**MACHINE LEARNING LAB ASSIGNMENT**

**[DS7A-709]**

WEEK-1

**MASTER OF TECHNOLOGY**

In

**DATA SCIENCE**

***Submitted by***

**RAHUL SHARMA**

***Submitted To***

**DR. DINESH BHATI**

**(Lecturer)**

**SCHOOL OF DATA SCIENCE AND FORECASTING**

(UNIVERSITY TEACHING DEPARTMENT)

**DEVI AHILYA VISHWAVIDYALAYA**

**Indore (M.P.)**

# Milestone 1: Downloading NLP libraries (NLTK)

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

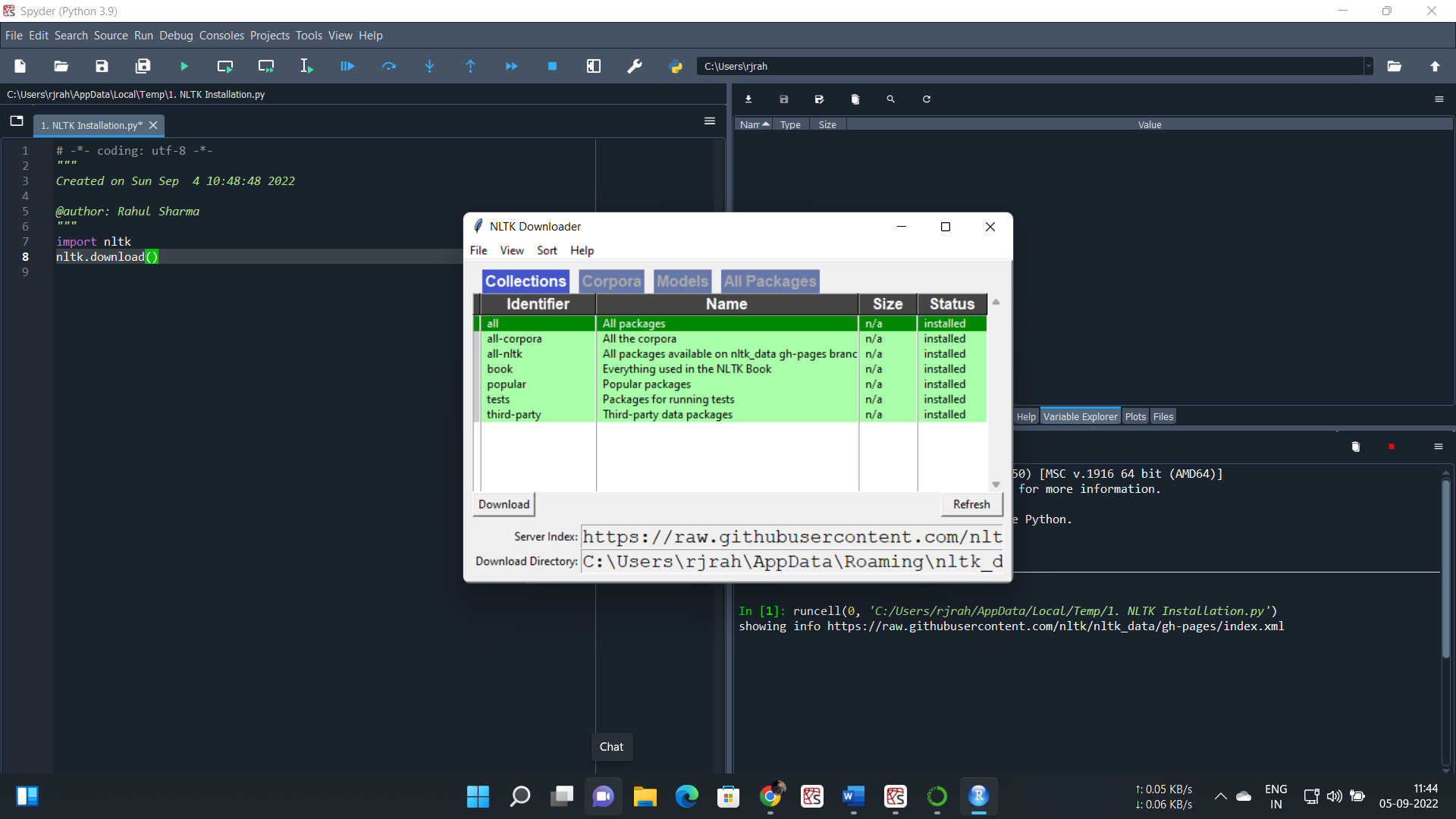
## Interactive installer

Run the Python interpreter and type the commands:

import nltk

nltk.download()

