiales de seguridad del aeropuerto pasan junto a él
con un perro grande. {'language': 'es', 'score':
0.9999951631258189}
#> El perro está olfateando alrededor del equipaje de las
personas tratando de detectar drogas o explosivos. {'language':
'es', 'score': 0.9999938903880353}
```

36. How to merge two tokens as one?

Difficulty Level: L3

Q. Merge the first name and last name as single token in the given sentence

Input:

```
text="Robert Langdon is a famous character in various books and
movies "
```

```
Robert Langdon
is
a
famous
character
in
various
books
and
movies
```

```
# Using retokenize() method of Doc object to merge two tokens
doc = nlp(text)
with doc.retokenize() as retokenizer:
   retokenizer.merge(doc[0:14])
for token in doc:
  print(token.text)
#> Robert Langdon
#> is
#> a
#> famous
#> character
#> in
#> various
#> books
#> and
#> movies
```

37. How to extract Noun phrases from a text?

Difficulty Level: L2

Q. Extract and print the noun phrases in given text document

Input:

```
text="There is a empty house on the Elm Street"
```

Expected Output:

```
[a empty house, the Elm Street]
```

```
# Create a spacy doc of the text
doc = nlp(text)

# Use `noun_chunks` attribute to extract the Noun phrases
chunks = list(doc.noun_chunks)
chunks

#> [a empty house, the Elm Street]
```

38. How to extract Verb phrases from the text

?

Difficulty Level: L3

Q. Extract the Verb Phrases from the given text

Input:

```
text=("I may bake a cake for my birthday. The talk will
introduce reader about Use of baking")
```

Desired Output:

```
may bake
will introduce
```

Show Solution >

```
# Import textacy library
!pip install textacy
import textacy

# Regex pattern to identify verb phrase
pattern = r'(<VERB>?<ADV>*<VERB>+)'
doc = textacy.make_spacy_doc(text,lang='en_core_web_sm')

# Finding matches
verb_phrases = textacy.extract.pos_regex_matches(doc, pattern)

# Print all Verb Phrase
for chunk in verb_phrases:
    print(chunk.text)

#> may bake
#> will introduce
```

39. How to extract first name and last names present in the document?

Difficulty Level: L3

Q. Extract any two consecutive Proper Nouns that occour in the text document

Input:

```
text="Sherlock Holmes and Clint Thomas were good friends. I am
a fan of John Mark"
```

```
Sherlock Holmes
Clint Thomas
```

John Mark

Show Solution >

```
# Import and initialize spacy's matcher
from spacy.matcher import Matcher
matcher = Matcher(nlp.vocab)
doc=nlp(text)
# Function that adds patterns to the matcher and finds the
respective matches
def extract matches(doc):
  pattern = [{'POS': 'PROPN'}, {'POS': 'PROPN'}]
  matcher.add('FULL_NAME', None, pattern)
  matches = matcher(doc)
   for match id, start, end in matches:
     span = doc[start:end]
    print(span.text)
extract_matches(doc)
#> Sherlock Holmes
#> Clint Thomas
#> John Mark
```

40. How to identify named entities in the given text

Difficulty Level: L2

Q. Identify and print all the named entities with their labels in the below text

Input

```
text=" Walter works at Google. He lives in London."
```

Desired Output:

```
Walter PERSON
Google ORG
London GPE
```

```
# Load spacy modelimport spacy
nlp=spacy.load("en_core_web_sm")doc=nlp(text)
# Using the ents attribute of doc, identify labels
for entity in doc.ents:
    print(entity.text,entity.label_)

#> Walter PERSON
#> Google ORG
#> London GPE
```

41. How to identify all the names of Organizations present in the text with NER?

Difficulty Level: L2

Q. Identify and extract a list of all organizations/Companies mentioned in the given news article

Input:

```
text =" Google has released it's new model which has got
attention of everyone. Amazon is planning to expand into Food
delivery, thereby giving competition . Apple is coming up with
new iphone model. Flipkart will have to catch up soon."
```

Expected Solution

```
['Google', 'Amazon', 'Apple', 'Flipkart']
```

Show Solution >

```
doc=nlp(text)
list_of_org=[]
for entity in doc.ents:
   if entity.label_=="ORG":
       list_of_org.append(entity.text)

print(list_of_org)

#> ['Google', 'Amazon', 'Apple', 'Flipkart']
```

42. How to replace all names of people in the text with 'UNKNOWN'

Difficulty Level: L3

Q. Identify and replace all the person names in the news article with UNKNO\ Input:

```
news=" Walter was arrested yesterday at Brooklyn for murder. The suspicions and fingerprints pointed to Walter and his friend Pinkman . The arrest was made by inspector Hank"
```

Desired Output:

' UNKNOWN was arrested yesterday at Brooklyn for murder . The suspicions and fingerprints pointed to UNKNOWN $\,$ and his friend UNKNOWN . The arrest was made by inspector UNKNOWN'

Show Solution >

```
doc=nlp(news)

# Identifying the entities of category 'PERSON'
entities = [entity.text for entity in doc.ents if
entity.label_=='PERSON']
updated_text=[]

for token in doc:
    if token.text in entities:
        updated_text.append("UNKNOWN")
    else:
        updated_text.append(token.text)

" ".join(updated_text)

#> ' UNKNOWN was arrested yesterday at Brooklyn for murder .
The suspicions and fingerprints pointed to UNKNOWN and his
friend UNKNOWN . The arrest was made by inspector UNKNOWN'
```

43. How to visualize the named entities using spaCy

Difficulty Level: L2

Q. Display the named entities prsent in the given document along with their categories using spacy

Input:

