Web mining Assignment 2

19BCE2311 Gaurav Singh

Question:

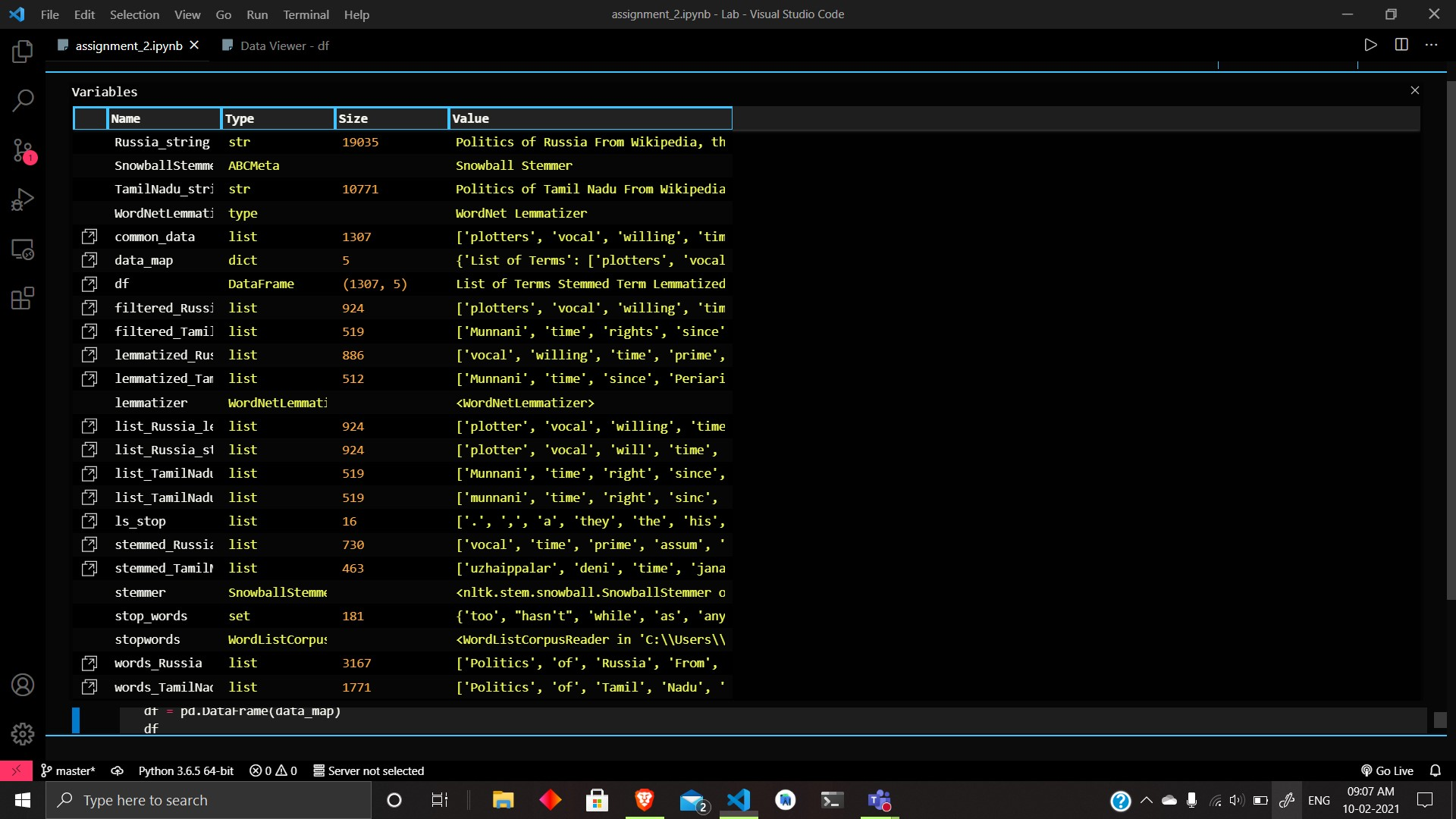
**Practice Programming Exercise:**

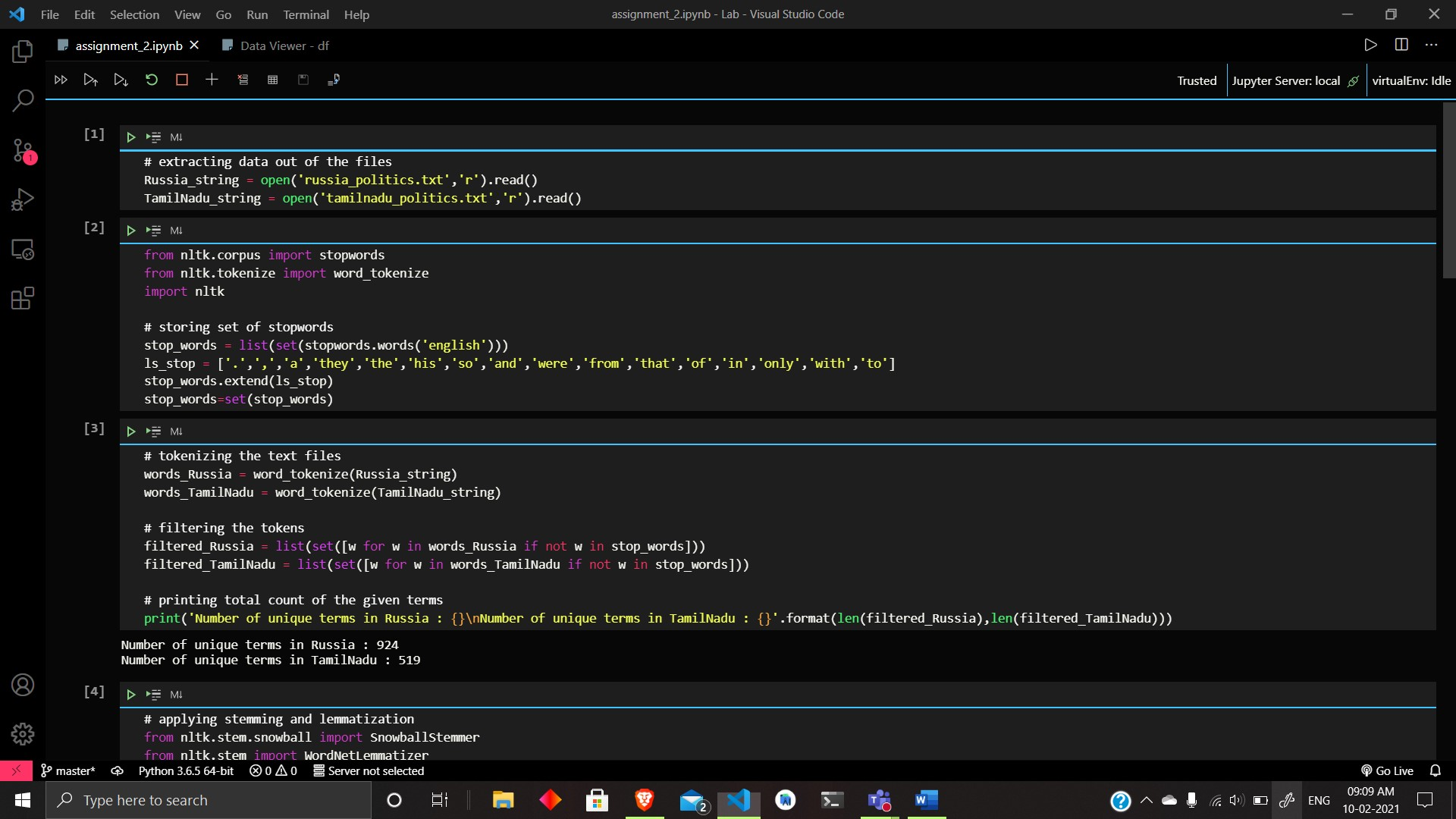
Write a python program to

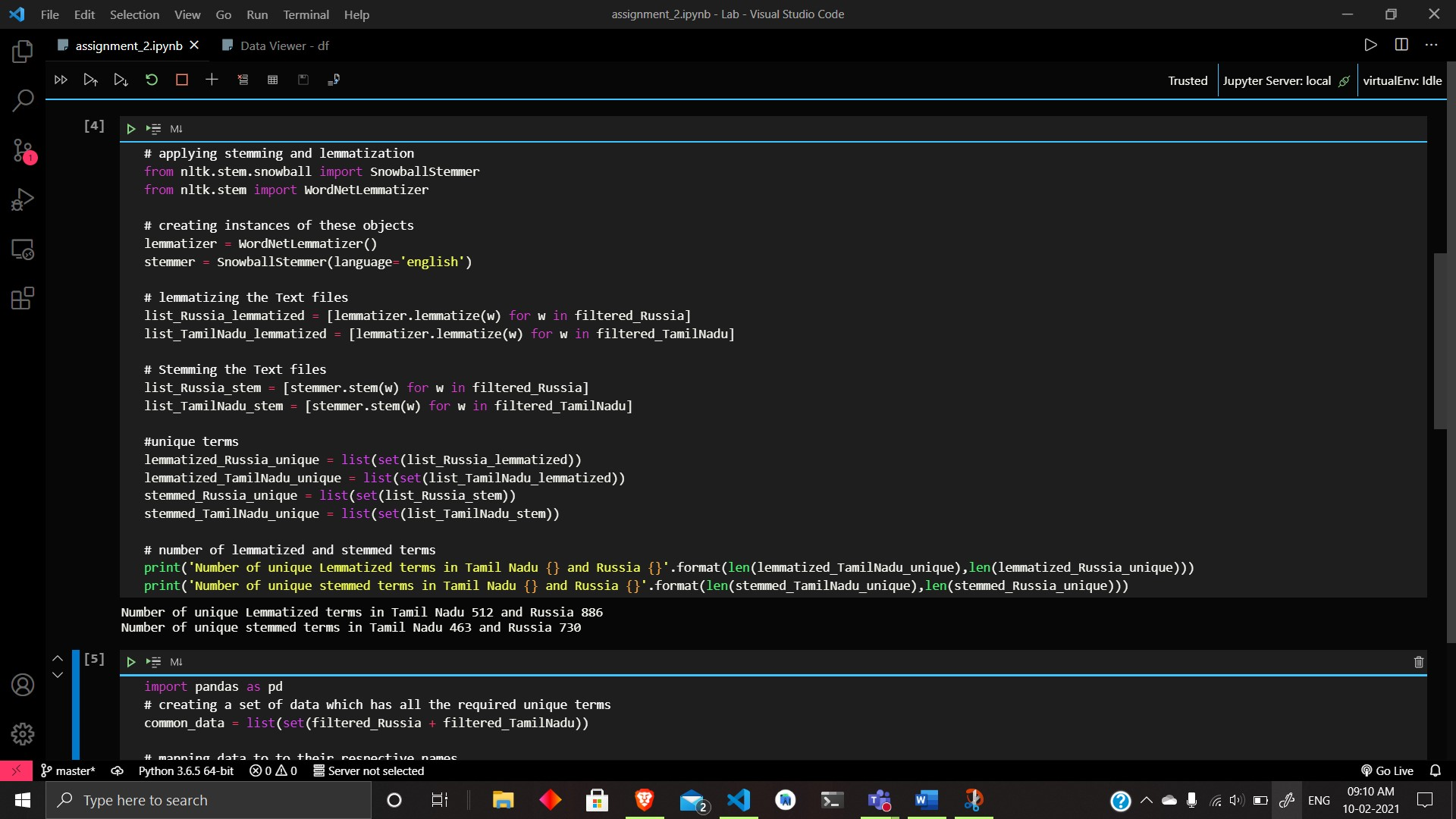
* Extract content from two text files attached
* Count the total number of unique terms in each text file (after removing stop words)
  + [List of additional Stop words to be considered = ['.',',','a','they','the','his','so','and','were','from','that','of','in','only','with','to']]
* Apply Stemming and lemmatization separately on the terms present in both files
* Print their number of unique terms after stemming and lemmatization separately.
* Display the result as Term-Document matrix representation using Pandas (use Bag-of-words model)