A new window opened, showing the NLTK Downloader. Clicked on the File menu and select Change Download Directory where we want to download.



Installing the whole package. Once the package is installed.

Firstly, imported the corpus:

from nltk.corpus import names

print (names.words()[:10])

*Output:*

['Abagael', 'Abagail', 'Abbe', 'Abbey', 'Abbi', 'Abbie', 'Abby', 'Abigael', 'Abigail', 'Abigale']#

*Code#*

print(len(names.words()))

*Output:*

7944

There are in total 7944 names

# Stemming

Stemming is a process of reverting an inflected or derived word to its root form. Stemmers remove morphological affixes from words, leaving only the word stem.

## PorterStemmer Interface

The Porter stemming algorithm is a process for removing the commoner morphological and inflexional endings from words in English.

*Code#*

from nltk.stem.porter import PorterStemmer

ps = PorterStemmer()

Test the stemmer on various pluralised words.

words=['machines', 'learning','stating', 'owned', 'agreed','reference']

for w in words:

print(w, " : ", ps.stem(w))

*Output:*

machines : machin

learning : learn

stating : state

owned : own

agreed : agre

reference : refer

Noted that, stemming sometimes involves chopping off letters, if necessary, as we can see in machin.

Now import a lemmatization algorithm based on Wordnet corpus built-in, and initialize a lemmatizer.

## WordNet Interface

WordNet is just another NLTK corpus reader, and can be imported like this:

*Code#*

from nltk.stem import WordNetLemmatizer

wnl = WordNetLemmatizer()

print("machines :", wnl.lemmatize("machines"))

print("learning :", wnl.lemmatize("learning"))

