**BIG DATA LAB4**

**Project Report**

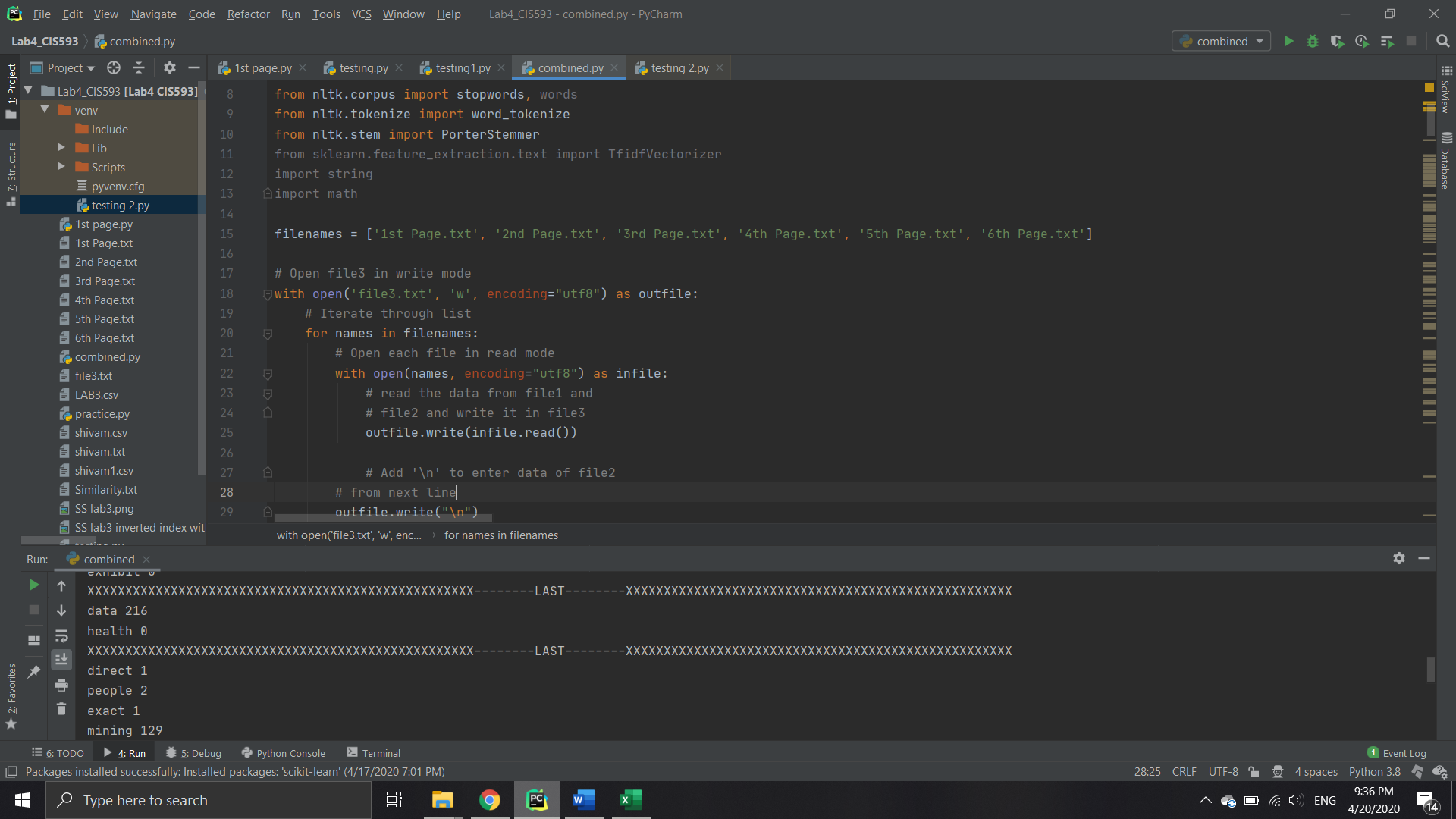
**Name – Shivam Joshi** **CSU ID – 2781474**

**Extra Credit – Done**

|  |  |
| --- | --- |
| About Extra Credit | Detailed explanation. |
| Code Explanation | 1. File Reading 2. Lower Case 3. Numeric Value Removal 4. White Space Removal 5. Punctuation Removal 6. Stop Word Removal |
| Code Explanation 2 | 1. Frequency of words 2. Stem Word Removal |
| Code Explanation 3 | 1. Cosine Similarities |
| Analysis of Results | Explanation of the results |

**About Extra Credit** – I have implemented the extra credit task in which I have created a document dictionaries in which I have stored the words which can be very common to each text file.