```
text=" Walter was arrested yesterday at Brooklyn for murder.
The suspicions and fingerprints pointed to Walter and his
friend Pinkman . He is from Paris "
```

Show Solution >

```
# Use spacy's displacy with the parameter style="ent"

from spacy import displacy
doc=nlp(text)
displacy.render(doc, style='ent', jupyter=True)
```

```
Walter Person was arrested yesterday DATE at Brooklyn GPE for murder. The suspicions and fingerprints pointed to Walter Person and his friend Pinkman

Person . He is from Parls GPE
```

44. How to implement dependency parsing?

Difficulty Level: L2

Q. Find the dependencies of all the words in the given text

Input:

```
text="Mark plays volleyball every evening."
```

Desired Output:

```
Mark nsubj
plays ROOT
volleyball dobj
every det
evening npadvmod
. punct
```

Show Solution >

```
# Using dep_ attribute od tokens in spaCy to access the
dependency of the word in sentence.
doc=nlp(text)

for token in doc:
    print(token.text,token.dep_)

#> Mark nsubj
#> plays ROOT
#> volleyball dobj
#> every det
#> evening npadvmod
#> . punct
```

45. How to find the ROOT word of any word in a sentence?

Difficulty Level: L3

Q. Find and print the root word / headword of any word in the given sentence

Input:

```
text="Mark plays volleyball. Sam is not into sports, he paints
a lot"
```

Desired Output:

```
Mark plays
plays plays
volleyball plays
. plays
Sam is
is paints
not is
into is
sports into
, paints
he paints
paints paints
a lot
lot paints
```

```
# use the head attribute of tokens to find it's rootword
doc=nlp(text)
for token in doc:
  print(token.text, token.head)
#> Mark plays
#> plays plays
#> volleyball plays
#> . plays
#> Sam is
#> is paints
#> not is
#> into is
#> sports into
#> , paints
#> he paints
#> paints paints
#> a lot
#> lot paints
```

46. How to visualize the dependency tree in spaCy

Difficulty Level: L2

Q. Visualize the dependencies of various tokens of the given text using spaCy

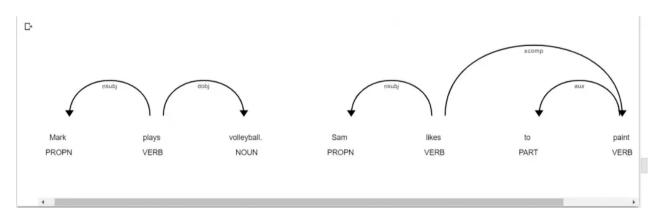
Input:

```
text="Mark plays volleyball. Sam is not into sports, he paints
a lot"
```

Show Solution >

```
# Use spacy's displacy with the parameter style="dep"
doc=nlp(text)

from spacy import displacy
displacy.render(doc, style='dep', jupyter=True)
```



47. How to detect all the Laptop names present in the text?

Difficulty Level: L4

Q. Detect all the Laptop names present in the given document .

Input:

```
text="For my offical use, I prefer lenova. For gaming purposes,
I love asus"
```

Expected Output