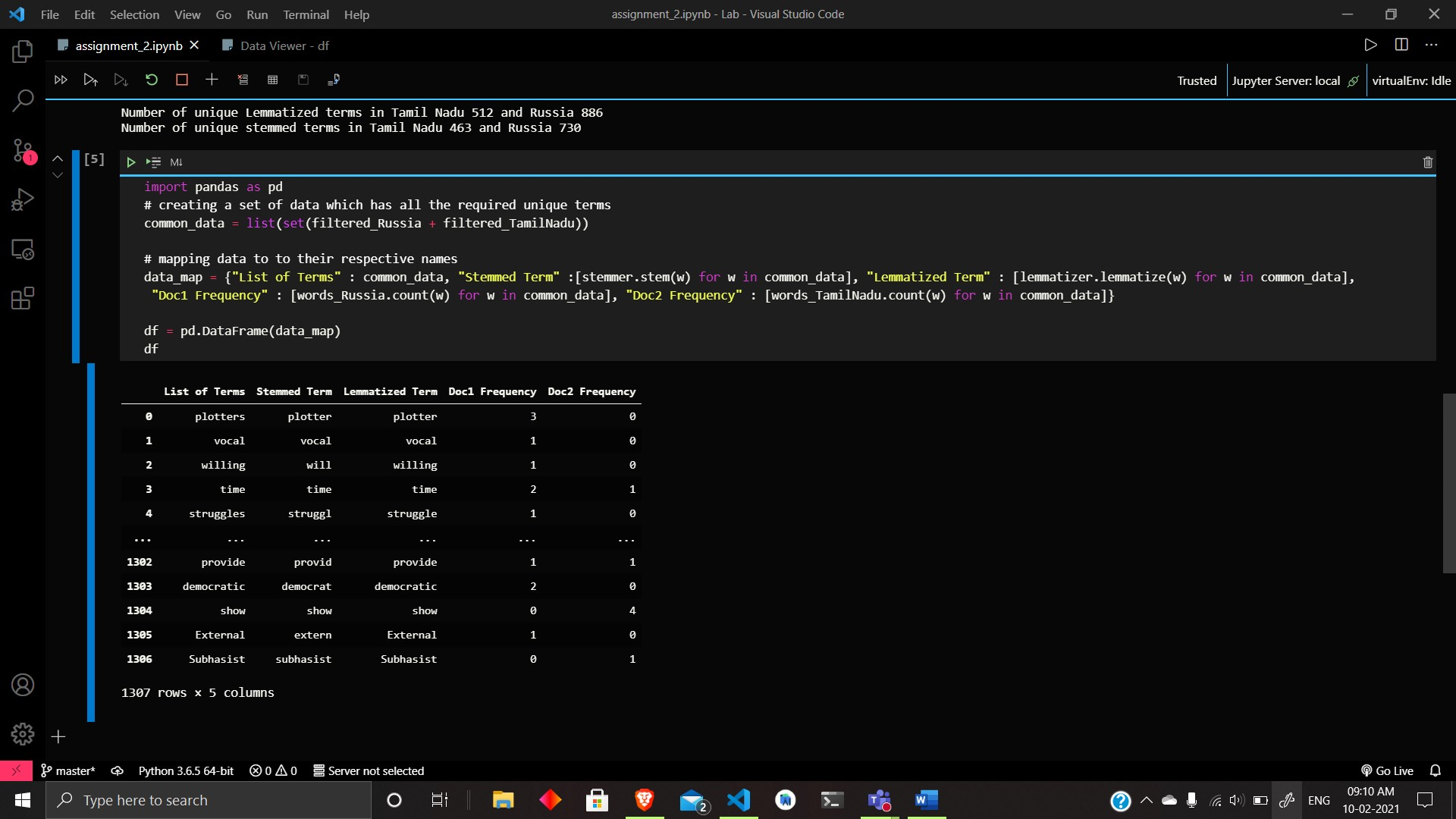
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **List of Terms** | **Stemmed Term** | **Lemmatized Term** | **DOC1 Frequency** | **DOC2 Frequency** |
| Programming | Prog | Program | 10 | 0 |

