

## Web mining Assignment 1

19BCE2311 Gaurav Singh

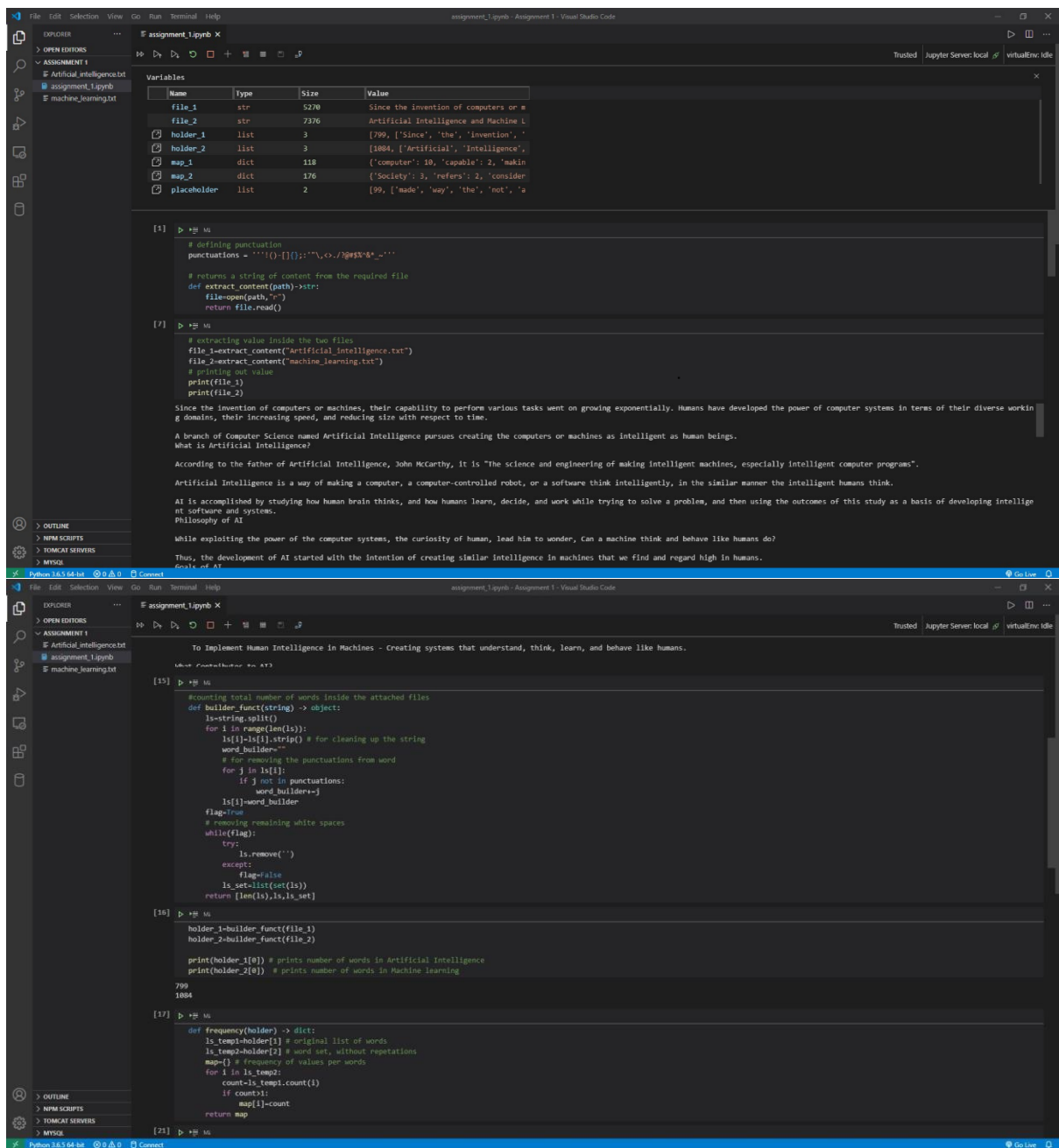
Question:

### **Practice Programming Exercise:**

Write a python program to

- Extract content from two text files attached
- Count the total number of words in each text file
- Count the frequency of repetition of each word found in each file.
- Group in a list the words common for two text files and show their total count

