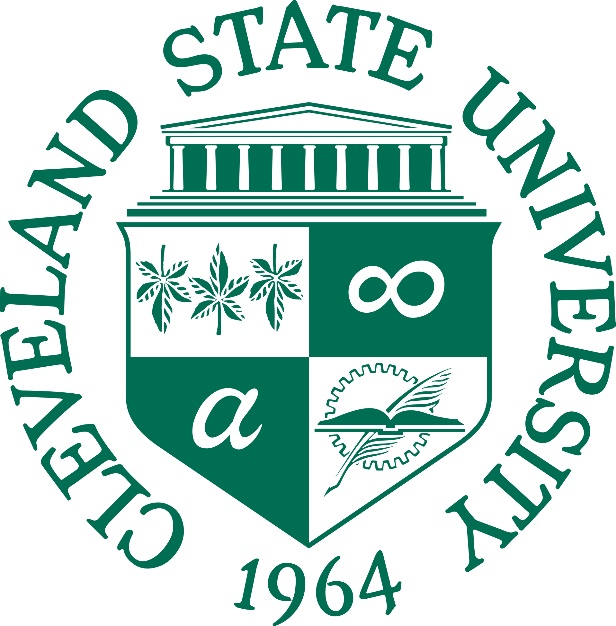
**Department of Computer Science**

**Cleveland State University**

**Cleveland, Ohio.**



**Fall 2020**

|  |  |
| --- | --- |
| **Lab # 4** | |
| Students Name & ID | Maaz Muhammad Khan  2788572 |
| Due date | 3/31/2020 |
| Teacher | Dr. Sunnie S Chung |
| Subject / Course | Big Data |

Contents

[Abstract: 3](#_Toc52141978)

[Tool: 4](#_Toc52141979)

[Steps: 4](#_Toc52141980)

[References: 10](#_Toc52141981)

# 

# Abstract:

This report deals with the text analysis to find the Top N most related documents in a

collection per a given user query (topics) in a Question Answering (QA) System, each

document can be transformed to be represented as a vector of weights on the topic

terms (topic words/keywords/phrases in bi-gram or a tri-gram) in TF-IDF..

# Tool:

Python is used as primary language for coding on Spyder IDE.

Additional Information:

I was facing error in scrapping text for edx website . I got a permission refusal from edx website that’s why I only considered 5 documents to implement my lab 4.