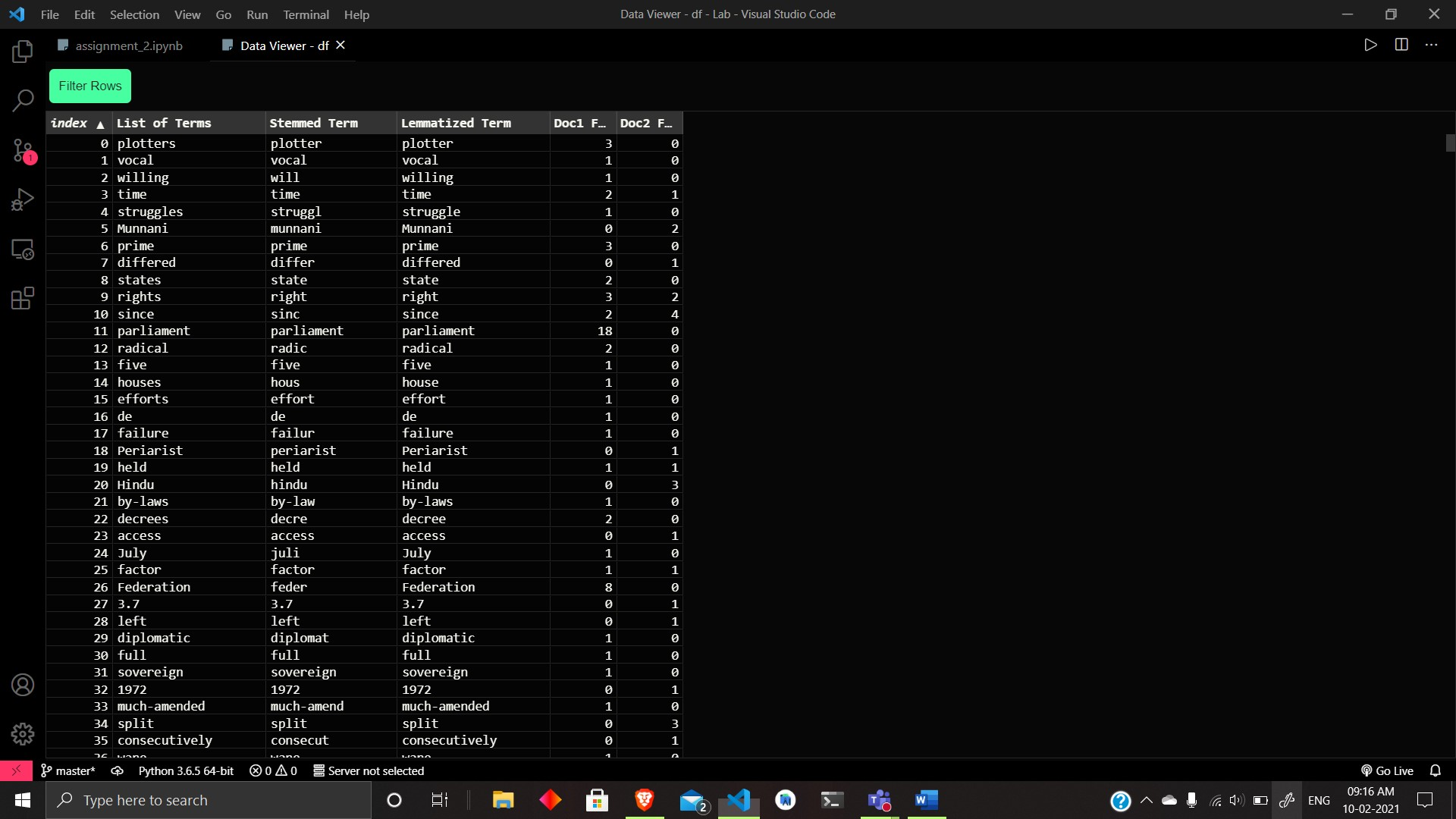
Screenshots











Code:

# extracting data out of the files

Russia\_string = open('russia\_politics.txt','r').read()

TamilNadu\_string = open('tamilnadu\_politics.txt','r').read()

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

import nltk

# storing set of stopwords

stop\_words = list(set(stopwords.words('english')))

ls\_stop = ['.',',','a','they','the','his','so','and','were','from','that','of','in','only','with','to']

stop\_words.extend(ls\_stop)

stop\_words=set(stop\_words)

# tokenizing the text files

words\_Russia = word\_tokenize(Russia\_string)

words\_TamilNadu = word\_tokenize(TamilNadu\_string)

# filtering the tokens

filtered\_Russia = list(set([w for w in words\_Russia if not w in stop\_words]))

filtered\_TamilNadu = list(set([w for w in words\_TamilNadu if not w in stop\_words]))

# printing total count of the given terms

print('Number of unique terms in Russia : {}\nNumber of unique terms in TamilNadu : {}'.format(len(filtered\_Russia),len(filtered\_TamilNadu)))

# applying stemming and lemmatization

from nltk.stem.snowball import SnowballStemmer

from nltk.stem import WordNetLemmatizer

# creating instances of these objects

lemmatizer = WordNetLemmatizer()

stemmer = SnowballStemmer(language='english')

# lemmatizing the Text files

list\_Russia\_lemmatized = [lemmatizer.lemmatize(w) for w in filtered\_Russia]

