Importing Libraries

In [1]:

```
!pip install spacy
```

```
Requirement already satisfied: spacy in c:\users\sumit\appdata\roaming\pytho
n\python38\site-packages (2.3.7)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\programdata\ana
conda3\lib\site-packages (from spacy) (2.24.0)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\programdata\a
naconda3\lib\site-packages (from spacy) (1.0.5)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\programdata\anacond
a3\lib\site-packages (from spacy) (4.50.2)
Requirement already satisfied: plac<1.2.0,>=0.9.6 in c:\users\sumit\appdata
\roaming\python\python38\site-packages (from spacy) (1.1.3)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\programdata\anacond
a3\lib\site-packages (from spacy) (2.0.5)
Requirement already satisfied: thinc<7.5.0,>=7.4.1 in c:\users\sumit\appdata
\roaming\python\python38\site-packages (from spacy) (7.4.5)
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in c:\users\sumit\appdata
\roaming\python\python38\site-packages (from spacy) (1.0.5)
Requirement already satisfied: setuptools in c:\programdata\anaconda3\lib\si
te-packages (from spacy) (50.3.1.post20201107)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\programdata\anaco
nda3\lib\site-packages (from spacy) (3.0.5)
Requirement already satisfied: numpy>=1.15.0 in c:\users\sumit\appdata\roami
ng\python\python38\site-packages (from spacy) (1.21.0)
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in c:\programdata\anacon
da3\lib\site-packages (from spacy) (0.8.2)
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in c:\users\sumit\app
data\roaming\python\python38\site-packages (from spacy) (1.0.0)
Requirement already satisfied: blis<0.8.0,>=0.4.0 in c:\programdata\anaconda
3\lib\site-packages (from spacy) (0.7.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda
3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2020.6.20)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\programdata\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->sp
acy) (1.25.11)
Requirement already satisfied: idna<3,>=2.5 in c:\programdata\anaconda3\lib
\site-packages (from requests<3.0.0,>=2.13.0->spacy) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\programdata\anaconda3
\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy) (3.0.4)
```

In [2]:

```
!pip install gensim
```

```
Requirement already satisfied: gensim in c:\users\sumit\appdata\roaming\pyth
on\python38\site-packages (3.8.3)
Requirement already satisfied: numpy>=1.11.3 in c:\users\sumit\appdata\roami
ng\python\python38\site-packages (from gensim) (1.21.0)
Requirement already satisfied: smart-open>=1.8.1 in c:\programdata\anaconda3
\lib\site-packages (from gensim) (5.1.0)
Requirement already satisfied: six>=1.5.0 in c:\programdata\anaconda3\lib\si
te-packages (from gensim) (1.15.0)
Requirement already satisfied: Cython==0.29.14 in c:\users\sumit\appdata\roa
ming\python\python38\site-packages (from gensim) (0.29.14)
Requirement already satisfied: scipy>=0.18.1 in c:\programdata\anaconda3\lib
\site-packages (from gensim) (1.5.2)
```

In [3]:

```
# for text preprocessing
import re
import spacy

from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
import string

# import numpy for matrix operation
import numpy as np
import pandas as pd
# Importing Gensim
import gensim
from gensim import corpora
# to suppress warnings
from warnings import filterwarnings
filterwarnings('ignore')
```

In [4]:

```
nlp = spacy.load('en_core_web_sm')
```

In [5]:

```
D1 = "Who will responsible for students carrier because in two lockdown our study is just
D2 =  "unable to concetrate on study lots of problem occuring during the attending class s
D3 =  "Financial problem no network coverage I don't have money me to buy a laptop or  des
D4= "Financial problem and lack of study materialunable to concetrate on studyno practical
D5= "stay motivated/ lack of resources/ money problem/ health issueUnable to attend online
```

In [6]:

```
# the complete corpus as below:
corpus = [D1,D2,D3,D4,D5]
corpus
```

Out[6]:

```
["Who will responsible for students carrier because in two lockdown our s
tudy is just aweii no practical experience feeling low and lack of confid
enceTaking ongoing online classes is quite tedious. Another issue is stay
ing motivated. My entire family, including myself, became optimistic in t
he face of financial difficulties.Financial problem no network coverageIn
this pandemic no hope from anyone nobody can help in this situation everb
ody facing same problemhowever my college is not helping/ demanding for e
xtra due fee/my financial condition worst in this pandemicnothing Relativ
es are not able to Help. I m helplessI don't have laptop to study unable
to attend classes and environment issuelack of study material no network
coverage in my villagelots of problems nobody can solve it, college fees,
solution is not possibleWho will be responsible for are problems related
to study neither college nor University taking care of their students mos
t of my friends facing the financial problemI am feeling helpless hopeles
sIn my village electricity is not available I have to charge my laptop an
d mobile for attending the classes and still sir  is awarding according t
heir attendance which is not good way because in my village internet is v
ery slowI cannot speak against my college otherwise I will lose internal
```

