1. Text tokenization and lower casing
2. Removing special characters
3. Contraction expansion
4. Removing stopwords
5. Stemming
6. Lemmatization
7. Aggregated Features
8. Date-Time Features
9. TF-IDF Features
10. Word Level TF-IDF
11. Character Level TF-IDF
12. Word Embedding Features
13. Count Features
14. Bag-of-n-Grams

```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download("wordnet")
nltk.download("omw-1.4")
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
True
```

```
!pip install contractions
```

```
Collecting contractions
  Downloading contractions-0.1.73-py2.py3-none-any.whl (8.7 kB)
Collecting textsearch>=0.0.21 (from contractions)
  Downloading textsearch-0.0.24-py2.py3-none-any.whl (7.6 kB)
Collecting anyascii (from textsearch>=0.0.21->contractions)
  Downloading anyascii-0.3.2-py3-none-any.whl (289 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 289.9/289.9 kB 4.7 MB/s eta 0:00:00
Collecting pyahocorasick (from textsearch>=0.0.21->contractions)
  Downloading pyahocorasick-2.0.0-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (11
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 110.8/110.8 kB 7.0 MB/s eta 0:00:00
Installing collected packages: pyahocorasick, anyascii, textsearch, contractions
Successfully installed anyascii-0.3.2 contractions-0.1.73 pyahocorasick-2.0.0 textsearch-0.0.24
```

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
# Read data
with open("/content/lab1.docx.txt", "r") as f:
    text = f.read()
```

```
data = text.split('\n')
```

```
['\ufeffDiscussions about AI abound.  But critics have begun to recognize that these debates have focused mostly on technical issues.
That is, there are certain problems that need to be addressed and have technical solutions.  Debate rages, accordingly, about how to
resolve these issues and move the applications of AI forward.  Perhaps Terry Winograd (1996) was correct when he lamented some time ago
that interest in philosophy, what he calls high theory, has dissipated.',
 '        The application of AI, nonetheless, has pushed the discussion beyond technical devices and their possible uses.  For example,
critics have begun to recognize that AI can be quite alienating (Dreyfus 1992; Ritzer 1993).   Workers at Amazon have provided an
interesting case study.  While blaming AI, they claim to be overworked and do not stayed employed for long.  They complain regularly
about manipulation, stress, job insecurity, and so on.  Clearly, AI is not viewed to be their friend (Livingstone 2018; Vicent 2019).',
 '        On the other hand, AI can be quite dangerous.  Take driverless cars!  In this application, through the use of AI cars can
learn how to navigate streets, other cars, pedestrians, and occasionally unanticipated obstacles.  Any failure can result in a
catastrophe, even death.  Questions about the ability of AI to master truly complex activities—those with fluid or shifting frames—have
come to the forefront (Goodfellow, Bengio, and Courville 2016).',
 '        And what about ethical issues?  The focus of developing algorithms, for example, is not necessarily on job loss or intrusions
into privacy.  These issues seems to make context difficult to ignore.  Persons tend to become especially nervous when their jobs or
privacy are threatened.  Although alienation and learning involve context, the ability to survive and the quality of life seem to go to
the heart of the matter.',
 '        The time appears to be ripe, accordingly, to raise the issue of context in the development of advanced technology.  Shoshana
Zuboff (2019) strives to initiate this sort of discussion with her recent foray into the impact of late capitalism on AI.  In this
regard, Kate Crawford (2021) strives to move beyond the “nowhere spaces” where she contends most discussion of AI take place.  She
believes, for example, that AI is enmeshed in the world’s ecology.  While these entreaties are interesting and relevant, the context
provided by Twentieth Century philosophy is missing.  Indeed, the pragmatic framework that is currently the focus of attention of these
efforts provides some interesting insights into whether AI can learn or deal with knotty social issues',
 '        The emphasis of this manuscript, however, is the anti-Cartesian maneuver that characterizes much of contemporary theory.
This change has enormous impact on the potential of AI.  After all, Cartesianism is at the core of digitalization and modern data
processing.  Accordingly, the importance of this change in philosophical orientation for understanding the mind, facts, learning, and
communication is a vital consideration.  Basic to this reassessment is that the limits of AI become obvious in the absence of
Cartesianism.',
 '        Throughout the Western intellectual tradition, true knowledge has been viewed to be timeless (Grayling 2019).  That is, this
information is assumed to be divorced from contexts and other human contingencies.  If immersed in these situations, knowledge can
never surpass opinion and only supply anecdotal evidence.  Therefore, most philosophers sought foundations that are universal to
establish firm epistemological and moral principles.  Martin Heidegger (1969) refers to this trend as the onto-theological tradition.',
 '        In many ways, Cartesianism epitomizes this tendency.  In fact, the aim of Cartesians is to advance clear and distinct
knowledge, severed from opinion and other sources of human error (Bordo 1987).  In this regard, these thinkers are not necessarily
unique, although their strategy is novel.  Rather than speculate about ethereal metaphysical principles, such as Ideas, gods, or cosmic
unity, Cartesians make a straightforward proposal known as dualism.']
```