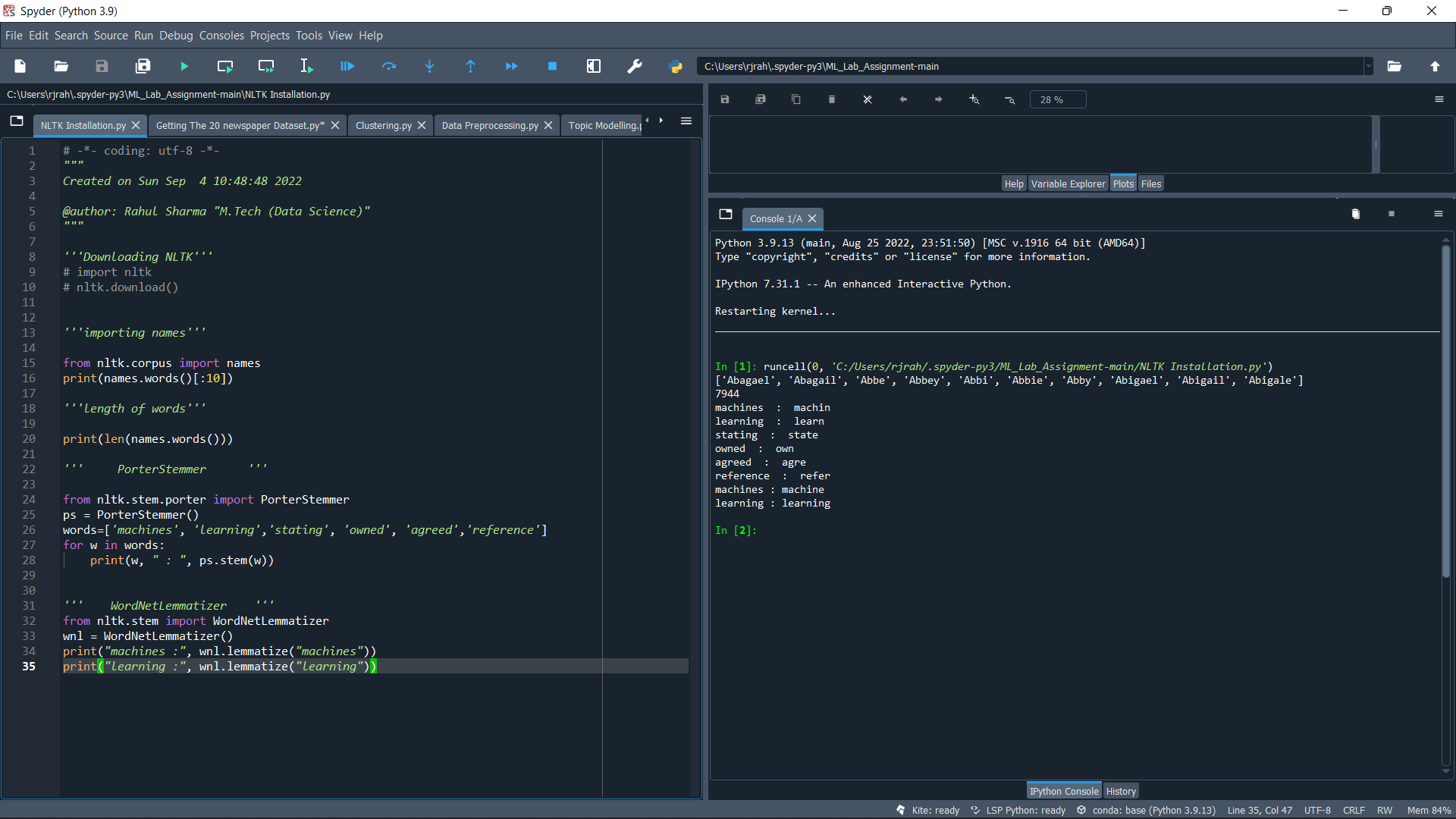
*Output:*

machines : machine

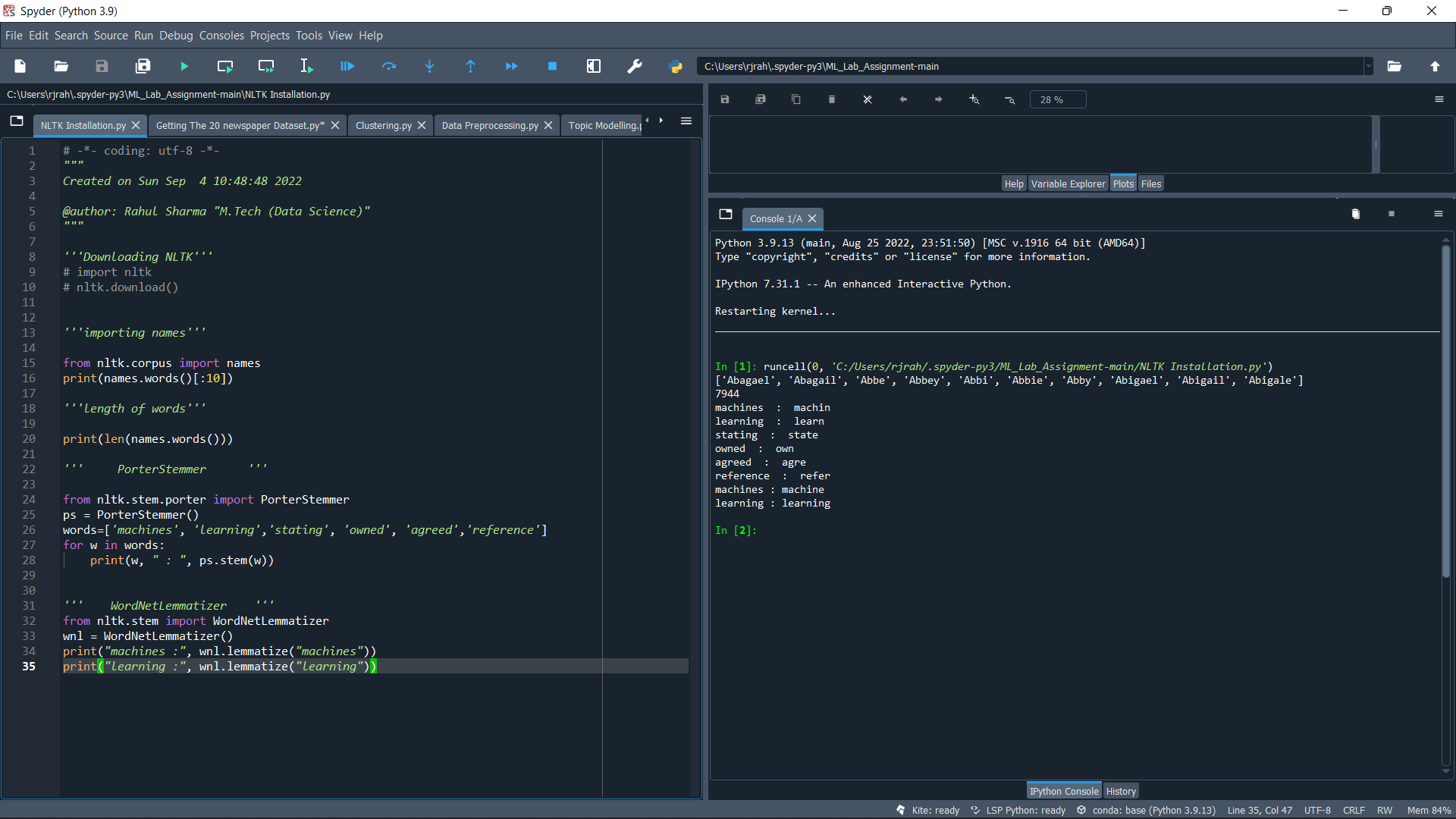
learning : learning

It turns out that this algorithm only lemmatizes on nouns by default.