```
lenova laptop
asus laptop
```

```
# Import EntityRuler of spacy model
import spacy
nlp=spacy.load("en core_web_sm")
from spacy.pipeline import EntityRuler
# Functions to create patterns of laptop name to match
def create versioned(name):
    return [
        [{'LOWER': name}],
        [{'LOWER': {'REGEX': f'({name}\d+\.?\d*.?\d*)'}}],
        [{'LOWER': name}, {'TEXT': {'REGEX': '(\d+\.?\d*.?
\d*) '}}]]
def create_patterns():
    versioned_languages = ['dell', 'HP',
'asus','msi','Apple','HCL','sony','samsung','lenova','acer']
    flatten = lambda l: [item for sublist in l for item in
sublist
    versioned patterns = flatten([create versioned(lang) for
lang in versioned languages])
    lang patterns = [
        [{'LOWER': 'dell'}, {'LIKE_NUM': True}],
        [{'LOWER': 'HP'}],
        [{'LOWER': 'asus'}, {'LOWER': '#'}],
        [{'LOWER': 'msi'}, {'LOWER': 'sharp'}],
        [{'LOWER': 'Apple'}],
        [{'LOWER': 'HCL'}, {'LOWER': '#'}],
        [{'LOWER': 'sony'}],
        [{'LOWER': 'samsung'}],
        [{'LOWER': 'toshiba'}],
        [{'LOWER': 'dell'}, {'LOWER': 'inspiron'}],
        [{'LOWER': 'acer'}, {'IS_PUNCT': True, 'OP': '?'},
{'LOWER': 'c'}],
       [{'LOWER': 'golang'}],
        [{'LOWER': 'lenova'}],
        [{'LOWER': 'HP'}, {'LOWER': 'gaming'}],
        [{'LOWER': 'Fujitsu'}],
        [{'LOWER': 'micromax'}],
    return versioned_patterns + lang_patterns
# Add the Entity Ruler to the pipeline
ruler=EntityRuler(nlp)
ruler.add_patterns([{'label':'laptop','pattern':p} for p in
create_patterns()])
nlp.add_pipe(ruler)
# Identify the car names now
doc=nlp("For my offical use, I prefer lenova. For gaming
purposes, I love asus")
for ent in doc.ents:
 print(ent.text,ent.label)
#> lenova laptop
#> asus laptop
```

48. How to summarize text using gensim?

Difficulty Level: L3

Q. Extract the summary of the given text based using <code>gensim</code> package based on the TextRank Algorithm.

Input:

original_text="""Studies show that exercise can treat mild to moderate depression as effectively as antidepressant medication—but without the side-effects, of course. As one example, a recent study done by the Harvard T.H. Chan School of Public Health found that running for 15 minutes a day or walking for an hour reduces the risk of major depression by 26%. In addition to relieving depression symptoms, research also shows that maintaining an exercise schedule can prevent you from relapsing.

Exercise is a powerful depression fighter for several reasons. Most importantly, it promotes all kinds of changes in the brain, including neural growth, reduced inflammation, and new activity patterns that promote feelings of calm and well-being. It also releases endorphins, powerful chemicals in your brain that energize your spirits and make you feel good. Finally, exercise can also serve as a distraction, allowing you to find some quiet time to break out of the cycle of negative thoughts that feed depression.

Exercise is not just about aerobic capacity and muscle size. Sure, exercise can improve your physical health and your physique, trim your waistline, improve your sex life, and even add years to your life. But that's not what motivates most people to stay active.

People who exercise regularly tend to do so because it gives them an enormous sense of well-being. They feel more energetic throughout the day, sleep better at night, have sharper memories, and feel more relaxed and positive about themselves and their lives. And it's also powerful medicine for many common mental health challenges.

Regular exercise can have a profoundly positive impact on depression, anxiety, ADHD, and more. It also relieves stress, improves memory, helps you sleep better, and boosts your overall mood. And you don't have to be a fitness fanatic to reap the benefits. Research indicates that modest amounts of exercise can make a difference. No matter your age or fitness level, you can learn to use exercise as a powerful tool to feel better.

Ever noticed how your body feels when you're under stress? Your muscles may be tense, especially in your face, neck, and shoulders, leaving you with back or neck pain, or painful headaches. You may feel a tightness in your chest, a pounding pulse, or muscle cramps. You may also experience problems such as insomnia, heartburn, stomachache, diarrhea, or frequent urination. The worry and discomfort of all these physical symptoms can in turn lead to even more stress, creating a vicious cycle between your mind and body.

Exercising is an effective way to break this cycle. As well as releasing endorphins in the brain, physical activity helps to relax the muscles and relieve tension in the body. Since the body and mind are so closely linked, when your body feels better so, too, will your mind. Evidence suggests that by really focusing on your body and how it feels as you exercise, you can actually help your nervous system become "unstuck" and begin to move out of the immobilization stress response that characterizes PTSD or trauma.

Instead of allowing your mind to wander, pay close attention to the physical sensations in your joints and muscles, even your insides as your body moves. Exercises that involve cross movement and that engage both arms and legs—such as walking (especially in sand), running, swimming, weight training, or dancing—are some of your best choices.

Outdoor activities like hiking, sailing, mountain biking, rock climbing, whitewater rafting, and skiing (downhill and cross-country) have also been shown to reduce the symptoms of PTSD."""

Desired Output:

As one example, a recent study done by the Harvard T.H. Chan School of Public Health found that running for 15 minutes a day

or walking for an hour reduces the risk of major depression by 26%.

No matter your age or fitness level, you can learn to use exercise as a powerful tool to feel better.

The worry and discomfort of all these physical symptoms can in turn lead to even more stress, creating a vicious cycle between your mind and body.

As well as releasing endorphins in the brain, physical activity helps to relax the muscles and relieve tension in the body.

Show Solution >

Importing the summarize function from gensim module
import gensim

from gensim.summarization.summarizer import summarize

Pass the document along with desired word count to get the summary

my_summary=summarize(original_text,word_count=100)
print(my_summary)

#> As one example, a recent study done by the Harvard T.H. Chan School of Public Health found that running for 15 minutes a day or walking for an hour reduces the risk of major depression by 26%.

#> No matter your age or fitness level, you can learn to use exercise as a powerful tool to feel better.

#> The worry and discomfort of all these physical symptoms can in turn lead to even more stress, creating a vicious cycle between your mind and body.

#> As well as releasing endorphins in the brain, physical activity helps to relax the muscles and relieve tension in the body.

49. How to summarize text based on the LexRank algorithm?

Difficulty Level: L3

Q. Extract the summary of the given text based on the TextRank Algorithm.

Input:

original_text="""Studies show that exercise can treat mild to moderate depression as effectively as antidepressant medication—but without the side-effects, of course. As one example, a recent study done by the Harvard T.H. Chan School of Public Health found that running for 15 minutes a day or walking for an hour reduces the risk of major depression by 26%. In addition to relieving depression symptoms, research also shows that maintaining an exercise schedule can prevent you from relapsing.

Exercise is a powerful depression fighter for several reasons. Most importantly, it promotes all kinds of changes in the brain, including neural growth, reduced inflammation, and new activity patterns that promote feelings of calm and well-being. It also releases endorphins, powerful chemicals in your brain that energize your spirits and make you feel good. Finally, exercise can also serve as a distraction, allowing you to find some quiet time to break out of the cycle of negative thoughts that feed depression.

Exercise is not just about aerobic capacity and muscle size. Sure, exercise can improve your physical health and your physique, trim your waistline, improve your sex life, and even add years to your life. But that's not what motivates most people to stay active.

People who exercise regularly tend to do so because it gives them an enormous sense of well-being. They feel more energetic throughout the day, sleep better at night, have sharper memories, and feel more relaxed and positive about themselves and their lives. And it's also powerful medicine for many common mental health challenges.

Regular exercise can have a profoundly positive impact on depression, anxiety, ADHD, and more. It also relieves stress, improves memory, helps you sleep better, and boosts your overall mood. And you don't have to be a fitness fanatic to reap the benefits. Research indicates that modest amounts of exercise can make a difference. No matter your age or fitness level, you can learn to use exercise as a powerful tool to feel better.

Ever noticed how your body feels when you're under stress? Your muscles may be tense, especially in your face, neck, and shoulders, leaving you with back or neck pain, or painful headaches. You may feel a tightness in your chest, a pounding pulse, or muscle cramps. You may also experience problems such as insomnia, heartburn, stomachache, diarrhea, or frequent urination. The worry and discomfort of all these physical symptoms can in turn lead to even more stress, creating a vicious cycle between your mind and body.

Exercising is an effective way to break this cycle. As well as releasing endorphins in the brain, physical activity helps to relax the muscles and relieve tension in the body. Since the body and mind are so closely linked, when your body feels better so, too, will your mind. Evidence suggests that by really focusing on your body and how it feels as you exercise, you can actually help your nervous system become "unstuck" and begin to move out of the immobilization stress response that characterizes PTSD or trauma.

Instead of allowing your mind to wander, pay close attention to the physical sensations in your joints and muscles, even your insides as your body moves. Exercises that involve cross movement and that engage both arms and legs—such as walking (especially in sand), running, swimming, weight training, or dancing—are some of your best choices.

Outdoor activities like hiking, sailing, mountain biking, rock climbing, whitewater rafting, and skiing (downhill and cross-country) have also been shown to reduce the symptoms of PTSD."""

Desired Output:

Since the body and mind are so closely linked, when your body feels better so, too, will your mind. Evidence suggests that by really focusing on your body and how it feels as you exercise, you can actually help your nervous system become "unstuck" and begin to move out of the immobilization stress response that characterizes PTSD or trauma.>, <Sentence: Instead of allowing your mind to wander, pay close attention to the physical sensations in your joints and muscles, even your insides as your body moves.

Show Solution >