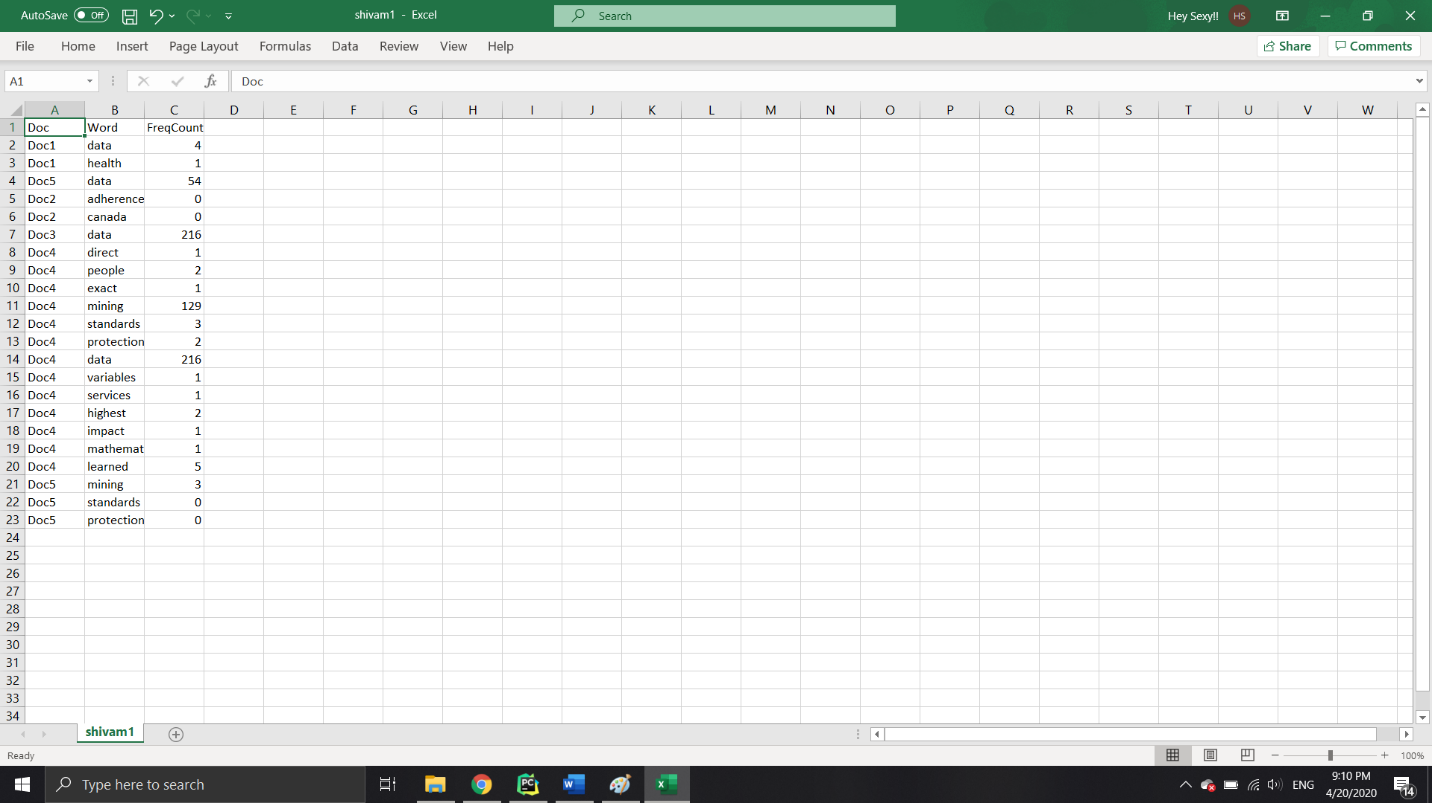
And with the help of that dictionaries, I have compared each word in every document file and took the count of those words in each respective file.



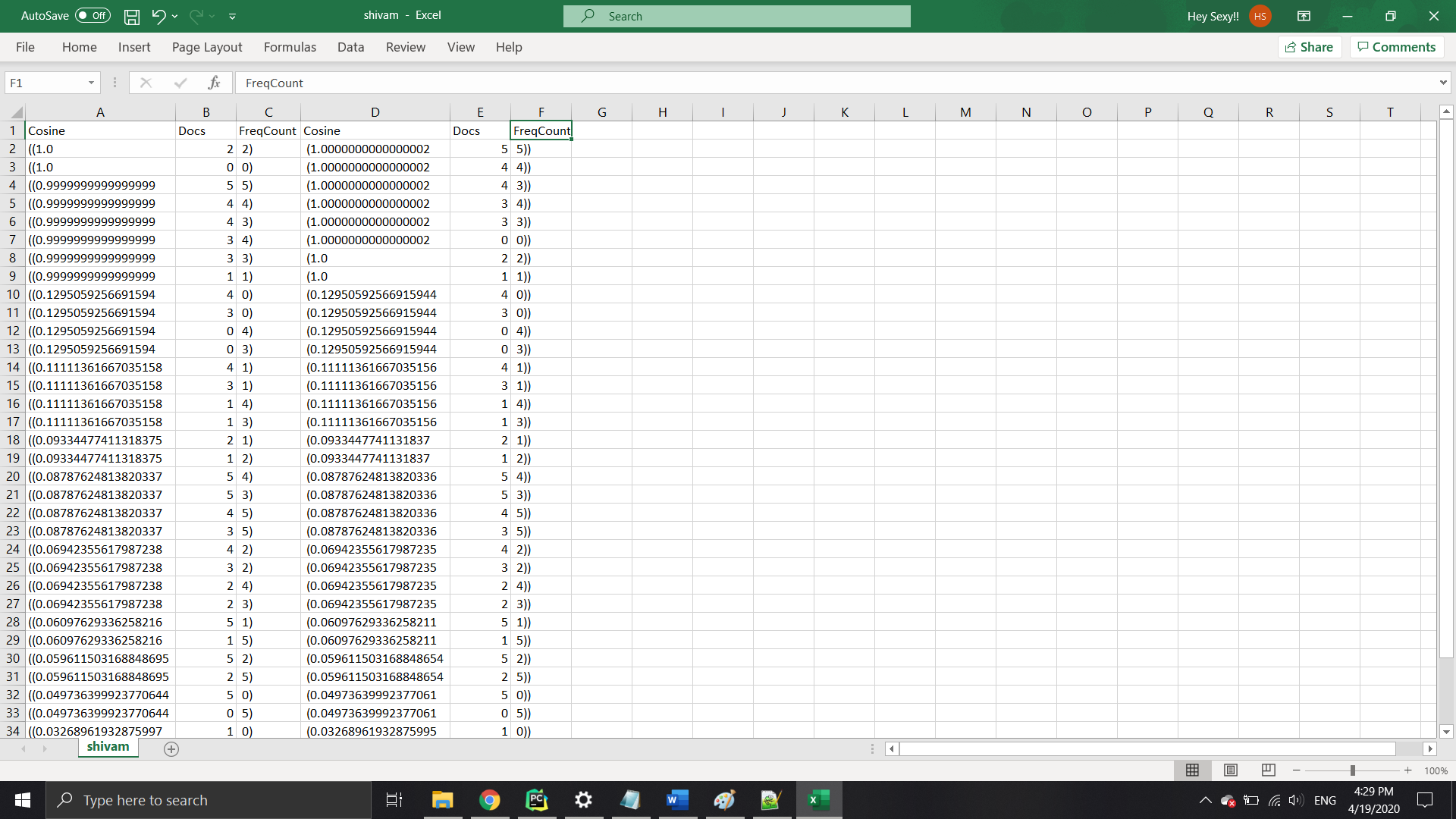
Like the above screenshot. I have stored every text file in one big text file.