list\_TamilNadu\_lemmatized = [lemmatizer.lemmatize(w) for w in filtered\_TamilNadu]

# Stemming the Text files

list\_Russia\_stem = [stemmer.stem(w) for w in filtered\_Russia]

list\_TamilNadu\_stem = [stemmer.stem(w) for w in filtered\_TamilNadu]

#unique terms

lemmatized\_Russia\_unique = list(set(list\_Russia\_lemmatized))

lemmatized\_TamilNadu\_unique = list(set(list\_TamilNadu\_lemmatized))

stemmed\_Russia\_unique = list(set(list\_Russia\_stem))

stemmed\_TamilNadu\_unique = list(set(list\_TamilNadu\_stem))

# number of lemmatized and stemmed terms

print('Number of unique Lemmatized terms in Tamil Nadu {} and Russia {}'.format(len(lemmatized\_TamilNadu\_unique),len(lemmatized\_Russia\_unique)))

print('Number of unique stemmed terms in Tamil Nadu {} and Russia {}'.format(len(stemmed\_TamilNadu\_unique),len(stemmed\_Russia\_unique)))

import pandas as pd

# creating a set of data which has all the required unique terms

common\_data = list(set(filtered\_Russia + filtered\_TamilNadu))

# mapping data to to their respective names

data\_map = {"List of Terms" : common\_data, "Stemmed Term" :[stemmer.stem(w) for w in common\_data], "Lemmatized Term" : [lemmatizer.lemmatize(w) for w in common\_data], "Doc1 Frequency" : [words\_Russia.count(w) for w in common\_data], "Doc2 Frequency" : [words\_TamilNadu.count(w) for w in common\_data]}

df = pd.DataFrame(data\_map)

df