Document 1= "https://en.wikipedia.org/wiki/Engineering",

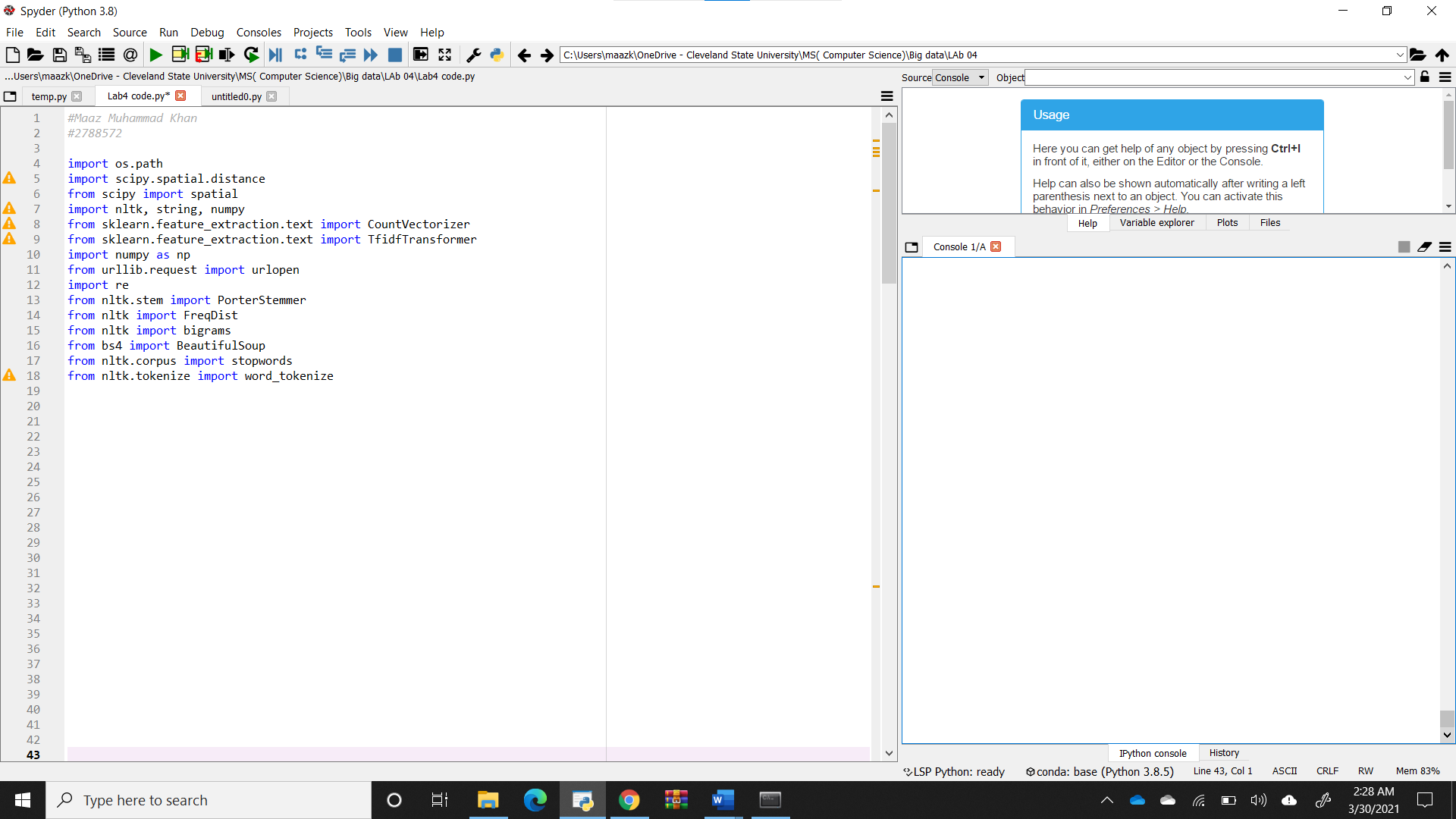
Document 2= "http://my.clevelandclinic.org/research",

Document 3= "https://en.wikipedia.org/wiki/Data\_mining",

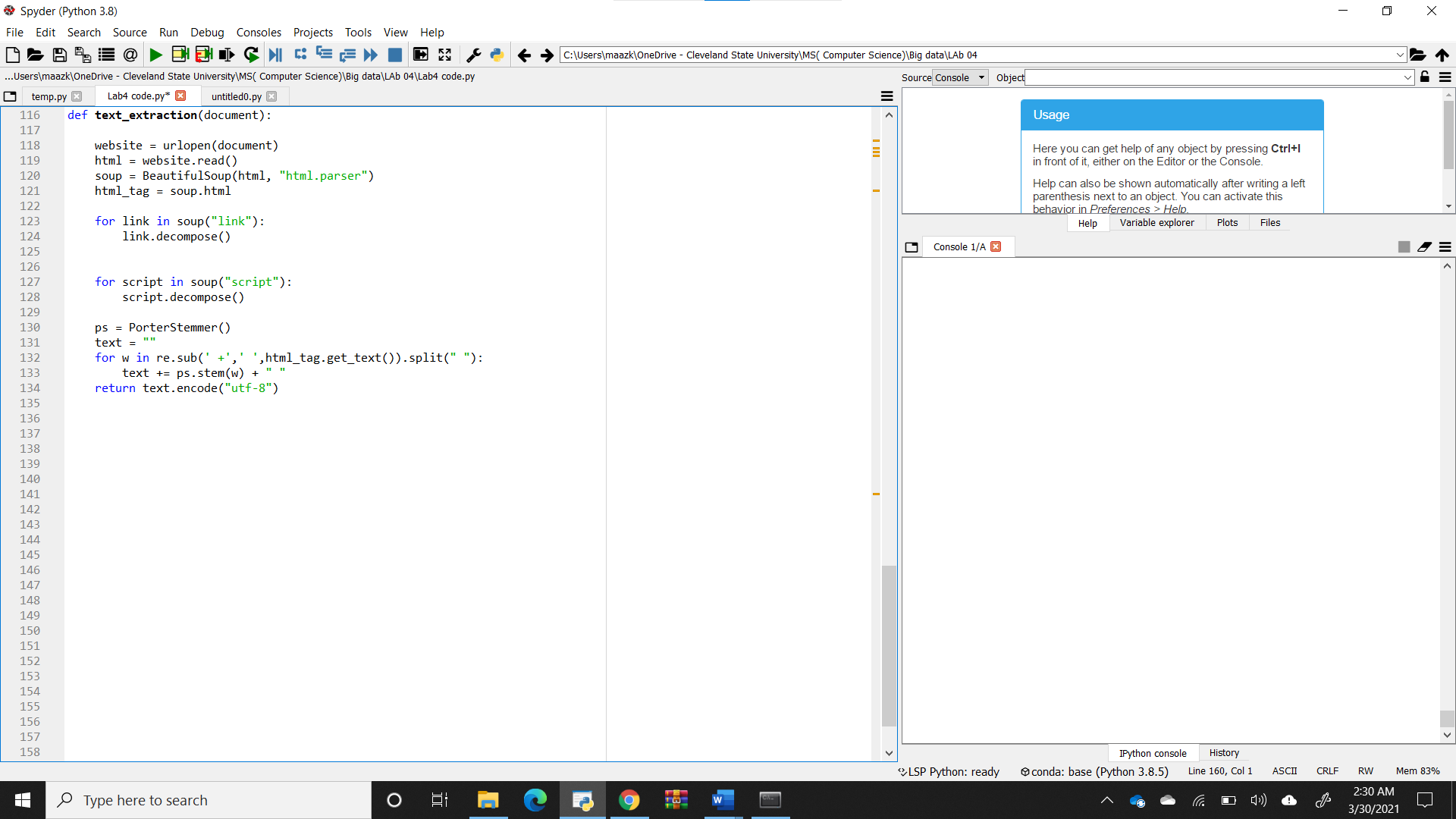
Document 4= "https://en.wikipedia.org/wiki/Data\_mining#Data\_mining",