```
import sumy
from sumy.summarizers.lex_rank import LexRankSummarizer

#Plain text parsers since we are parsing through text
from sumy.parsers.plaintext import PlaintextParser
from sumy.nlp.tokenizers import Tokenizer

parser=PlaintextParser.from_string(original_text,Tokenizer("eng lish"))

summarizer=LexRankSummarizer()
my_summary=summarizer(parser.document,2)
print(my_summary)

#> (<Sentence: Since the body and mind are so closely linked, when your body feels better so, too, will your mind.Evidence suggests that by really focusing on your body and how it feels</pre>
```

when your body feels better so, too, will your mind. Evidence suggests that by really focusing on your body and how it feels as you exercise, you can actually help your nervous system become "unstuck" and begin to move out of the immobilization stress response that characterizes PTSD or trauma.>, <Sentence: Instead of allowing your mind to wander, pay close attention to the physical sensations in your joints and muscles, even your insides as your body moves.>)

50. How to summarize text using Luhn algorithm?

Q. Extract the summary of the given text based on the Luhn Algorithm.

Difficulty Level: L3

Input:

original_text="""Studies show that exercise can treat mild to moderate depression as effectively as antidepressant medication—but without the side-effects, of course. As one example, a recent study done by the Harvard T.H. Chan School of Public Health found that running for 15 minutes a day or walking for an hour reduces the risk of major depression by 26%. In addition to relieving depression symptoms, research also shows that maintaining an exercise schedule can prevent you from relapsing.

Exercise is a powerful depression fighter for several reasons. Most importantly, it promotes all kinds of changes in the brain, including neural growth, reduced inflammation, and new activity patterns that promote feelings of calm and well-being. It also releases endorphins, powerful chemicals in your brain that energize your spirits and make you feel good. Finally, exercise can also serve as a distraction, allowing you to find some quiet time to break out of the cycle of negative thoughts that feed depression.

Exercise is not just about aerobic capacity and muscle size. Sure, exercise can improve your physical health and your physique, trim your waistline, and even add years to your life. But that's not what motivates most people to stay active. People who exercise regularly tend to do so because it gives them an enormous sense of well-being. They feel more energetic throughout the day, sleep better at night, have sharper memories, and feel more relaxed and positive about themselves and their lives. And it's also powerful medicine for many common mental health challenges.

Regular exercise can have a profoundly positive impact on depression, anxiety, ADHD, and more. It also relieves stress, improves memory, helps you sleep better, and boosts your overall mood. And you don't have to be a fitness fanatic to reap the benefits. Research indicates that modest amounts of exercise can make a difference. No matter your age or fitness level, you can learn to use exercise as a powerful tool to feel better.

Ever noticed how your body feels when you're under stress? Your muscles may be tense, especially in your face, neck, and shoulders, leaving you with back or neck pain, or painful headaches. You may feel a tightness in your chest, a pounding pulse, or muscle cramps. You may also experience problems such as insomnia, heartburn, stomachache, diarrhea, or frequent urination. The worry and discomfort of all these physical symptoms can in turn lead to even more stress, creating a vicious cycle between your mind and body.

Exercising is an effective way to break this cycle. As well as releasing endorphins in the brain, physical activity helps to relax the muscles and relieve tension in the body. Since the body and mind are so closely linked, when your body feels better so, too, will your mind. Evidence suggests that by really focusing on your body and how it feels as you exercise, you can actually help your nervous system become "unstuck" and begin to move out of the immobilization stress response that characterizes PTSD or trauma.

Instead of allowing your mind to wander, pay close attention to the physical sensations in your joints and muscles, even your insides as your body moves. Exercises that involve cross movement and that engage both arms and legs—such as walking (especially in sand), running, swimming, weight training, or dancing—are some of your best choices.

Outdoor activities like hiking, sailing, mountain biking, rock climbing, whitewater rafting, and skiing (downhill and cross-country) have also been shown to reduce the symptoms of PTSD."""

Desired Output:

Finally, exercise can also serve as a distraction, allowing you to find some quiet time to break out of the cycle of negative thoughts that feed depression. Since the body and mind are so closely linked, when your body feels better so, too, will your mind. Evidence suggests that by really focusing on your body and how it feels as you exercise, you can actually help your nervous system become "unstuck" and begin to move out of the immobilization stress response that characterizes PTSD or trauma

Show Solution >