Every word and there count are stored in the table.

If every word in my dictionary matches with all the document than only it will save that word in the Inverted Index. After this I have implemented the cosine similarities on those words.



**Cosine Similarities**



**Code Explanation with output**

Code start with opening and reading the respective text file. on this text file there are multiple operation done such as

File = fopen.read()

To convert every upper case to lower case

File = file.lower()

To remove all the numeric value

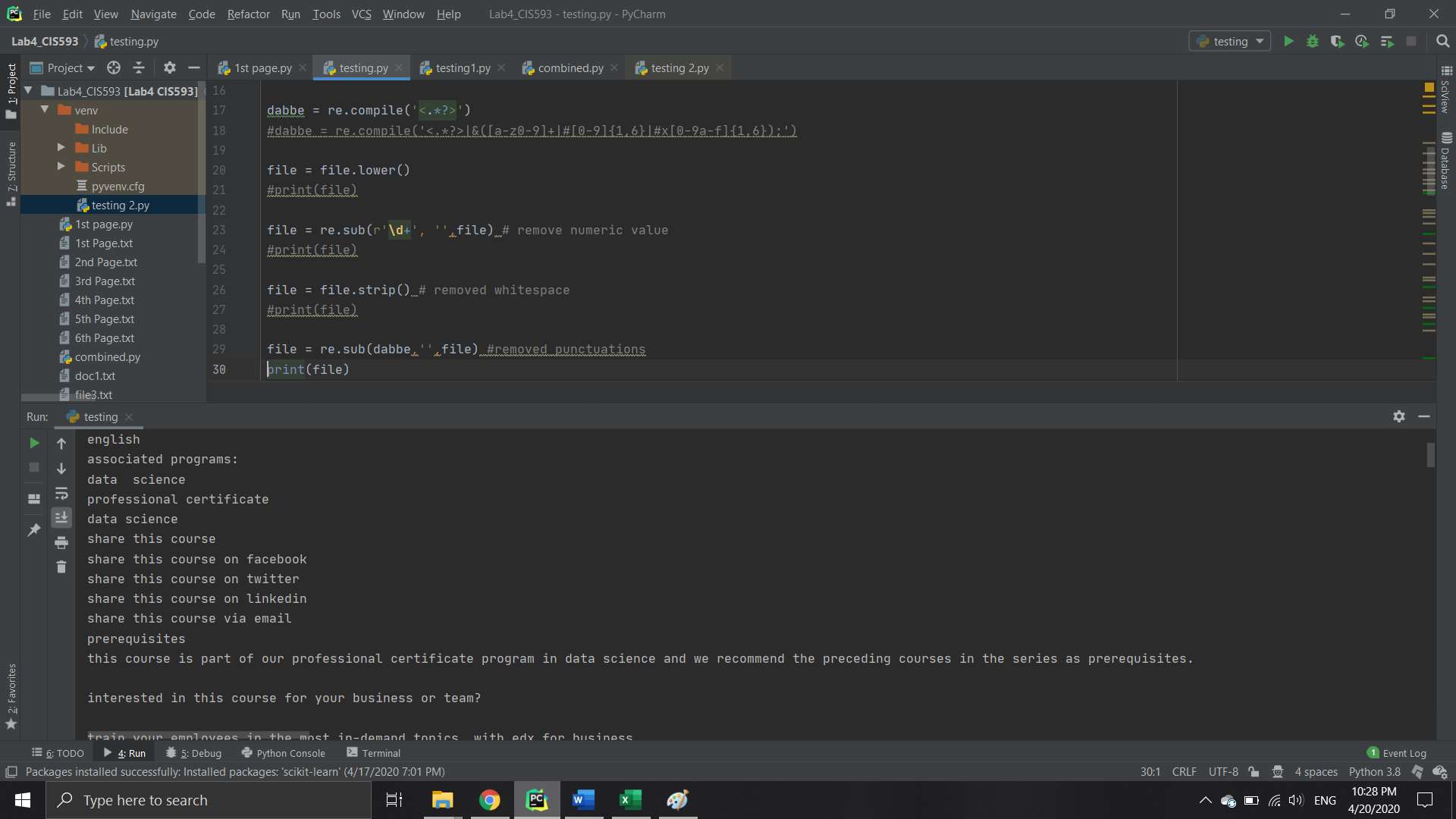
File = re.sub(r’\d+’,”,file)