Screenshots :



```

[21] In [ ]:
map_1=frequency(holder_1)
map_2=frequency(holder_2)
#prints words and their frequency as defined by keys and values
print(map_1)
print(map_2)

('computer': 10, 'capable': 2, 'making': 2, 'understand': 2, 'the': 4, 'expert': 2, 'manner': 2, 'advice': 2, 'terms': 2, 'similar': 2, 'some': 2, 'machines': 4, 'can': 10, 'language': 2, 'affecting': 2, 'be': 3, 'systems': 3, 'AI': 20, 'creating': 2, 'intelligence': 5, 'it': 8, 'which': 3, 'text': 2, 'to': 2, 'way': 2, 'their': 5, 'following': 2, 'questions': 2, 'users': 2, 'what': 3, 'perform': 2, 'without': 3, 'to': 25, 'change': 2, 'real': 2, 'reasoning': 2, 'they': 2, 'of': 33, 'different': 2, 'sensors': 2, 'how': 2, 'solve': 3, 'provide': 2, 'should': 3, 'they': 3, 'structure': 2, 'the': 39, 'not': 2, 'tasks': 2, 'answer': 2, 'new': 2, 'robots': 2, 'in': 10, 'program': 2, 'technique': 2, 'areas': 3, 'use': 3, 'recognition': 2, 'without': 2, 'from': 2, 'that': 5, 'machine': 3, 'development': 2, 'artificial': 5, 'A': 5, 'world': 2, 'based': 2, 'modification': 2, 'like': 2, 'is': 17, 'systems': 7, 'humans': 7, 'it': 9, 'have': 3, 'as': 8, 'meant': 2, 'information': 4, 'exhibit': 2, 'and': 28, 'think': 5, 'multiple': 2, 'its': 3, 'with': 8, 'while': 2, 'in': 2, 'science': 2, 'learning': 2, 'power': 2, 'possible': 2, 'or': 8, 'with': 2, 'knowledge': 3, 'a': 15, 'etc': 2, 'computer': 2, 'intelligent': 8, 'huge': 2, 'computer': 2, 'human': 4, 'has': 2, 'speed': 2, 'programming': 3, 'noise': 2, 'learn': 3, 'science': 2, 'program': 2, 'are': 5, 'by': 8, 'software': 5, 'lead': 2, 'problem': 2, 'intelligence': 4, 'various': 2, 'on': 6, 'such': 6, 'system': 2, 'recognize': 2, 'behave': 2)
('society': 3, 'refers': 2, 'considerations': 7, 'paper': 6, 'computer': 2, 'perceive': 2, 'artificial': 9, 'the': 3, 'force': 2, 'spam': 2, 'interact': 3, 'sound': 2, 'it': 2, 'guiding': 3, 'stakeholders': 4, 'recent': 3, 'applications': 2, 'will': 3, 'recognition': 2, 'recognizes': 2, 'we': 6, 'environment': 2, 'experience': 2, 'associated': 2, 'comes': 2, 'can': 7, 'language': 2, 'already': 3, 'could': 2, 'be': 8, 'opportunities': 3, 'focus': 2, 'at': 2, 'technology': 9, 'including': 4, 'purchases': 2, 'AI': 37, 'intelligence': 3, 'narrow': 2, 'internet': 16, 'it': 3, 'impact': 2, 'natural': 2, 'which': 2, 'of': 2, 'all': 2, 'social': 2, 'eg': 2, 'when': 4, 'everyday': 2, 'way': 3, 'their': 4, 'approach': 2, 'people': 2, 'policy': 4, 'creation': 3, 'also': 2, 'users': 3, 'important': 2, 'perform': 2, 'driving': 2, 'recommendations': 5, 'to': 17, 'ethical': 4, 'your': 3, 'services': 5, 'ensuring': 2, 'of': 33, 'different': 2, 'advanced': 2, 'understanding': 3, 'societal': 5, 'email': 2, 'how': 2, 'paints': 2, 'future': 2, 'bring': 2, 'there': 3, 'about': 2, 'this': 3, 'domain': 2, 'help': 2, 'particular': 2, 'our': 2, 'security': 2, 'made': 3, 'plan': 2, 'the': 61, 'not': 3, 'tasks': 3, 'set': 3, 'new': 8, 'critical': 2, 'principles': 5, 'in': 21, 'activity': 2, 'rapidly': 2, 'trust': 3, 'this': 5, 'through': 4, 'for': 10, 'years': 2, 'use': 6, 'only': 2, 'specific': 5, 'traditionally': 2, 'today': 2, 'deployment': 4, 'debates': 2, 'from': 2, 'what': 2, 'that': 15, 'accountability': 3, 'provides': 2, 'machine': 5, 'development': 2, 'artificial': 7, 'world': 2, 'lives': 2, 'based': 2, 'look': 2, 'field': 2, 'particularly': 2, 'an': 7, 'uses': 2, 'humanlike': 2, 'is': 20, 'challenges': 7, 'significant': 3, 'systems': 2, 'emails': 2, 'many': 2, 'regarding': 2, 'as': 6, 'include': 3, 'other': 4, 'and': 43, 'its': 5, 'with': 9, 'in': 3, 'surrounding': 3, 'make': 3, 'science': 3, 'issues': 4, 'as': 3, 'transparency': 2, 'key': 4, 'impacts': 4, 'learning': 6, 'reason': 2, 'safety': 3, 'possible': 2, 'or': 6, 'bias': 2, 'offers': 3, 'a': 10, 'more': 3, 'development': 2, 'developing': 2, 'algorithms': 2, 'has': 7, 'but': 3, 'behind': 4, 'data': 3, 'grow': 2, 'potential': 4, 'these': 2, 'decisions': 2, 'learn': 5, 'are': 10, 'by': 4, 'intelligence': 13, 'an': 6, 'such': 2, 'economic': 3, 'process': 2)

[22] In [ ]:
# function to find common words inside both the files and their total count
def merge_count(holder_1,holder_2) -> object:
    set1=set(holder_1[2])
    set2=set(holder_2[2])
    list_temp=list(set1.intersection(set2))
    return [len(list_temp),list_temp]

[23] In [ ]:
placeholder=merge_count(holder_1,holder_2)
#prints list of words common in both the files and their total number
print(placeholder[0])
print(placeholder[1])

99
[made, 'way', 'the', 'not', 'and', 'think', 'their', 'functions', 'its', 'tasks', 'people', 'with', 'while', 'work', 'paper', 'computer', 'new', 'programming', 'making', 'in', 'understand', 'the', 'science', 'users', 'in', 'interact', 'what', 'perform', 'learning', 'possible', 'way', 'use', 'to', 'on', 'change', 'sound', 'a', 'complex', 'applications', 'specific', 'intelligence', 'you', 'applications', 'recognition', 'developing', 'of', 'we', 'environment', 'has', 'from', 'some', 'different', 'that', 'associated', 'fields', 'data', 'machine', 'development', 'artificial', 'used', 'can', 'language', 'how', 'these', 'be', 'speech', 'world', 'provide', 'based', 'example', 'technology', 'increasing', 'time', 'an', 'learn', 'AI', 'like', 'creating', 'intelligence', 'is', 'this', 'it', 'trying', 'are', 'developed', 'natural', 'systems', 'by', 'which', 'intelligence', 'various', 'on', 'they', 'while', 'humans', 'many', 'such', 'have', 'as']