```
import sumy
from sumy.summarizers.luhn import LuhnSummarizer

#Plain text parsers since we are parsing through text
from sumy.parsers.plaintext import PlaintextParser
from sumy.nlp.tokenizers import Tokenizer

parser=PlaintextParser.from_string(original_text,Tokenizer("eng lish"))

summarizer=LuhnSummarizer()
my_summary=summarizer(parser.document,2)
print(my_summary)
```

#> (<Sentence: Finally, exercise can also serve as a
distraction, allowing you to find some quiet time to break out
of the cycle of negative thoughts that feed depression.>,
<Sentence: Since the body and mind are so closely linked, when
your body feels better so, too, will your mind.Evidence
suggests that by really focusing on your body and how it feels
as you exercise, you can actually help your nervous system
become "unstuck" and begin to move out of the immobilization
stress response that characterizes PTSD or trauma.>)

51. How to summarize text based on LSA algorithm?

Difficulty Level: L3

Q. Extract the summary of the given text based on the LSA Algorithm.

Input:

original_text="""Studies show that exercise can treat mild to moderate depression as effectively as antidepressant medication—but without the side-effects, of course. As one example, a recent study done by the Harvard T.H. Chan School of Public Health found that running for 15 minutes a day or walking for an hour reduces the risk of major depression by 26%. In addition to relieving depression symptoms, research also shows that maintaining an exercise schedule can prevent you from relapsing.

Exercise is a powerful depression fighter for several reasons. Most importantly, it promotes all kinds of changes in the brain, including neural growth, reduced inflammation, and new activity patterns that promote feelings of calm and well-being. It also releases endorphins, powerful chemicals in your brain that energize your spirits and make you feel good. Finally, exercise can also serve as a distraction, allowing you to find some quiet time to break out of the cycle of negative thoughts that feed depression.

Exercise is not just about aerobic capacity and muscle size. Sure, exercise can improve your physical health and your physique, trim your waistline, and even add years to your life. But that's not what motivates most people to stay active. People who exercise regularly tend to do so because it gives

them an enormous sense of well-being. They feel more energetic throughout the day, sleep better at night, have sharper memories, and feel more relaxed and positive about themselves and their lives. And it's also powerful medicine for many common mental health challenges.

Regular exercise can have a profoundly positive impact on depression, anxiety, ADHD, and more. It also relieves stress, improves memory, helps you sleep better, and boosts your overall mood. And you don't have to be a fitness fanatic to reap the benefits. Research indicates that modest amounts of exercise can make a difference. No matter your age or fitness level, you can learn to use exercise as a powerful tool to feel better.

Ever noticed how your body feels when you're under stress? Your muscles may be tense, especially in your face, neck, and shoulders, leaving you with back or neck pain, or painful headaches. You may feel a tightness in your chest, a pounding pulse, or muscle cramps. You may also experience problems such as insomnia, heartburn, stomachache, diarrhea, or frequent urination. The worry and discomfort of all these physical symptoms can in turn lead to even more stress, creating a vicious cycle between your mind and body.

Exercising is an effective way to break this cycle. As well as releasing endorphins in the brain, physical activity helps to relax the muscles and relieve tension in the body. Since the body and mind are so closely linked, when your body feels better so, too, will your mind. Evidence suggests that by really focusing on your body and how it feels as you exercise, you can actually help your nervous system become "unstuck" and begin to move out of the immobilization stress response that characterizes PTSD or trauma.

Instead of allowing your mind to wander, pay close attention to the physical sensations in your joints and muscles, even your insides as your body moves. Exercises that involve cross movement and that engage both arms and legs—such as walking (especially in sand), running, swimming, weight training, or dancing—are some of your best choices.

Outdoor activities like hiking, sailing, mountain biking, rock climbing, whitewater rafting, and skiing (downhill and cross-country) have also been shown to reduce the symptoms of PTSD."""

Desired Output:

In addition to relieving depression symptoms, research also shows that maintaining an exercise schedule can prevent you from relapsing. People who exercise regularly tend to do so because it gives them an enormous sense of well-being.

Show Solution >

```
import sumy
from sumy.summarizers.lsa import LsaSummarizer

#Plain text parsers since we are parsing through text
from sumy.parsers.plaintext import PlaintextParser
from sumy.nlp.tokenizers import Tokenizer

parser=PlaintextParser.from_string(original_text,Tokenizer("eng lish"))

summarizer=LsaSummarizer()
my_summary=summarizer(parser.document,2)
print(my_summary)
```

#> (<Sentence: In addition to relieving depression symptoms,
research also shows that maintaining an exercise schedule can
prevent you from relapsing.>, <Sentence: People who exercise
regularly tend to do so because it gives them an enormous sense
of well-being.>)

52. How to convert documents into json format?

Difficulty Level: L3

Q. Covert the given text documents into json format for spacy usage

Input:

```
text1="Netflix has released a new series"
text2="It was shot in London"
text3="It is called Dark and the main character is Jonas"
text4="Adam is the evil character"
```

Desired Output:

```
{'id': 0,
'paragraphs': [{'cats': [],
   'raw': 'Netflix has released a new series',
  'sentences': [{'brackets': [],
    'tokens': [{'dep': 'nsubj',
      'head': 2,
       'id': 0,
       'ner': 'U-ORG',
       'orth': 'Netflix',
       'tag': 'NNP'},
      { 'dep': 'aux',
       'head': 1,
       'id': 1,
       'ner': '0',
       'orth': 'has',
       'tag': 'VBZ'},
      { 'dep': 'ROOT',
       'head': 0,
       'id': 2,
       'ner': '0',
       'orth': 'released',
       'tag': 'VBN'},
      {'dep': 'det', 'head': 2, 'id': 3, 'ner': '0', 'orth':
'a', 'tag': 'DT'},
      { 'dep': 'amod',
       'head': 1,
       'id': 4,
       'ner': '0',
       'orth': 'new',
       'tag': 'JJ'},
      { 'dep': 'dobj',
       'head': -3,
       'id': 5,
       'ner': '0',
       'orth': 'series',
       'tag': 'NN'}]}},
    ... (truncated)
```

```
# Covert into spacy documents
doc1=nlp(text1)
doc2=nlp(text2)
doc3=nlp(text3)
doc4=nlp(text4)

# Import docs_to_json
from spacy.gold import docs_to_json

# Converting into json format
json_data = docs_to_json([doc1,doc2,doc3,doc4])
json_data
```

```
{'id': 0,
'paragraphs': [{'cats': [],
   'raw': 'Netflix has released a new series',
   'sentences': [{'brackets': [],
     'tokens': [{'dep': 'nsubj',
       'head': 2,
       'id': 0,
       'ner': 'U-ORG',
       'orth': 'Netflix',
       'tag': 'NNP'},
      { 'dep': 'aux',
       'head': 1,
       'id': 1,
       'ner': '0',
       'orth': 'has',
       'tag': 'VBZ'},
      { 'dep': 'ROOT',
       'head': 0,
       'id': 2,
       'ner': '0',
       'orth': 'released',
       'tag': 'VBN'},
      {'dep': 'det', 'head': 2, 'id': 3, 'ner': '0', 'orth':
'a', 'tag': 'DT'},
      { 'dep': 'amod',
       'head': 1,
       'id': 4,
       'ner': '0',
       'orth': 'new',
       'tag': 'JJ'},
      { 'dep': 'dobj',
       'head': -3,
       'id': 5,
       'ner': '0',
       'orth': 'series',
       'tag': 'NN'}]}},
  {'cats': [],
  'raw': 'It was shot in London',
   'sentences': [{'brackets': [],
     'tokens': [{'dep': 'nsubjpass',
       'head': 2,
       'id': 0,
       'ner': '0',
       'orth': 'It',
       'tag': 'PRP'},
      { 'dep': 'auxpass',
       'head': 1,
       'id': 1,
       'ner': '0',
       'orth': 'was',
       'tag': 'VBD'},
      { 'dep': 'ROOT',
       'head': 0,
       'id': 2,
       'ner': '0',
       'orth': 'shot',
       'tag': 'VBN'},
      {'dep': 'prep',
       'head': -1,
       'id': 3,
       'ner': '0',
       'orth': 'in',
       'tag': 'IN'},
      { 'dep': 'pobj',
       'head': -1,
       'id': 4,
       'ner': 'U-GPE',
        orth': 'London',
       'tag': 'NNP'}]}],
 ...(truncated)
```

53. How to build a text classifier with TextBlob?

Difficulty Level: L3

Q Build a text classifier with available train data using textblob library

Input:

```
# Data to train the classifier
train = [
    ('I love eating sushi', 'food-review'),
    ('This is an amazing place!', 'Tourist-review'),
    ('Pizza is my all time favorite food', 'food-review'),
    ('I baked a cake yesterday, it was tasty', 'food-review'),
    ("What an awesome taste this sushi has", 'food-review'),
    ('It is a perfect place for outing', 'Tourist-review'),
    ('This is a nice picnic spot', 'Tourist-review'),
    ("Families come out on tours here", 'Tourist-review'),
    ('It is a beautiful place !', 'Tourist-review'),
    ('The place was warm and nice', 'Tourist-review')
test = [
    ('The sushi was good', 'food-review'),
    ('The place was perfect for picnics ', 'Tourist-review'),
    ("Burgers are my favorite food", 'food-review'),
    ("I feel amazing!", 'food-review'),
    ('It is an amazing place', 'Tourist-review'),
    ("This isn't a very good place", 'Tourist-review')
```

Desired Output:

```
Accuracy: 0.833333333333333
```

Show Solution >

54. How to train a text classifier using Simple transformers?

Difficulty Level: L4

Q. Build and train a text classifier for the given data using simpletransformers library

Input:

```
train_data = [
    ["The movie was amazing", 1],
    ["It was a boring movie", 0],
    ["I had a great experience",1],
    ["I was bored during the movie",0],
    ["The movie was great",1],
    ["The movie was bad",0],
```