To remove all the whitespace

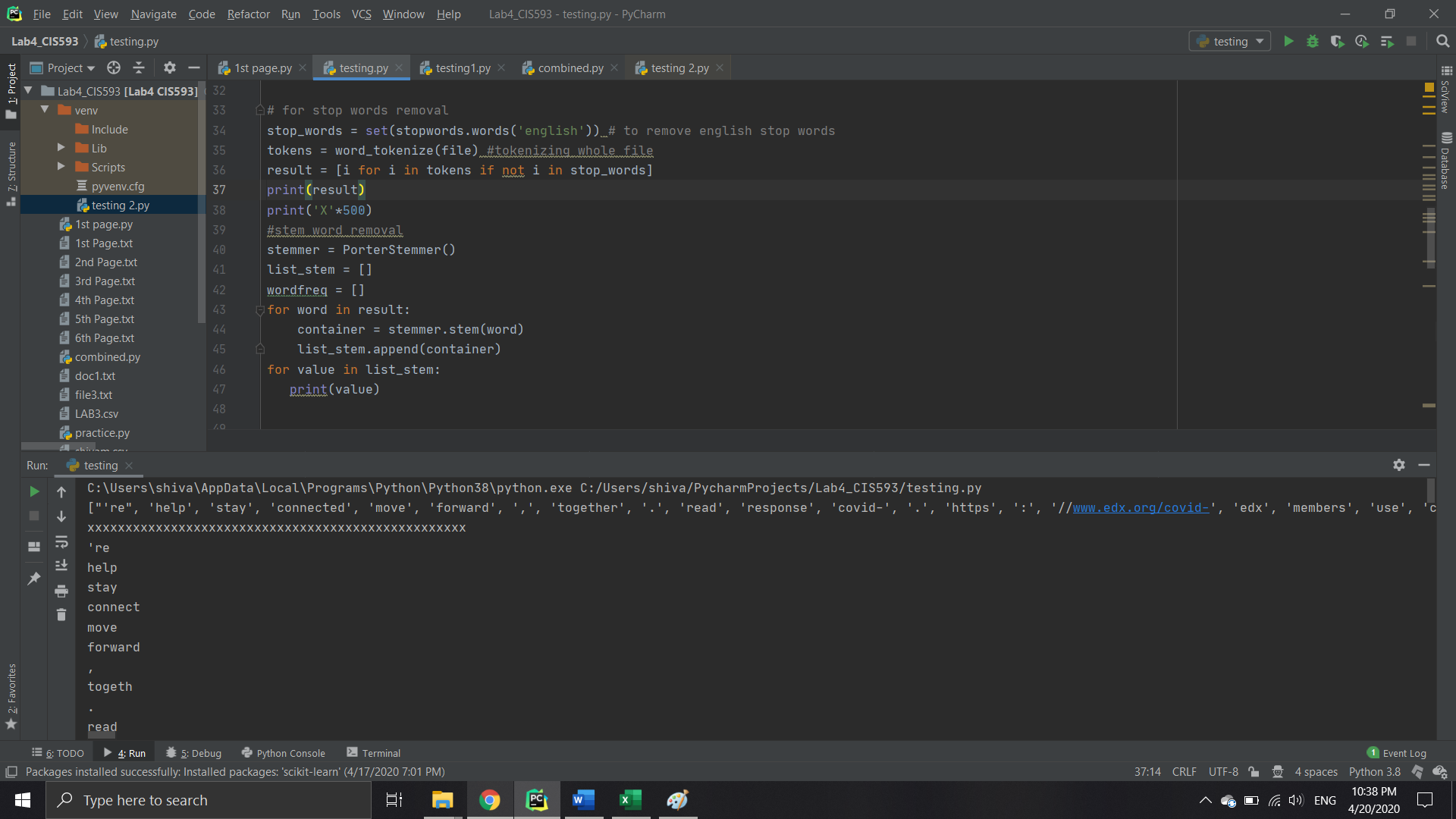
File = file.strip()

To remove all the punctuations

File = re.sub(‘<.\*?>’,”,file)