Document 5= "http://cis.csuohio.edu/~sschung/"

# Steps:

1. Setting up all the libraries and packages for use in project
2. Part 1: Preprocessing to Build Document Vectors for Web Page Content Analysis

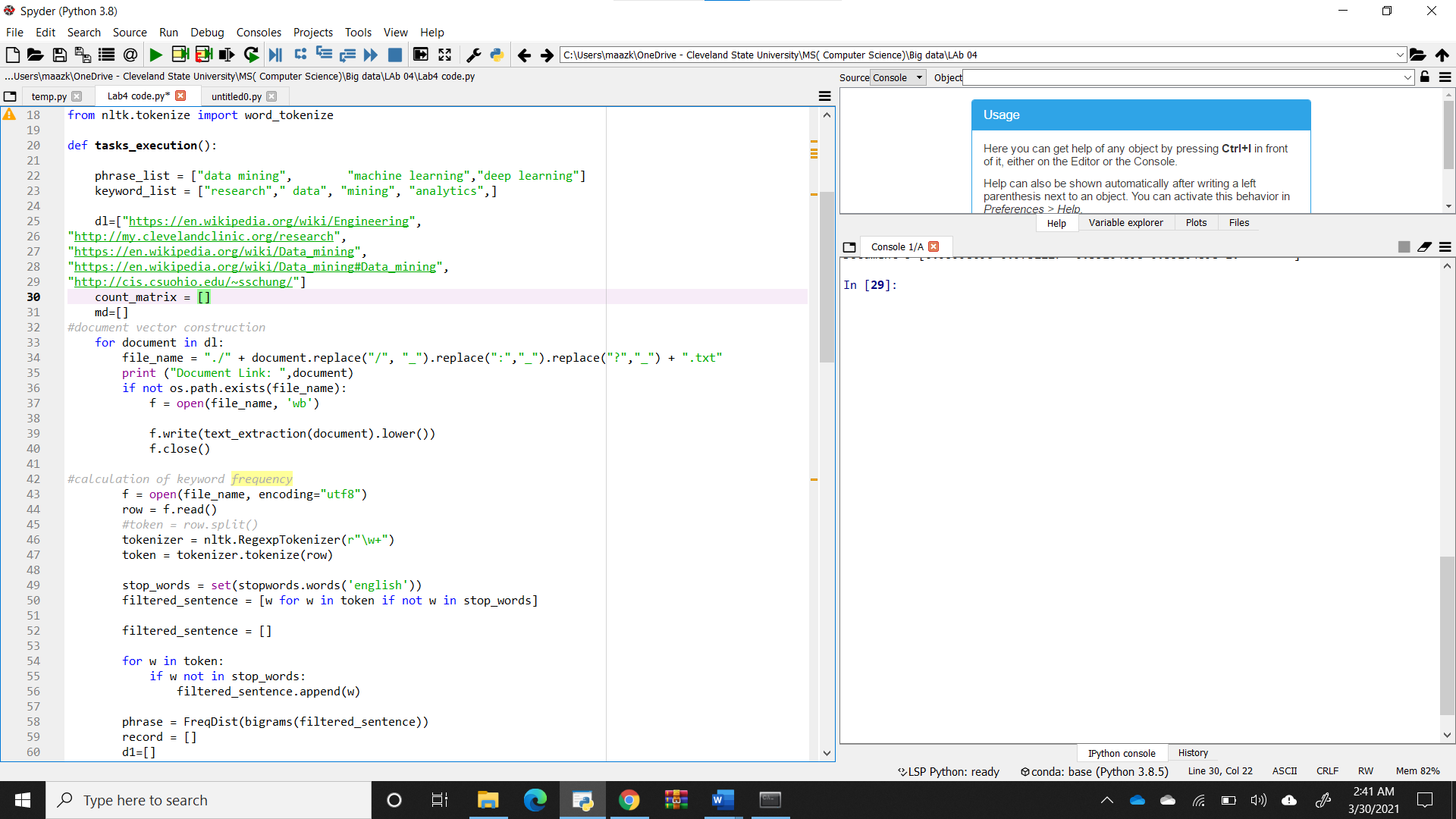
Then wrote a script for text extraction from website and then applied porter stemming on the text