```
101 NLP Exercises (using modern libraries) - Machine Learning Plus

["The movie was good", 1]
]
```

Show Solution V

```
# Import requirements
!pip install simpletransformers
from simpletransformers.classification import
ClassificationModel, ClassificationArgs
import pandas as pd
import logging
logging.basicConfig(level=logging.INFO)
transformers logger = logging.getLogger("transformers")
transformers_logger.setLevel(logging.WARNING)
# Preparing train data
train df = pd.DataFrame(train data)
train df.columns = ["text", "labels"]
# Optional model configuration
model_args = ClassificationArgs(num_train_epochs=5)
# Create a ClassificationModel
model = ClassificationModel("bert", "bert-base-uncased",
args=model args,use cuda=False)
# Train the model
model.train model(train df)
# Make predictions with the model
predictions, raw outputs = model.predict(["The titanic was a
good movie"])
predictions
#> array([1])
```

55. How to perform text classification using spaCy?

Difficulty Level: L4

Q. Build a text classifier using spacy that can classify IMDB reviews as positive or negative

```
import spacy
nlp=spacy.load("en_core_web_sm")

textcat = nlp.create_pipe("textcat", config=
{"exclusive_classes": True, "architecture": "simple_cnn"})
nlp.add_pipe(textcat, last=True)
textcat = nlp.get_pipe("textcat")

# add label to text classifier
textcat.add_label("POSITIVE")
textcat.add_label("NEGATIVE")

def load_data(limit=0, split=0.8):
    """Load data from the IMDB dataset."""
```

```
# Partition off part of the train data for evaluation
    train data, = thinc.extra.datasets.imdb()
    random.shuffle(train data)
    train data = train data[-limit:]
    texts, labels = zip(*train data)
    cats = [{"POSITIVE": bool(y), "NEGATIVE": not bool(y)} for
y in labels]
    split = int(len(train data) * split)
    return (texts[:split], cats[:split]), (texts[split:],
cats[split:])
# load the IMDB dataset
print("Loading IMDB data...")
(train texts, train cats), (dev texts, dev cats) = load data()
train texts = train texts[:n texts]
train_cats = train_cats[:n_texts]
train data = list(zip(train texts, [{"cats": cats} for cats in
train cats]))
# get names of other pipes to disable them during training
pipe exceptions = ["textcat", "trf wordpiecer", "trf tok2vec"]
other pipes = [pipe for pipe in nlp.pipe names if pipe not in
pipe_exceptions]
# Training the text classifier
with nlp.disable_pipes(*other_pipes): # only train textcat
   optimizer = nlp.begin_training()
   if init tok2vec is not None:
      with init tok2vec.open("rb") as file :
        textcat.model.tok2vec.from bytes(file .read())
        print("Training the model...")
        print("{:^5}\t{:^5}\t{:^5}\t{:^5}\".format("LOSS", "P",
"R", "F"))
        batch sizes = compounding (4.0, 32.0, 1.001)
        for i in range(n_iter):
            losses = {}
            # batch up the examples using spaCy's minibatch
            random.shuffle(train data)
            batches = minibatch(train_data, size=batch_sizes)
            for batch in batches:
                texts, annotations = zip(*batch)
                nlp.update(texts, annotations, sgd=optimizer,
drop=0.2, losses=losses)
```

56. How to translate the text (using simpletransformers)?

Difficulty Level: L3

Q. Translate the given list of texts from English to Dutch using simpletransformers package

Input:

```
['Our experienced writers travel the world to bring you
informative and inspirational features, destination roundups,
travel ideas, tips and beautiful photos in order to help you
plan your next holiday',
                  'Each part of Germany is different, and there
are thousands of memorable places to visit.',
                 "Christmas Markets originated in Germany, and
the tradition dates to the Late Middle Ages.",
                  "Garmisch-Partenkirchen is a small town in
Bavaria, near Germany's highest mountain Zugspitze, which rises
to 9,718 feet (2,962 meters)",
                  "It's one of the country's top alpine
destinations, extremely popular during the winter",
                  "In spring, take a road trip through Bavaria
and enjoy the view of the dark green Alps and the first alpine
wildflowers. "]
```

Desired Output:

```
['Unsere erfahrenen Autoren reisen die Welt, um Ihnen informative und inspirierende Funktionen, Destination Rund', 'Jeder Teil Deutschlands ist anders, und es gibt Tausende von denkwürdigen Orten zu besuchen.', 'Weihnachtsmärkte entstanden in Deutschland, und die Tradition stammt aus dem späten Mittelalter.', 'Garmisch-Partenkirchen ist eine kleine Stadt in Bayern, nahe Deutschland.Die Zug', 'Es ist eines der Top-Alpenziele des Landes, sehr beliebt im Winter', 'Im Frühjahr machen Sie eine Roadtrip durch Bayern und genießen den Blick auf die dunkelgrünen Alpen']
```

```
# Install the package
!pip install simpletransformers
# Import the model
from simpletransformers.seq2seq import Seq2SeqModel
# Setting desired arguments
my args = {
               "train batch size": 2,
               "num train epochs": 10,
               "save_eval_checkpoints": False,
               "save_model_every_epoch": False,
               "evaluate during training": True,
               "evaluate_generated_text": True
# Instantiating the model
my model=Seq2SeqModel (encoder decoder name="Helsinki-NLP/opus-
de", encoder_decoder_type="marian", args=my_args, use_cuda=False)
# translating the text
my model.predict(['Our experienced writers travel the world to
bring you informative and inspirational features, destination
roundups, travel ideas, tips and beautiful photos in order to
help you plan your next holiday',
                  'Each part of Germany is different, and there
are thousands of memorable places to visit.',
                  "Christmas Markets originated in Germany, and
the tradition dates to the Late Middle Ages.",
                  "Garmisch-Partenkirchen is a small town in
Bavaria, near Germany's highest mountain Zugspitze, which rises
to 9,718 feet (2,962 meters)",
                  "It's one of the country's top alpine
destinations, extremely popular during the winter",
                  "In spring, take a road trip through Bavaria
and enjoy the view of the dark green Alps and the first alpine
wildflowers. "])
```