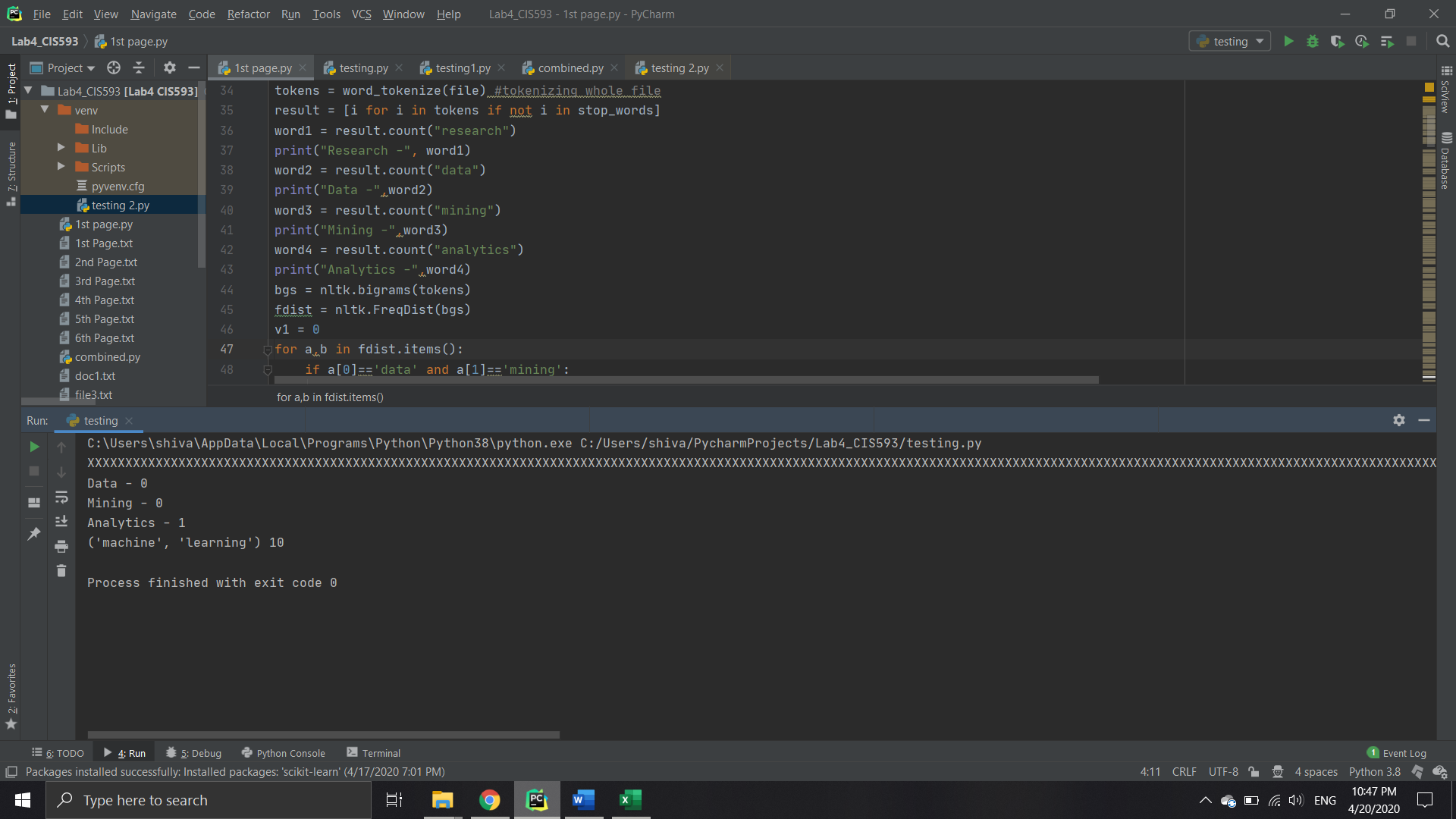


After I have implemented the stop word removal and stem word removal

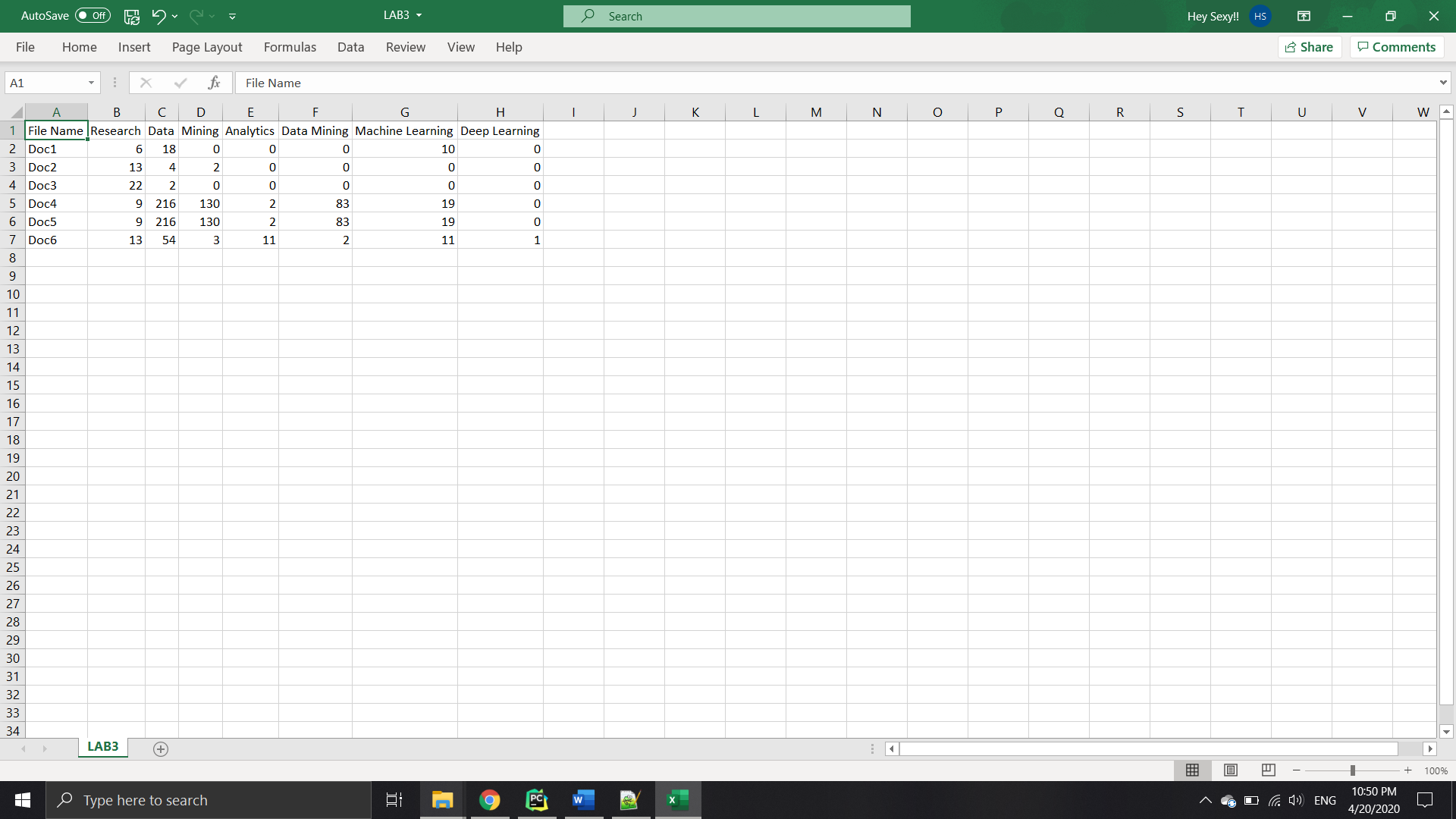


Finally after removing all these unnecessary values

Its time to search for the words which are given in the



CSV FILE



indexFile = open('LAB3.csv', 'w')

fieldNames = ['File Name','Research', 'Data', 'Mining','Analytics', 'Data Mining', 'Machine Learning', 'Deep Learning']

csvWriter = csv.DictWriter(indexFile, fieldnames=fieldNames, lineterminator ='\n')

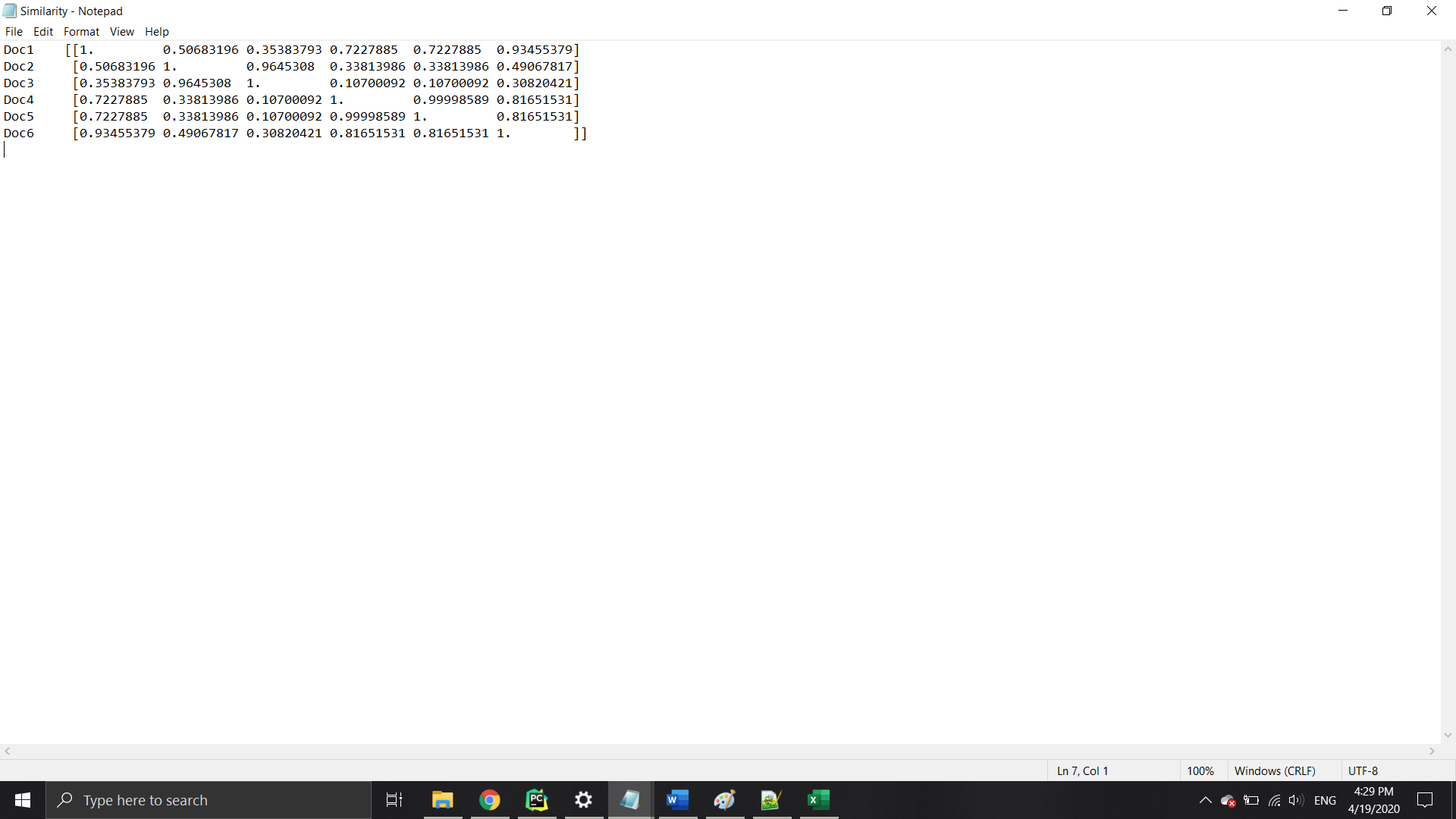
csvWriter.writeheader()