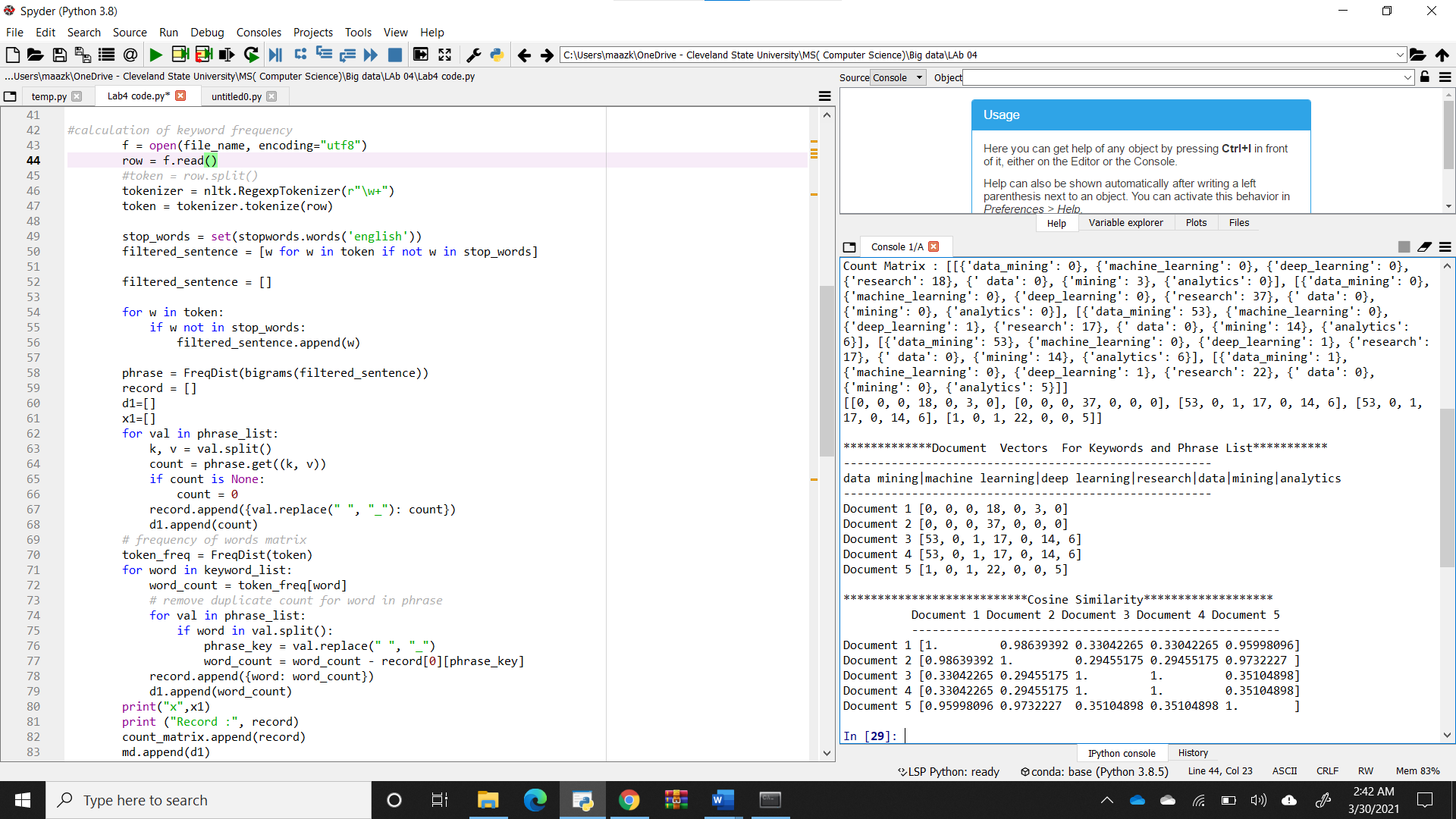


1. Converted text into lower while saving in a text file.

Removed stop words using NLTK

Applied [RegexpTokenizer.tokenize(text)](https://kite.com/python/docs/nltk.tokenize.RegexpTokenizer.tokenize) with RegexpTokenizer , it returns text as a list of words with punctuation's removed.



1. Created Data vector of keyword\_list and phrase\_List for every document. 

Document Link: https://en.wikipedia.org/wiki/Engineering

x []

Record : [{'data\_mining': 0}, {'machine\_learning': 0}, {'deep\_learning': 0}, {'research': 18}, {' data': 0}, {'mining': 3}, {'analytics': 0}]

Document Link: http://my.clevelandclinic.org/research