```
#> ['Unsere erfahrenen Autoren reisen die Welt, um Ihnen
informative und inspirierende Funktionen, Destination Rund',
#> 'Jeder Teil Deutschlands ist anders, und es gibt Tausende
von denkwürdigen Orten zu besuchen.',
#> 'Weihnachtsmärkte entstanden in Deutschland, und die
Tradition stammt aus dem späten Mittelalter.',
#> 'Garmisch-Partenkirchen ist eine kleine Stadt in Bayern,
nahe Deutschland.Die Zug',
#> 'Es ist eines der Top-Alpenziele des Landes, sehr beliebt im
Winter',
#> 'Im Frühjahr machen Sie eine Roadtrip durch Bayern und
genießen den Blick auf die dunkelgrünen Alpen']
```

57. How to create a Question-Answering system from given context

Difficulty Level: L4

Q. Build a Question Answering model that answers questions from the given context using transformers package

Input:

```
context=""" Harry Potter is the best book series according to
many people. Harry Potter was written by JK.Rowling .
It is afantasy based novel that provides a thrilling experience
to readers."""
question="What is Harry Potter ?"
```

Desired Output:

```
{'score': 0.2375375191101107, 'start': 17, 'end': 37, 'answer': 'the best book series'}
```

Show Solution >

```
#Install and import the pipeline of transformers
!pip install transformers
from transformers import pipeline
# Get thetask-specific pipeline
my_model=pipeline(task="question-answering")
context = r""" Harry Potter is the best book series according
to many people. Harry Potter was written by JK.Rowling .
It is afantasy based novel that provides a thrilling experience
to readers."""
# Pass the question and context to the model to obtain answer
print(my model(question="What is Harry Potter ?",
context=context))
print(my_model(question="Who wrote Harry Potter ?",
context=context))
#> {'score': 0.2375375191101107, 'start': 17, 'end': 37,
'answer': 'the best book series'}
#> {'score': 0.9813234768798256, 'start': 92, 'end': 102,
'answer': 'JK.Rowling'}
```

58. How to do text generation starting from a given piece of text?

Difficulty Level: L4

Q. Generate text based on the the starting provided.

Input:

```
starting="It was a bright"
```

Desired Output:

```
'It was a bright day in New Jersey\'s capitol," the senator told a reporter after the rally. "It\'s a sunny day in New Hampshire, there\'s a great deal of sunshine.'
```

Show Solution >

```
# Import pipeline from transformers package
from transformers import pipeline

# Get the task-specific pipeline
my_model=pipeline(task="text-generation")

# Pass the starting sequence as input to generate text
my_model(starting)

#> [{'generated_text': 'It was a bright day in New Jersey\'s
capitol," the senator told a reporter after the rally. "It\'s a
sunny day in New Hampshire, there\'s a great deal of
sunshine."'}]
```

59. How to classify a text as positive or negative sentiment with transformers?

Difficulty Level: L4

Q. Find out whether a given text is postive or negative sentiment along with score for predictions

Input text:

```
text1="It is a pleasant day, I am going for a walk"
text2="I have a terrible headache"
```

Desired Output:

```
[{'label': 'POSITIVE', 'score': 0.9998570084571838}]
[{'label': 'NEGATIVE', 'score': 0.9994378089904785}]
```

Show Solution ✓

```
# Import pipeline from transformers package
from transformers import pipeline

# Get the task specific pipeline
my_model = pipeline("sentiment-analysis")

# Predicting the sentiment with score
print(my_model(text1))
print(my_model(text2))

[{'label': 'POSITIVE', 'score': 0.9998570084571838}]
[{'label': 'NEGATIVE', 'score': 0.9994378089904785}]
```

I hope you found this useful. For more such posts, stay tuned to our page!

More Articles

Feb 01, 2021

Text Summarization Approaches for NLP – Practical Guide with Generative Examples

Oct 24, 2020

Complete Guide to Natural Language Processing (NLP) – with Practical Examples

Oct 14, 2020

Building chatbot with Rasa and spaCy

Sep 20, 2020

SpaCy Text Classification – How to Train Text Classification Model in spaCy (Solved Example)?

Sep 16, 2020

101 NLP Exercises (using modern libraries)

Jul 14, 2020

Connect with us

Get our new articles, videos and live sessions info.

Join 54,000+ fine folks. Stay as long as you'd like. Unsubscribe anytime.

What's your e-mail? * Yes, Notify Me

We Accept



Empowering you to master Data Science, Al and Machine Learning.

About

About Us
Terms of Use
Privacy Policy
Refund Policy

LEARNING PATHS

Learning Path
Programming for DS
ML Algorithms
ML Ops
Deep Learning
Time Series
DS Industry Projects
Supplementary Courses

OFFERINGS

All Courses
Complete Univ Access
Industry DS Projects
Youtube
List of Blogs
30 Day DS Interviews Prep
Tasklist for DS Projects
DS Project Templates

• HELP

Drop a Query

FAQ's

Contact Us

Testimonials

Subscribe to newsletter

Copyright 2023 | All Rights Reserved by machinelearningplus

- Privacy Policy
- Terms of service
- Terms & Conditions