csvWriter.writerow({'File Name': 'Doc1', 'Research': word1, 'Data': word2, 'Mining': word3,

'Analytics': word4, 'Data Mining': v1, 'Machine Learning': vv1, 'Deep Learning': vvv1})

That’s how I stored each word which were given in the assignment in the CSV file.

After this I have applied the cosine similarities on this output, which shows me how document are related to each other.



from sklearn.metrics.pairwise import pairwise\_distances

from sklearn.metrics.pairwise import cosine\_similarity

out = pd.read\_csv('LAB3.csv')

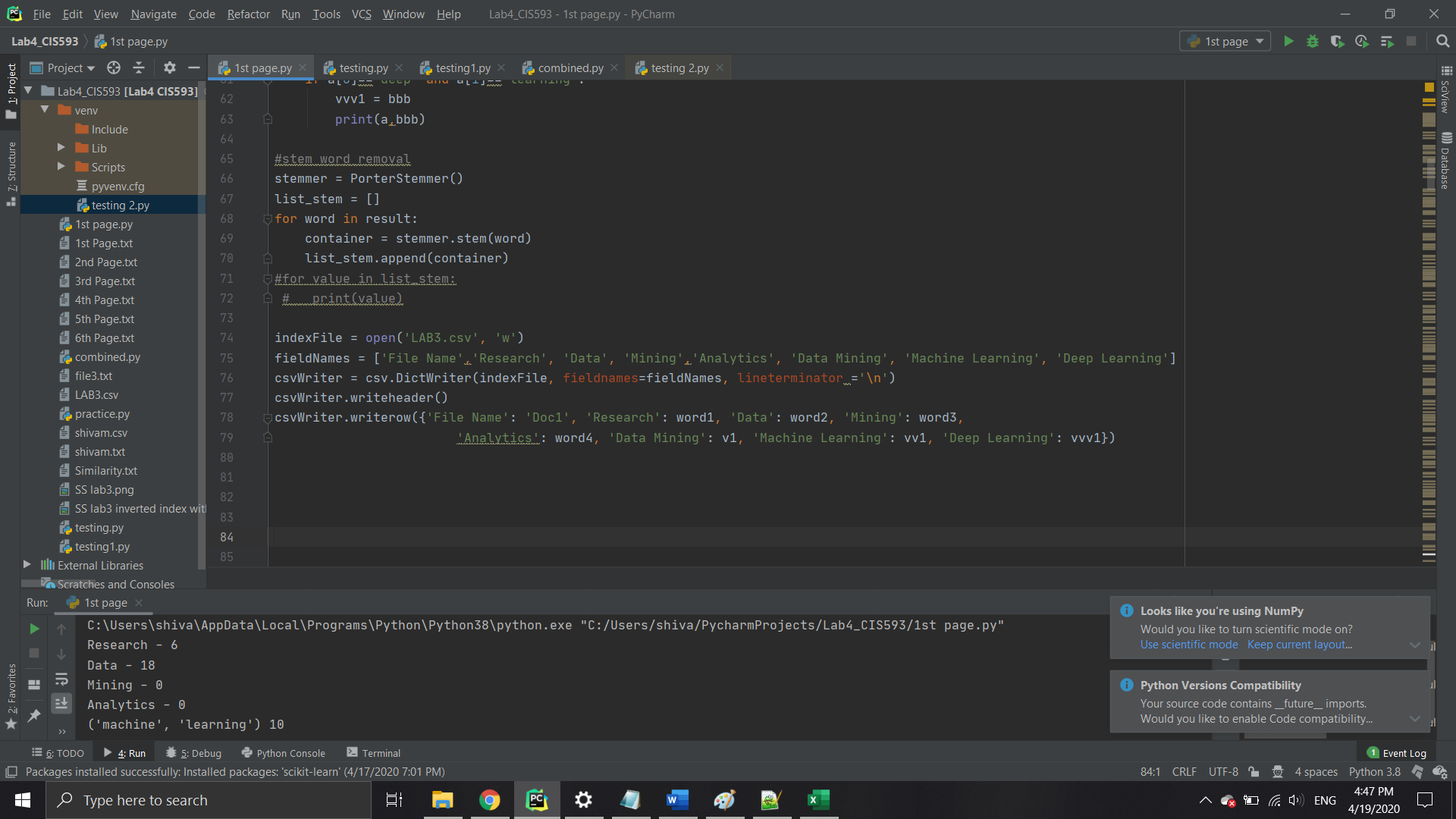
out = pd.get\_dummies(out)