x []

Record : [{'data\_mining': 0}, {'machine\_learning': 0}, {'deep\_learning': 0}, {'research': 37}, {' data': 0}, {'mining': 0}, {'analytics': 0}]

Document Link: https://en.wikipedia.org/wiki/Data\_mining

x []

Record : [{'data\_mining': 53}, {'machine\_learning': 0}, {'deep\_learning': 1}, {'research': 17}, {' data': 0}, {'mining': 14}, {'analytics': 6}]

Document Link: https://en.wikipedia.org/wiki/Data\_mining#Data\_mining

x []

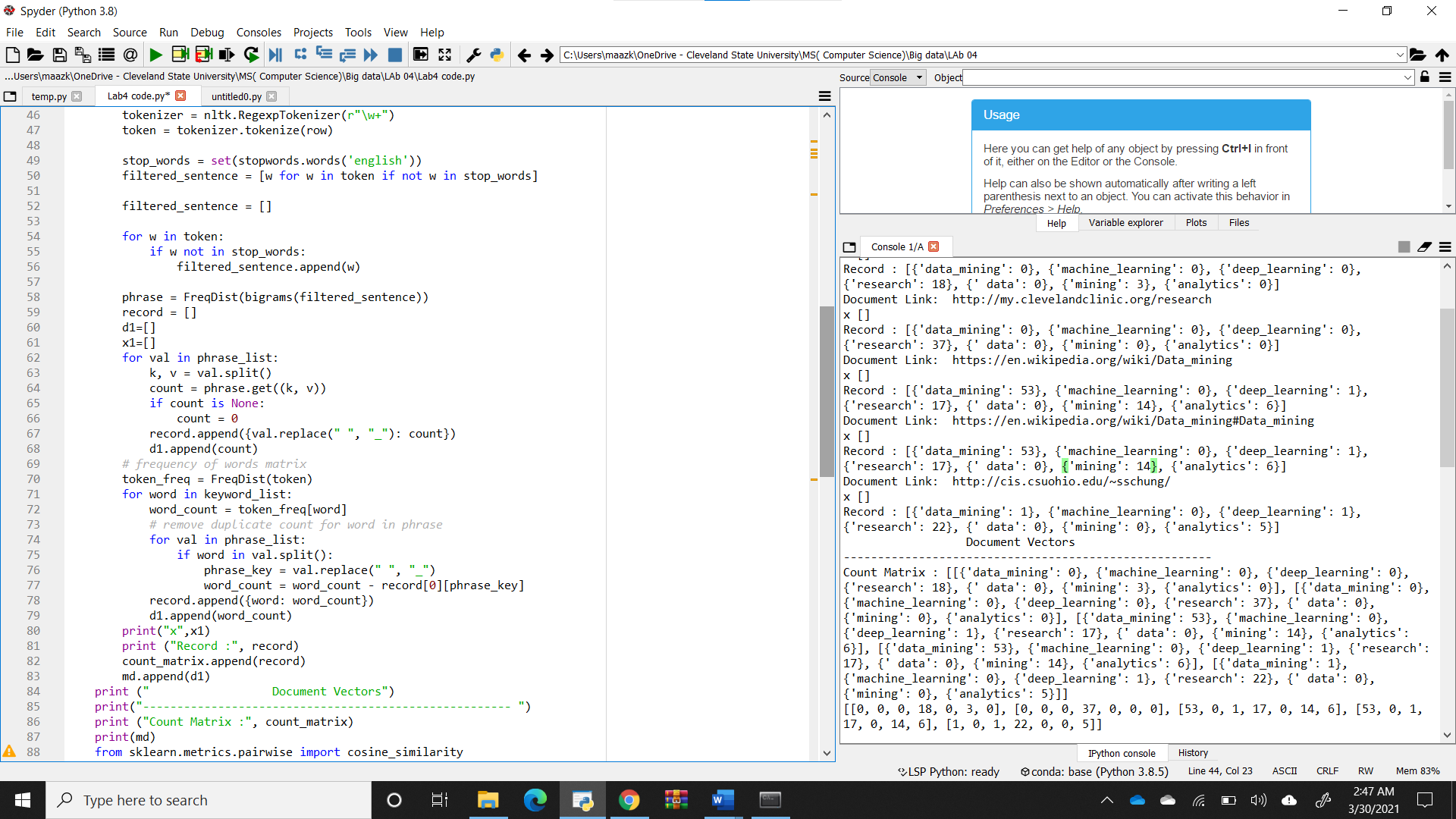
Record : [{'data\_mining': 53}, {'machine\_learning': 0}, {'deep\_learning': 1}, {'research': 17}, {' data': 0}, {'mining': 14}, {'analytics': 6}]