*Source Code#*

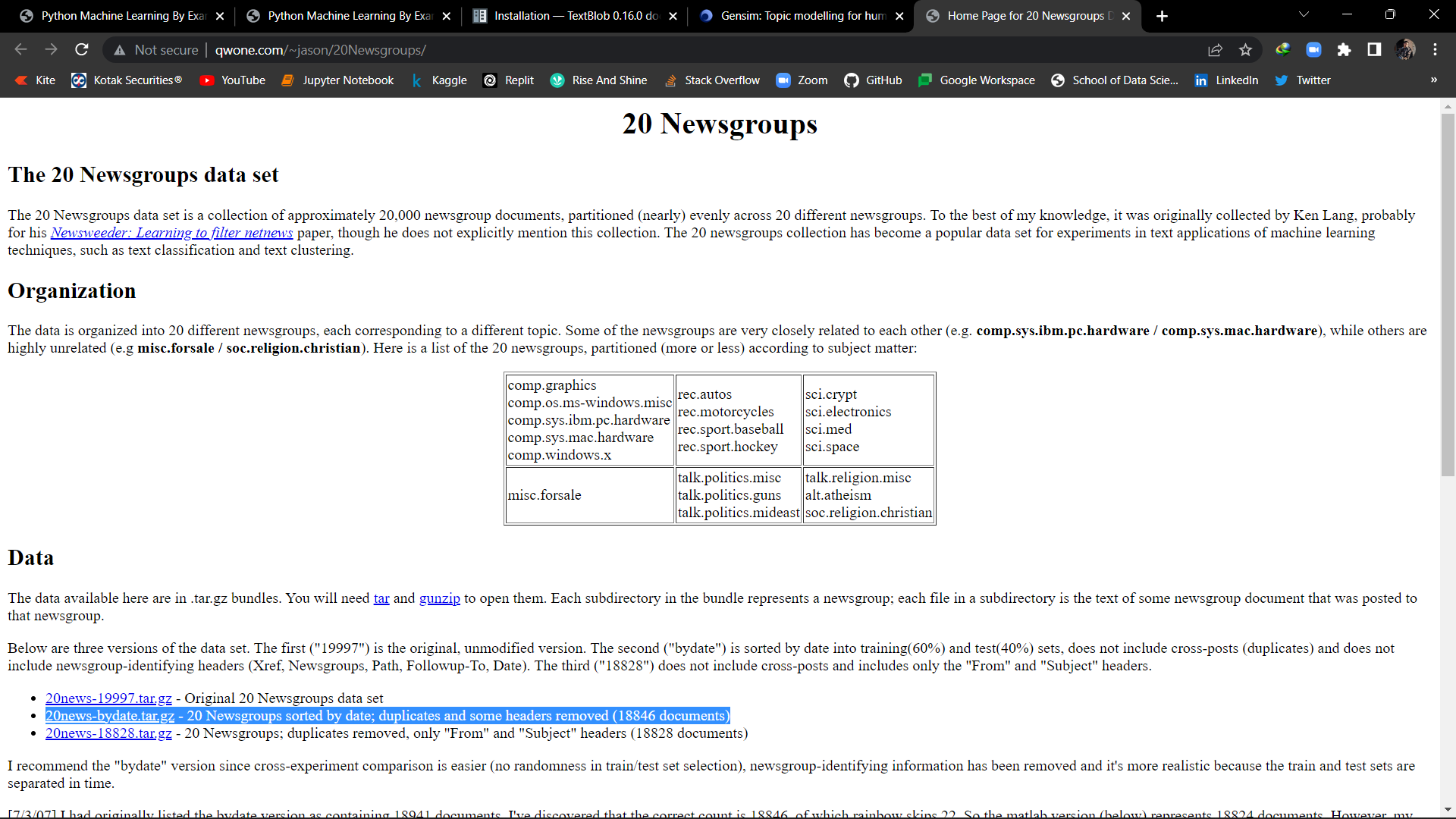


*Output:*



# Milestone 2: The 20 Newsgroups data set

The 20 newsgroups dataset comprises around 18000 newsgroups posts on 20 topics split into two subsets: one for training (or development) and the other one for testing (or for performance evaluation). The split between the train and test set is based on a message posted before and after a specific date.



The data available here are in .tar.gz bundles. You will need [tar](http://www.gnu.org/software/tar/tar.html) and [gunzip](http://www.gnu.org/software/gzip/gzip.html) to open them [20news-bydate.tar.gz](http://qwone.com/~jason/20Newsgroups/20news-bydate.tar.gz) - 20 Newsgroups sorted by date; duplicates and some headers removed (18846 documents)

# Milestone 3: Getting the data

To load the data, we can import the loader function for the 20 newsgroups data as follows:

from sklearn.datasets import fetch\_20newsgroups

groups = fetch\_20newsgroups()

# Milestone 4: Thinking about features

*Code#*

print(groups.keys())

print(groups['target\_names'])

print(groups.target)

*Output:*

dict\_keys(['data', 'filenames', 'target\_names', 'target', 'DESCR'])

['alt.atheism', 'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware', 'comp.windows.x', 'misc.forsale', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey', 'sci.crypt', 'sci.electronics', 'sci.med', 'sci.space', 'soc.religion.christian', 'talk.politics.guns', 'talk.politics.mideast', 'talk.politics.misc', 'talk.religion.misc']

[7 4 4 ... 3 1 8]

*Code#*

import numpy as np

print(np.unique(groups.target))

print(groups.data[0])

*Output:*

[ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19]

From: lerxst@wam.umd.edu (where's my thing)

Subject: WHAT car is this!?

Nntp-Posting-Host: rac3.wam.umd.edu

Organization: University of Maryland, College Park

Lines: 15

I was wondering if anyone out there could enlighten me on this car I saw

the other day. It was a 2-door sports car, looked to be from the late 60s/

early 70s. It was called a Bricklin. The doors were really small. In addition,

the front bumper was separate from the rest of the body. This is

all I know. If anyone can tellme a model name, engine specs, years

of production, where this car is made, history, or whatever info you

have on this funky looking car, please e-mail.