Text Preprocessing

Steps to preprocess text data:

1. Convert the text into lowercase
2. Split text into words
3. Remove the stop loss words
4. Remove the Punctuation, any symbols and special characters
5. Normalize the word (I'll be using Lemmatization for normalization)

In [7]:

```
# Apply Preprocessing on the Corpus

# stop loss words
stop = set(stopwords.words('english'))

# punctuation
exclude = set(string.punctuation)

# lemmatization
lemma = WordNetLemmatizer()

# One function for all the steps:
def clean(doc):

    # convert text into lower case + split into words
    stop_free = " ".join([i for i in doc.lower().split() if i not in stop])

    # remove any stop words present
    punc_free = ''.join(ch for ch in stop_free if ch not in exclude)

    # remove punctuations + normalize the text
    normalized = " ".join(lemma.lemmatize(word) for word in punc_free.split())
    return normalized

# clean data stored in a new list
clean_corpus = [clean(doc).split() for doc in corpus]
```

In [8]:

```
clean_corpus
```

Out[8]:

```
[['responsible',
  'student',
  'carrier',
  'two',
  'lockdown',
  'study',
  'aweii',
  'practical',
  'experience',
  'feeling',
  'low',
  'lack',
  'confidencetaking',
  'ongoing',
  'online',
  'class',
  'quite',
  'tedious'.
```

Creating Document Term Matrix

Using gensim for Document Term Matrix(DTM), we don't need to create the DTM matrix from scratch explicitly. The gensim library has internal mechanism to create the DTM.

The only requirement for gensis package is we need to pass the cleaned data in the form of tokenized words.

In [9]:

```
# Creating the term dictionary of our courpus that is of all the words (Sepcific to Genism
# where every unique term is assigned an index.

dict_ = corpora.Dictionary(clean_corpus)

# Converting list of documents (corpus) into Document Term Matrix using the dictionary
doc_term_matrix = [dict_.doc2bow(i) for i in clean_corpus]
doc_term_matrix
```

Out[9]:

```
[[(0, 8),
  (1, 1),
  (2, 1),
  (3, 3),
  (4, 1),
  (5, 1),
  (6, 3),
  (7, 1),
  (8, 1),
  (9, 3),
  (10, 4),
  (11, 1),
  (12, 1),
  (13, 1),
  (14, 1),
  (15, 8),
  (16, 2),
  (17, 5),
```

The output implies:

1. Document wise we have the index of the word and its frequency.
2. The 0th word is repeated 1 time, then the 1st word repeated 1 and so on ...

Implementation of LDA

In [10]:

```
# Creating the object for LDA model using gensim library

Lda = gensim.models.ldamodel.LdaModel
```

In [11]:

```
# Running and Training LDA model on the document term matrix.

ldamodel = Lda(doc_term_matrix, num_topics=1, id2word = dict_, passes=20, random_state=0,
```

In [12]:

```
# Prints the topics with the indexes: 0,1,2 :

ldamodel.print_topics()

# we need to manually check whethere the topics are different from one another or not
```

Out[12]:

```
[(0,
  '0.050*"problem" + 0.025*"study" + 0.017*"lot" + 0.015*"financial" + 0.014
*"college" + 0.014*"network" + 0.013*"class" + 0.011*"facing" + 0.010*"mone
y" + 0.009*"sufficient"')]
```

In [13]:

```
#Extracting Topics from the Corpus
print(ldamodel.print_topics(num_topics=1, num_words=30))

# num_topics mean: how many topics want to extract
# num_words: the number of words that want per topic
```

```
[(0, '0.050*"problem" + 0.025*"study" + 0.017*"lot" + 0.015*"financial" + 0.
014*"college" + 0.014*"network" + 0.013*"class" + 0.011*"facing" + 0.010*"mo
ney" + 0.009*"sufficient" + 0.009*"student" + 0.008*"feeling" + 0.008*"canno
t" + 0.008*"pandemic" + 0.008*"laptop" + 0.007*"care" + 0.007*"lack" + 0.007
*"take" + 0.007*"situation" + 0.007*"online" + 0.006*"also" + 0.006*"work" +
0.006*"unable" + 0.006*"attend" + 0.006*"help" + 0.006*"related" + 0.006*"no
thing" + 0.006*"hope" + 0.006*"would" + 0.006*"coverage"')]
```