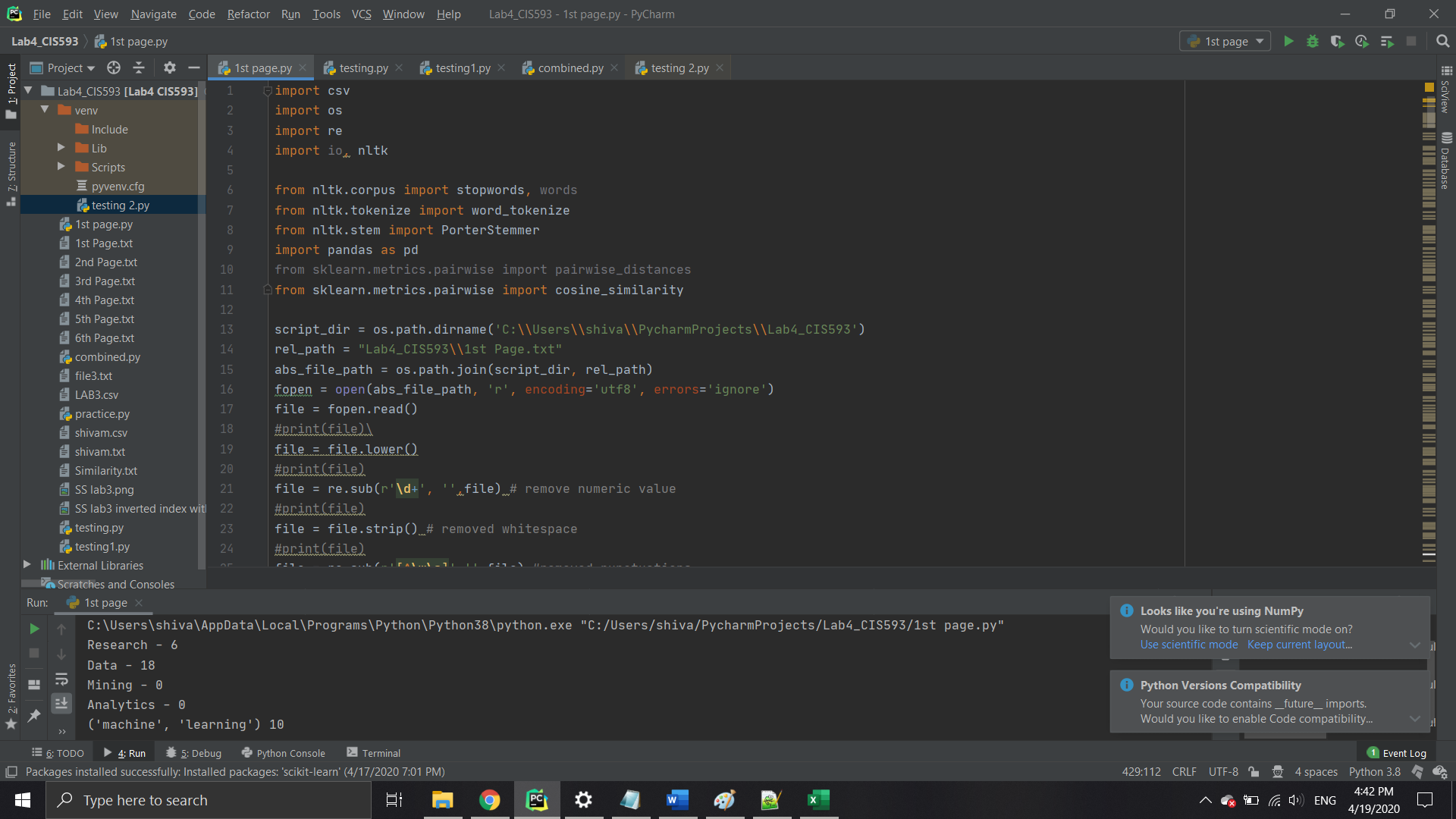
with open('Similarity.txt', 'w') as similarity:

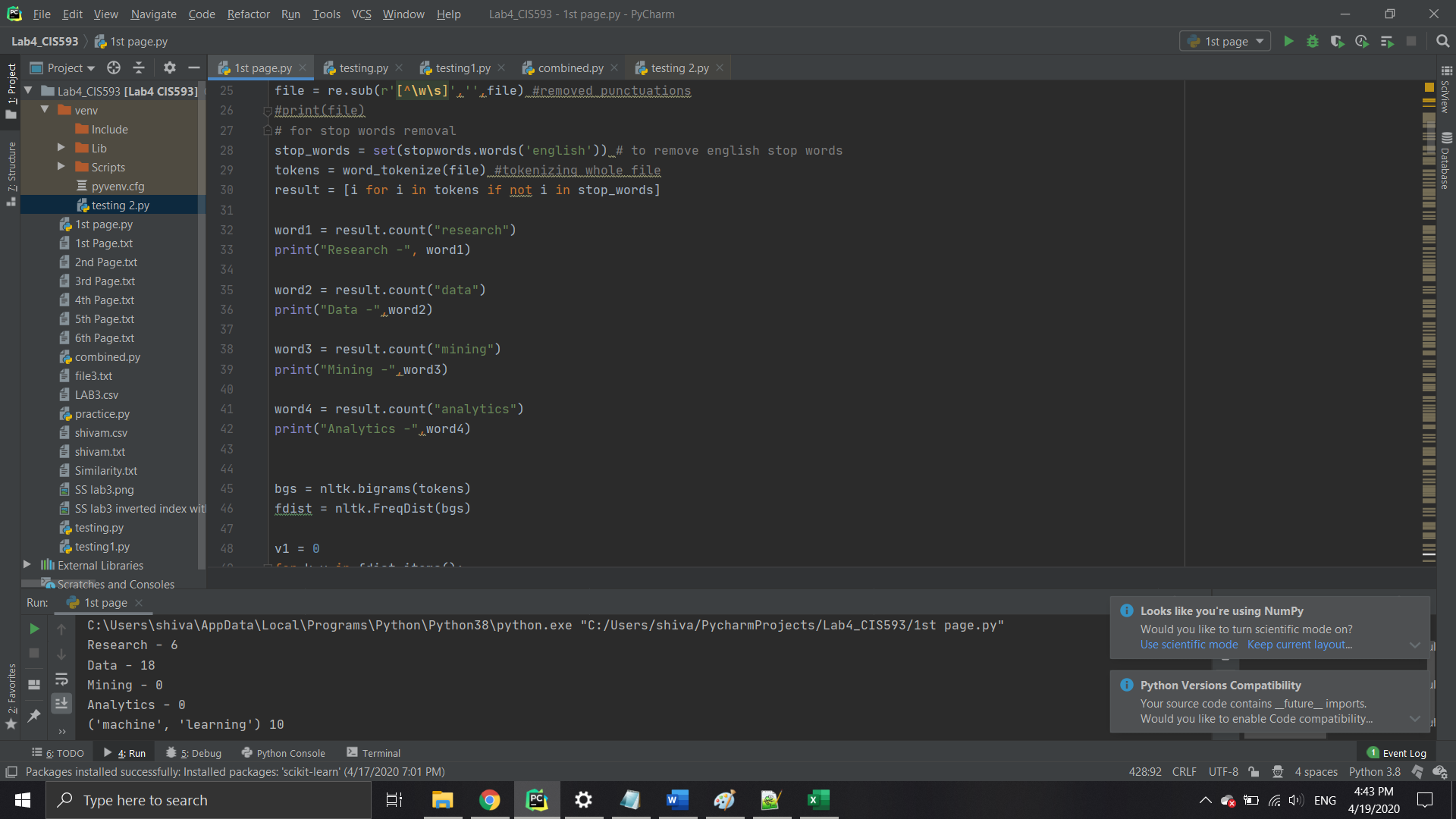
print(cosine\_similarity(out), file=similarity)

