"""Task for today : dataset - https://archive.ics.uci.edu/ml/datasets/Bag+of+Words

```
q1 = try to find out a count of each and every word in a respective file return a
list of tuple with word and its respective count
   sample example -  [('sudh', 6 ) , ('kumar',3)]
q2 = try to perform a reduce operation to get a count of all the word starting with
same alphabet
    sample examle = [(a,56) , (b,34),..........]
q3 = Try to filter out all the words from dataset .

.001.abstract = abstract
=.002 = delete

q4 = create a tuple set of all the records avaialble in all the five file and then
store it in sqllite DB .
(aah,>=,354,fdsf,wer)

Top 10 will be able to get kids neuron """
```

https://archive.ics.uci.edu/ml/machine-learning-databases/bag-of-words/

In [25]:

```python
import os
import pandas as pd
import sqlite3
import math
import logging
from functools import reduce
from collections import Counter
logging.basicConfig(filename='word_count.log',level=logging.DEBUG, format=' %(asctime)s -%
from functools import reduce
```

q1 = try to find out a count of each and every word in a respective file return a list of tuple with word and its respective count sample example - [('sudh', 6) , ('kumar',3)]

In [4]:

```python
def word_count(filename):
  try:
    if(os.path.getsize(filename) > 0):
      words = []
      with open(filename) as file:
        for line in file:
          words.append(line.rstrip())

      #creating list of tuples of each word and its count
      tup = [(i, words.count(i)) for i in words]

      #sorting the list of tuples taking the second element as key
      tup.sort(key = lambda x: x[1])

      #for every file a new count filename is created
      newfile = "count_%s" %filename

      #writing word count into the new file
      file_writer = open(newfile,'w')
      for ele in tup:
        file_writer.write(f'{ele}\n')
  except Exception as e:
    logging.error("filename %s is empty",filename)


#Passing argument as each filename
files = ['vocab.enron.txt','vocab.kos.txt','vocab.nips.txt','vocab.nytimes.txt','vocab.pub

for i in files:
  word_count(i)
```

In [5]:

```

```

q2 = try to perform a reduce operation to get a count of all the word starting with same alphabet sample examle = [(a,56) , (b,34),..........]

In [ ]:

```
#The function that takes each file, creates list of first word and passes to
#reduce function to get dictionary of key,value pairs {'a':400,'b':500} etc

def count_first(filename):
  try:
    if(os.path.getsize(filename) > 0):
      words = []
      count_map={}

      #storing all first characters into a list
      with open(filename,'r') as file:
        for line in file:
          words.append(line[0].rstrip())


      #counting occurance of each character using reduce function and storing in a dictio
      for i in words:
        count = reduce(lambda a,b: a + (1 if b == i else 0), words, 0)
        count_map[i] = count

      #storing the count of occurances into a list of tuples
      count_list = [(k,v) for k,v in count_map.items()]


      #storing the count of first characters into a new file
      newfile = "count_first_%s" %filename
      with open(newfile,'w') as file_writer:
        for ele in count_list:
          file_writer.write(f'{ele}\n')

  except Exception as e:
    logging.error("cant count first character of file %s as the file is empty", filename)



#passing argument as each filename
files = ['vocab.enron.txt','vocab.kos.txt','vocab.nips.txt','vocab.nytimes.txt','vocab.pub

for i in files:
  count_first(i)
```

q3 = Try to filter out all the words from dataset .

.001.abstract = abstract

=.002 = delete

In [ ]:

```
def filter_words(filename):
  try:
    if(os.path.getsize(filename) > 0):
      logging.info("word extraction problem 3 started for filename %s", filename)
      words = []

      #Checking each word and extracting the word as per the given condition
      with open(filename,'r') as file:
        for line in file:
          if '.001.abstract' in line:
            words.append('abstract \n')
          elif '=.002' in line:
            words.append('delete \n')
          else:
            words.append(line)


      #writing all extracted words into a new filename, "extract_words ***"
      newfile = "extract_words %s" %filename
      file_writer = open(newfile,'w')
      for ele in words:
        file_writer.write(ele)

  except Exception as e:
    logging.error("the file %s has no content", filename)



files = ['vocab.enron.txt','vocab.kos.txt','vocab.nips.txt','vocab.nytimes.txt','vocab.puk

for i in files:
  filter_words(i)
```

q4 = create a tuple set of all the records avaialble in all the five file and then store it in sqllite DB .
(aah,>=,354,fdsf,wer)

In [56]:

```
def tuple_set(filename):

      logging.info("the execution of sqlite 3 tuple set has started")
      tup_set = []


      with open(filename,'r') as file:
        for line in file:
          tup_set.append(line.rstrip())

      #converting the list into a tuple
      tup_set = tuple(tup_set)

      #creating a dynamic sqlite3 db name and table name
      db_name = "tuple_sql.db"


      #creating and connecting to the db
      db = sqlite3.connect(db_name)
      cur = db.cursor()

      #check if table exists, if it does, insert all values,
      #if not then create table then insert

      cur.execute(''' SELECT count(name) FROM sqlite_master WHERE type='table' AND name='t
        #if the count is 1, then table exists
        if cur.fetchone()[0]==1 :
          for ele in tup_set:
            cur.execute("insert or replace into tupletable values(?)",  (ele,))
        else :

          #If the table dont exist create table with unique column to avoid duplicate entrie
          cur.execute("create table tupletable (value text not null, unique(value) on confli
          for ele in tup_set:
            cur.execute("insert or replace into tupletable values(?)", (ele,))


        db.commit()
        db.close()





#passing each file to the tuple_set function
files = ['vocab.enron.txt','vocab.kos.txt','vocab.nips.txt','vocab.nytimes.txt','vocab.pub

for i in files:
  tuple_set(i)
```

In [57]:

```
db = sqlite3.connect('tuple_sql.db')
cur = db.cursor()

cur.execute("select name from sqlite_master where type='table'")
cur.fetchall()
```

Out[57]:

```
[('tupletable',)]
```

In [58]:

```
cur.execute("select count(*) from tupletable")
cur.fetchall()
```

Out[58]:

```
[(234151,)]
```

In [ ]:

```

```