```python
nltk_tokens = nltk.sent_tokenize(text)
print(nltk_tokens)
```

```
['\ufeffDiscussions about AI abound.', 'But critics have begun to recognize that these debates have focused mostly on technical issues.'
```

```python
import nltk

nltk_tokens_word = nltk.word_tokenize(text)
print (nltk_tokens_word)
```

```
['\ufeffDiscussions', 'about', 'AI', 'abound', '.', 'But', 'critics', 'have', 'begun', 'to', 'recognize', 'that', 'these', 'debates', 'h
```

```python
for i in range(len(nltk_tokens_word)):
    nltk_tokens_word[i] = nltk_tokens_word[i].lower()
    nltk_tokens_word[i] = nltk_tokens_word[i].replace(r"\W",'')
    nltk_tokens_word[i] = nltk_tokens_word[i].expandtabs()
nltk_tokens_word
```

```
['\ufeffdiscussions',
 'about',
 'ai',
 'abound',
 '.',
 'but',
 'critics',
 'have',
 'begun',
 'to',
 'recognize',
 'that',
 'these',
 'debates',
 'have',
 'focused',
 'mostly',
 'on',
 'technical',
 'issues',
 '.',
 'that',
 'is',
 ',',
       'there',
       'are',
       'certain',
       'problems',
       'that',
       'need',
       'to',
       'be',
       'addressed',
       'and',
       'have',
       'technical',
       'solutions',
       '.',
       'debate',
       'rages',
       ',',
       'accordingly',
       ',',
       'about',
       'how',
       'to',
       'resolve',
       'these',
       'issues',
       'and',
       'move',
       'the',
       'applications',
       'of',
       'ai',
       'forward',
       '.',
       'perhaps',
```

```
import contractions

expanded_words = []
for word in text.split():
  # using contractions.fix to expand the shortened words
  expanded_words.append(contractions.fix(word))

expanded_text = ' '.join(expanded_words)
print('Expanded_text: ' + expanded_text)
```

```
Expanded_text: Discussions about AI abound. But critics have begun to recognize that these debates have focused mostly on technical issu
```

Removing Stopwords

```
from nltk.tokenize import word_tokenize


stop_words = set(stopwords.words('english'))

filtered_sentence = [w for w in nltk_tokens_word if not w.lower() in stop_words]
#with no lower case conversion
filtered_sentence = []

for w in nltk_tokens_word:
    if w not in stop_words:
        filtered_sentence.append(w)

print(nltk_tokens_word)
print(filtered_sentence)
```

```
['\ufeffdiscussions', 'about', 'ai', 'abound', '.', 'but', 'critics', 'have', 'begun', 'to', 'recognize', 'that', 'these', 'debates', 'h
['\ufeffdiscussions', 'ai', 'abound', '.', 'critics', 'begun', 'recognize', 'debates', 'focused', 'mostly', 'technical', 'issues', '.',
```

Stemming

```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

ps = PorterStemmer()
```

```
for w in nltk_tokens_word:
    print(w, " : ", ps.stem(w))
```

```
discussions  :  discuss
about  :  about
ai  :  ai
abound  :  abound
.  :  .
but  :  but
critics  :  critic
have  :  have
begun  :  begun
to  :  to
recognize  :  recogn
that  :  that
these  :  these
debates  :  debat
have  :  have
focused  :  focus
mostly  :  mostli
on  :  on
technical  :  technic
issues  :  issu
.  :  .
that  :  that
is  :  is
,  :  ,
there  :  there
are  :  are
certain  :  certain
problems  :  problem
that  :  that
need  :  need
to  :  to
be  :  be
addressed  :  address
and  :  and
have  :  have
technical  :  technic
solutions  :  solut
.  :  .
debate  :  debat
rages  :  rage
,  :  ,
accordingly  :  accordingli
,  :  ,
about  :  about
how  :  how
to  :  to
resolve  :  resolv
these  :  these
issues  :  issu
and  :  and
move  :  move
the  :  the
applications  :  applic
of  :  of
ai  :  ai
forward  :  forward
.  :  .
perhaps  :  perhap
```