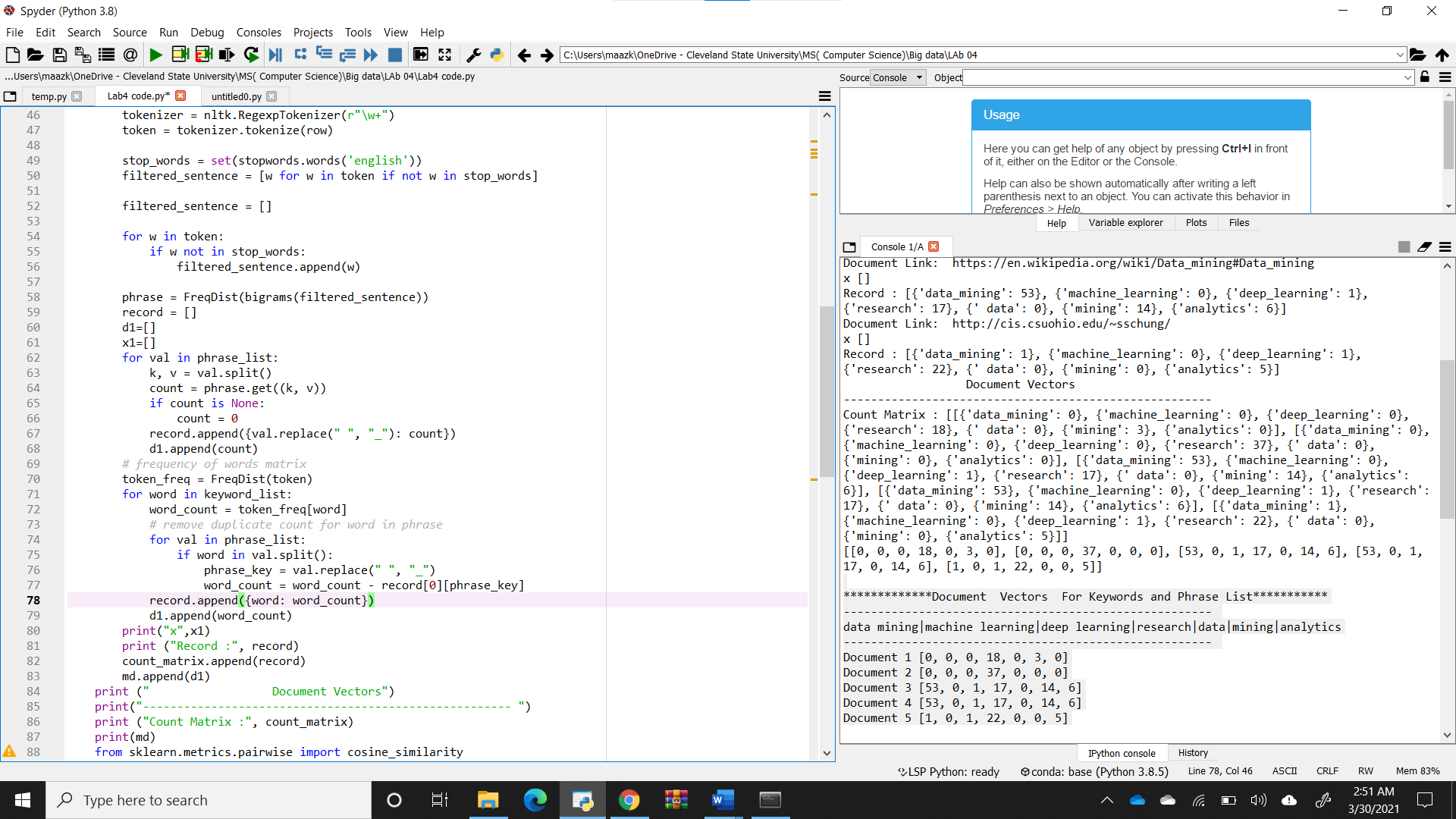
Document Link: http://cis.csuohio.edu/~sschung/