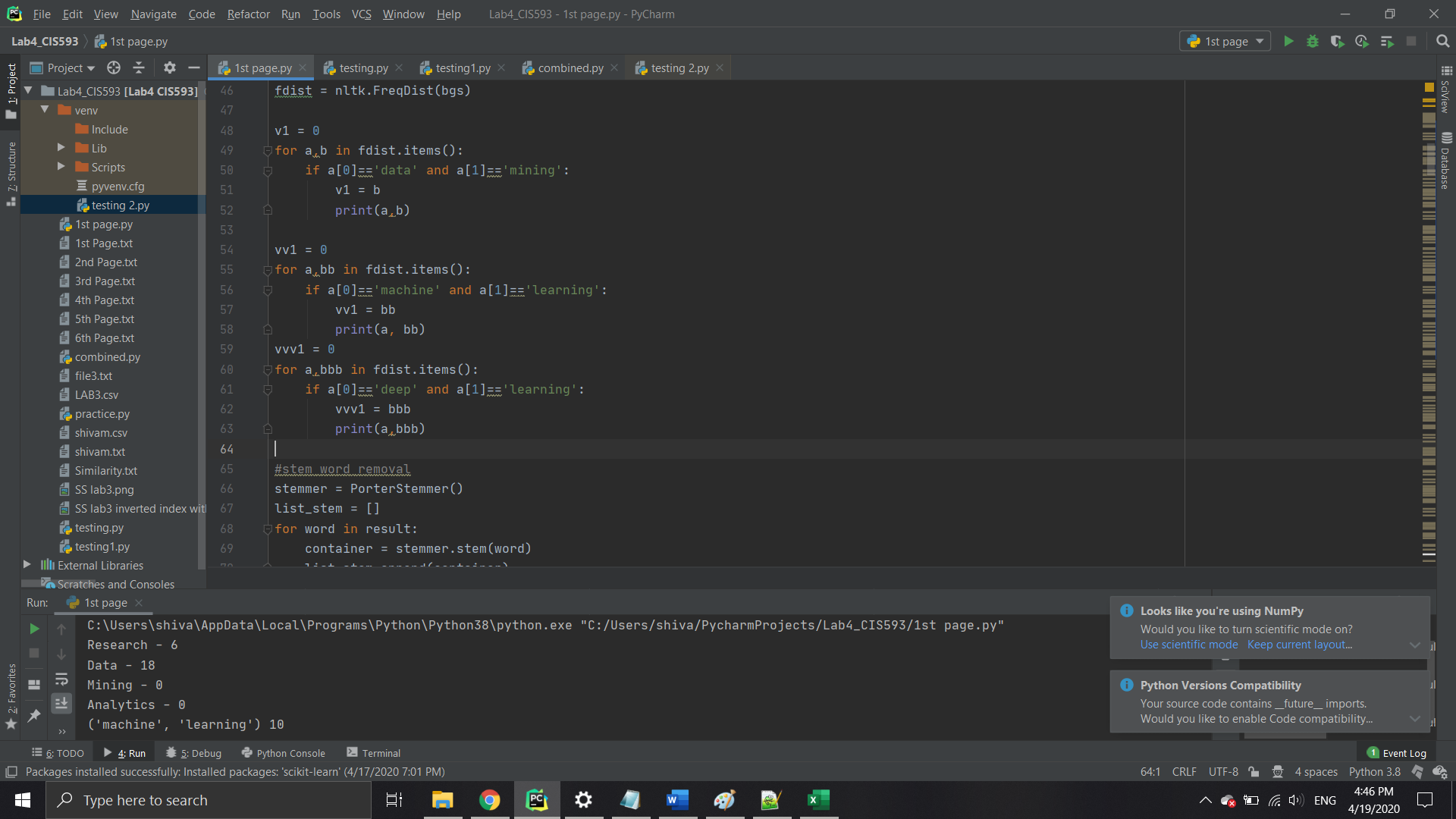
that’s how I got the cosine similarities.

CODE1.1



CODE1.2

CODE1.3

CODE1.4

**Analysis of the results**

1. There were two links which is identical that’s why the word count of those two file was same and the result value was same
2. Document 1 and Document 2 are 50 % identical according to the cosine similarties
3. Document 2 is approximately 7-9% similar to Document 4 and Document 3
4. By looking at the COSINE SIMILARITIES, document 4 and document 5 are similar.

**Tools Used**

1. Pycharm software
2. Python 3.0