```
from nltk.stem import WordNetLemmatizer
nltk.download("wordnet")
nltk.download("omw-1.4")

wnl = WordNetLemmatizer()
# Example inflections to reduce
# Perform lemmatization
print("{0:20}{1:20}".format("--Word--","--Stem--"))
for word in nltk_tokens_word:
   print ("{0:20}{1:20}".format(word, wnl.lemmatize(word, pos="v")))
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
--Word--            --Stem--
discussions        discussions
about               about
ai                  ai
abound              abound
.                   .
```

```
but                 but
critics             critics
have                have
begun               begin
to                  to
recognize           recognize
that                that
these               these
debates             debate
have                have
focused             focus
mostly              mostly
on                  on
technical           technical
issues              issue
.                   .
that                that
is                  be
,                   ,
there               there
are                 be
certain             certain
problems            problems
that                that
need                need
to                  to
be                  be
addressed           address
and                 and
have                have
technical           technical
solutions           solutions
.                   .
debate              debate
rages               rag
,                   ,
accordingly         accordingly
,                   ,
about               about
how                 how
to                  to
resolve             resolve
these               these
issues              issue
and                 and
move                move
the                 the
applications        applications
```

```
vect = TfidfVectorizer().fit(nltk_tokens_word)
X = vect.transform(nltk_tokens_word)
print(X)
```

```
  (0, 98)	1.0
  (1, 11)	1.0
  (2, 20)	1.0
  (3, 10)	1.0
  (5, 51)	1.0
  (6, 84)	1.0
  (7, 145)	1.0
  (8, 45)	1.0
  (9, 314)	1.0
  (10, 254)	1.0
  (11, 297)	1.0
  (12, 304)	1.0
  (13, 91)	1.0
  (14, 145)	1.0
  (15, 129)	1.0
  (16, 204)	1.0
  (17, 220)	1.0
  (18, 291)	1.0
  (19, 174)	1.0
  (21, 297)	1.0
  (22, 172)	1.0
  (24, 302)	1.0
  (25, 36)	1.0
  (26, 63)	1.0
  (27, 240)	1.0
  :	:
  (708, 299)	1.0
  (709, 281)	1.0
  (710, 172)	1.0
  (711, 214)	1.0
  (713, 251)	1.0
  (714, 296)	1.0
  (715, 278)	1.0
  (716, 11)	1.0
  (717, 118)	1.0
  (718, 198)	1.0
  (719, 238)	1.0
  (721, 286)	1.0
  (722, 37)	1.0
  (723, 154)	1.0
  (725, 140)	1.0
  (727, 224)	1.0
  (728, 81)	1.0
  (729, 323)	1.0
  (731, 59)	1.0
  (732, 190)	1.0
  (734, 280)	1.0
  (735, 242)	1.0
  (736, 180)	1.0
  (737, 37)	1.0
  (738, 105)	1.0
```

```
for case in nltk_tokens_word:
    if case.isnumeric():
        print(case)
```

```
1996
1992
1993
2018
2019
2016
2019
2021
2019
1969
1987
```

TF,IDF

```
vect = TfidfVectorizer().fit(nltk_tokens)
X = vect.transform(nltk_tokens)
print(X)
```