x []

Record : [{'data\_mining': 1}, {'machine\_learning': 0}, {'deep\_learning': 1}, {'research': 22}, {' data': 0}, {'mining': 0}, {'analytics': 5}]

Document Vectors

1. Then creating count matrix Count Matrix : [[{'data\_mining': 0}, {'machine\_learning': 0}, {'deep\_learning': 0}, {'research': 18}, {' data': 0}, {'mining': 3}, {'analytics': 0}], [{'data\_mining': 0}, {'machine\_learning': 0}, {'deep\_learning': 0}, {'research': 37}, {' data': 0}, {'mining': 0}, {'analytics': 0}], [{'data\_mining': 53}, {'machine\_learning': 0}, {'deep\_learning': 1}, {'research': 17}, {' data': 0}, {'mining': 14}, {'analytics': 6}], [{'data\_mining': 53}, {'machine\_learning': 0}, {'deep\_learning': 1}, {'research': 17}, {' data': 0}, {'mining': 14}, {'analytics': 6}], [{'data\_mining': 1}, {'machine\_learning': 0}, {'deep\_learning': 1}, {'research': 22}, {' data': 0}, {'mining': 0}, {'analytics': 5}]
2. Calculation of Data vector of keyword\_list and phrase\_List and created a matrix .



\*\*\*\*\*\*\*\*\*\*\*\*\*Document Vectors For Keywords and Phrase List\*\*\*\*\*\*\*\*\*\*\*