```

Code :

```

# defining punctuation
punctuations = '!"()-[]{};:'"\.,<>./?@#$$%^&*~`''

# returns a string of content from the required file
def extract_content(path)->str:
    file=open(path,"r")
    return file.read()

# extracting value inside the two files
file_1=extract_content("Artificial_intelligence.txt")
file_2=extract_content("machine_learning.txt")
# printing out value
print(file_1)
print(file_2)

#counting total number of words inside the attached files
def builder_funct(string) -> object:
    ls=string.split()
    for i in range(len(ls)):
        ls[i]=ls[i].strip() # for cleaning up the string
        word_builder=""
        # for removing the punctuations from word
        for j in ls[i]:
            if j not in punctuations:
                word_builder+=j
        ls[i]=word_builder

```

```

flag=True
# removing remaining white spaces
while(flag):
    try:
        ls.remove(' ')
    except:
        flag=False
    ls_set=list(set(ls))
return [len(ls),ls,ls_set]

holder_1=builder_funct(file_1)
holder_2=builder_funct(file_2)

print(holder_1[0]) # prints number of words in Artificial Intelligence
print(holder_2[0]) # prints number of words in Machine learning

def frequency(holder) -> dict:
    ls_temp1=holder[1] # original list of words
    ls_temp2=holder[2] # word set, without repetitions
    map={} # frequency of values per words
    for i in ls_temp2:
        count=ls_temp1.count(i)
        if count>1:
            map[i]=count
    return map

map_1=frequency(holder_1)
map_2=frequency(holder_2)
#prints words and their frequency as defined by keys and values
print(map_1)
print(map_2)

# function to find common words inside both the files and their total count
def merge_count(holder_1,holder_2) -> object:
    set1=set(holder_1[2])
    set2=set(holder_2[2])
    list_temp=list(set1.intersection(set2))
    return [len(list_temp),list_temp]

placeholder=merge_count(holder_1,holder_2)
#prints list of words common in both the files and their total number
print(placeholder[0])
print(placeholder[1])

```