```
  (0, 98)	0.6077963696418514
  (0, 20)	0.29343232710714984
  (0, 11)	0.41840723644417843
  (0, 10)	0.6077963696418514
  (1, 314)	0.12412397117587953
  (1, 304)	0.195895158349055
  (1, 297)	0.179300406910585
  (1, 291)	0.24744316457161303
  (1, 254)	0.26883752016703377
  (1, 220)	0.2058256280255748
  (1, 204)	0.2989911707941711
  (1, 174)	0.21728951394447577
  (1, 145)	0.43457902788895153
  (1, 129)	0.2989911707941711
  (1, 91)	0.2989911707941711
  (1, 84)	0.26883752016703377
  (1, 51)	0.2989911707941711
  (1, 45)	0.26883752016703377
  (2, 314)	0.12968049259292613
  (2, 302)	0.31237577997390453
  (2, 297)	0.3746539023848952
  (2, 291)	0.2585201808031976
  (2, 272)	0.31237577997390453
  (2, 240)	0.31237577997390453
  (2, 209)	0.31237577997390453
  :	:
  (38, 208)	0.2924375349945174
  (38, 172)	0.1738160221150615
  (38, 160)	0.1738160221150615
  (38, 36)	0.2363643653556932
  (38, 26)	0.2924375349945174
  (39, 323)	0.21999909011270793
  (39, 296)	0.21999909011270793
  (39, 286)	0.21999909011270793
  (39, 280)	0.21999909011270793
  (39, 278)	0.21999909011270793
  (39, 251)	0.21999909011270793
  (39, 242)	0.21999909011270793
  (39, 238)	0.19781189413656503
  (39, 224)	0.15988263209192122
  (39, 198)	0.21999909011270793
  (39, 190)	0.19781189413656503
  (39, 180)	0.21999909011270793
  (39, 154)	0.21999909011270793
  (39, 140)	0.21999909011270793
  (39, 118)	0.21999909011270793
  (39, 105)	0.21999909011270793
  (39, 81)	0.21999909011270793
  (39, 59)	0.19781189413656503
  (39, 37)	0.39562378827313005
  (39, 11)	0.15144745166630824
```

TF,IDF word level

```
vect = TfidfVectorizer().fit(nltk_tokens_word)
X = vect.transform(nltk_tokens_word)
print(X)
```

```
  (0, 98)	1.0
  (1, 11)	1.0
  (2, 20)	1.0
  (3, 10)	1.0
  (5, 51)	1.0
  (6, 84)	1.0
  (7, 145)	1.0
  (8, 45)	1.0
  (9, 314)	1.0
  (10, 254)	1.0
  (11, 297)	1.0
  (12, 304)	1.0
  (13, 91)	1.0
  (14, 145)	1.0
  (15, 129)	1.0
  (16, 204)	1.0
  (17, 220)	1.0
  (18, 291)	1.0
  (19, 174)	1.0
  (21, 297)	1.0
  (22, 172)	1.0
  (24, 302)	1.0
  (25, 36)	1.0
  (26, 63)	1.0
  (27, 240)	1.0
  :	:
  (708, 299)	1.0
  (709, 281)	1.0
  (710, 172)	1.0
  (711, 214)	1.0
  (713, 251)	1.0
  (714, 296)	1.0
  (715, 278)	1.0
  (716, 11)	1.0
  (717, 118)	1.0
  (718, 198)	1.0
  (719, 238)	1.0
  (721, 286)	1.0
  (722, 37)	1.0
  (723, 154)	1.0
  (725, 140)	1.0
  (727, 224)	1.0
  (728, 81)	1.0
  (729, 323)	1.0
  (731, 59)	1.0
  (732, 190)	1.0
  (734, 280)	1.0
  (735, 242)	1.0
  (736, 180)	1.0
  (737, 37)	1.0
  (738, 105)	1.0
```

TD IDF character lvl

```
from math import log
from collections import Counter
char_counts = Counter(text)
N = len(text)
N
```

```
4173
```

```
tf_idf = {}
for char, tf in char_counts.items():
    tf = tf/N

    n_contining_term = 1

    idf = log(1+(1/n_contining_term))

    tf_idf[char] = tf*idf

print(tf_idf)
```

```
{'\ufeff': 0.00016610284700693633, 'D': 0.000498308541020809, 'i': 0.04302063737479651, 's': 0.04235622598676876, 'c': 0.019932341640832
```

```
!pip install gensim
```

```
Requirement already satisfied: gensim in /usr/local/lib/python3.10/dist-packages (4.3.2)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from gensim) (1.23.5)
Requirement already satisfied: scipy>=1.7.0 in /usr/local/lib/python3.10/dist-packages (from gensim) (1.11.3)
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/python3.10/dist-packages (from gensim) (6.4.0)
```

```
from gensim.models import Word2Vec
word_tokens = [nltk.word_tokenize(text.lower())]
word2vec_model = Word2Vec(word_tokens, vector_size=100, window=5, min_count=1, sg=0)
word2vec_features = word2vec_model.wv['AI']  # Example for the word 'Royce'
```