------------------------------------------------------

data mining|machine learning|deep learning|research|data|mining|analytics

------------------------------------------------------

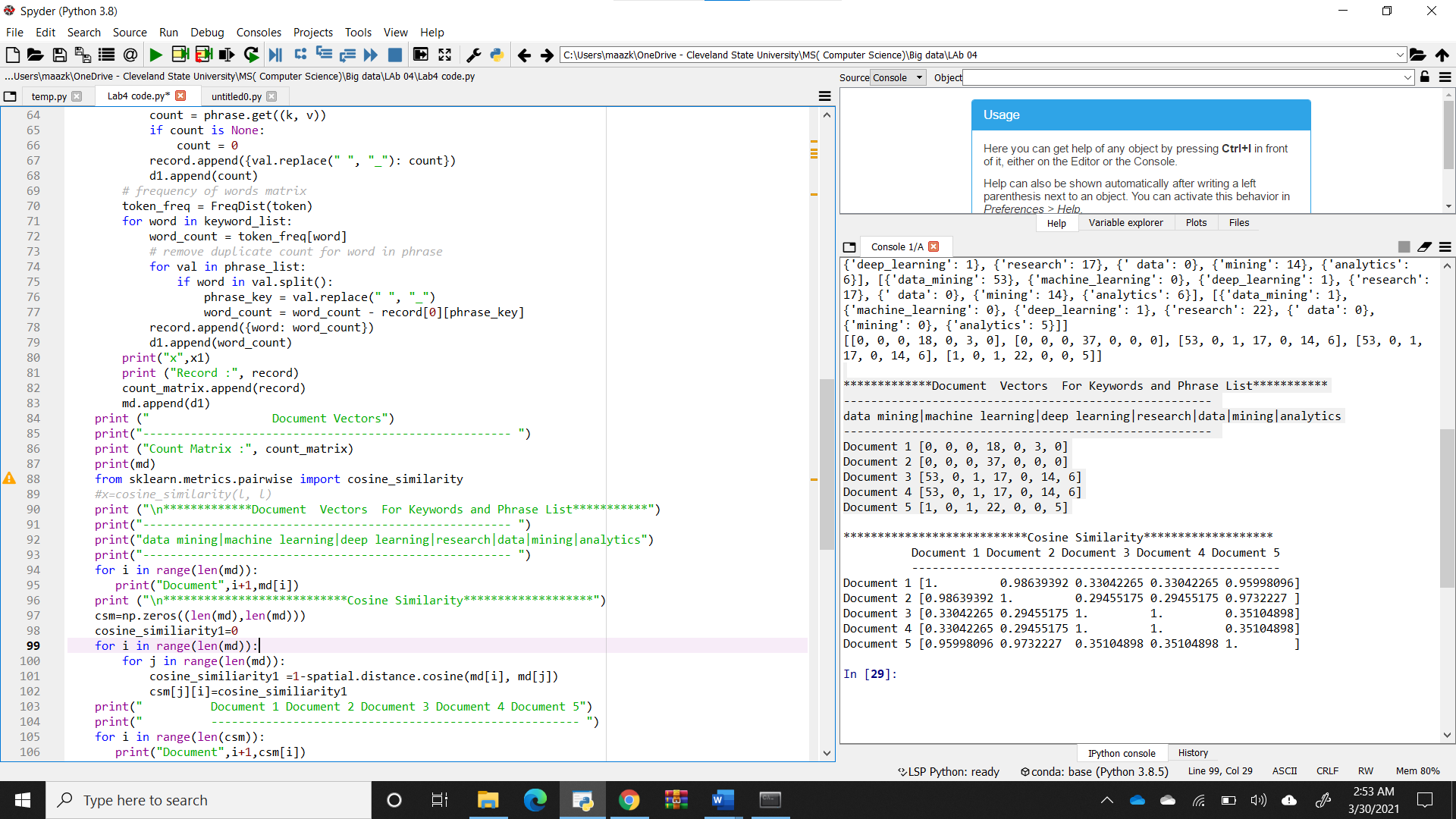
Document 1 [0, 0, 0, 18, 0, 3, 0]

Document 2 [0, 0, 0, 37, 0, 0, 0]

Document 3 [53, 0, 1, 17, 0, 14, 6]

Document 4 [53, 0, 1, 17, 0, 14, 6]

Document 5 [1, 0, 1, 22, 0, 0, 5]

1. Then calculated cosine similarity using scipy built-in method. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Cosine Similarity\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Cosine Similarity\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Document 1 Document 2 Document 3 Document 4 Document 5

------------------------------------------------------

Document 1 [1. 0.98639392 0.33042265 0.33042265 0.95998096]

Document 2 [0.98639392 1. 0.29455175 0.29455175 0.9732227 ]

Document 3 [0.33042265 0.29455175 1. 1. 0.35104898]

Document 4 [0.33042265 0.29455175 1. 1. 0.35104898]

Document 5 [0.95998096 0.9732227 0.35104898 0.35104898 1. ]

1. Part3 : Analysis and discussion.

**Discuss briefly about your topic analysis with your cosine similarity matrix focusing on that:**

**Whether each value (in Cosine Sim) of each pair of any two docs indicate the**

**similarity correctly?**

Cosine similarity comes out 1 for a documen when calculated with itself.

**Which 2 docs are most similar in terms of 7 given topics?**

Document 4 and 5 are most similar.

**The Topics of Doc6 is similar to the Topics of Doc 4 and 5?**

**Explain Why or Why Not in terms of 7 TFs? If not, what are the reasons?**

Because their content is all about data mining and have similar words.

Their word count and phrases count of topic words is same.

# References:

Wikipedia

Microsoft.com

http://eecs.csuohio.edu/~sschung/CIS660/CIS660F20.html#Lab