Thanks,

- IL

---- brought to you by your neighborhood Lerxst ----

*Code#*

print(groups.target[0])

print(groups.target\_names[groups.target[0]])

print(len(groups.data[0]))

print(len(groups.data[1]))

Output:

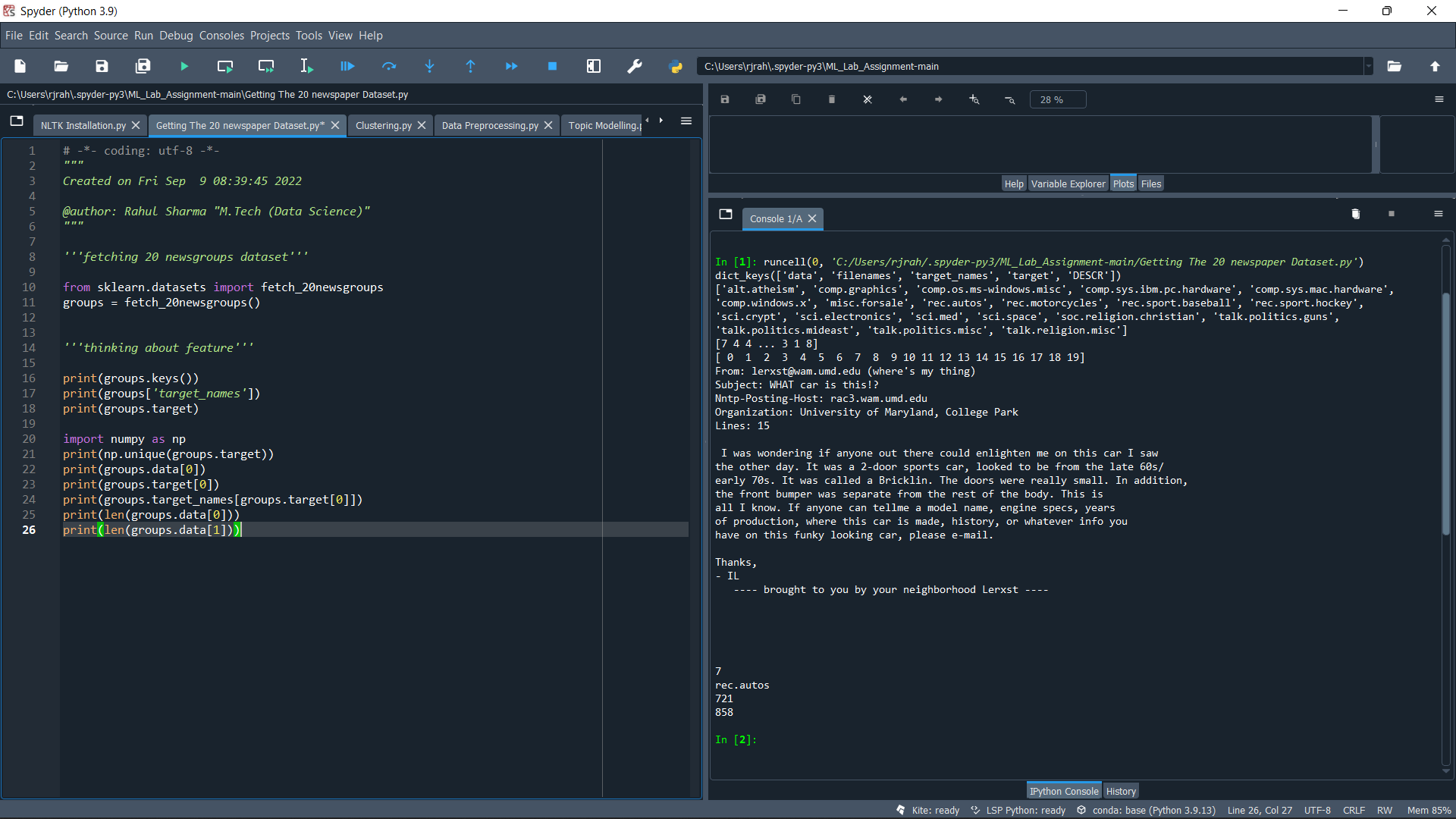
7

rec.autos

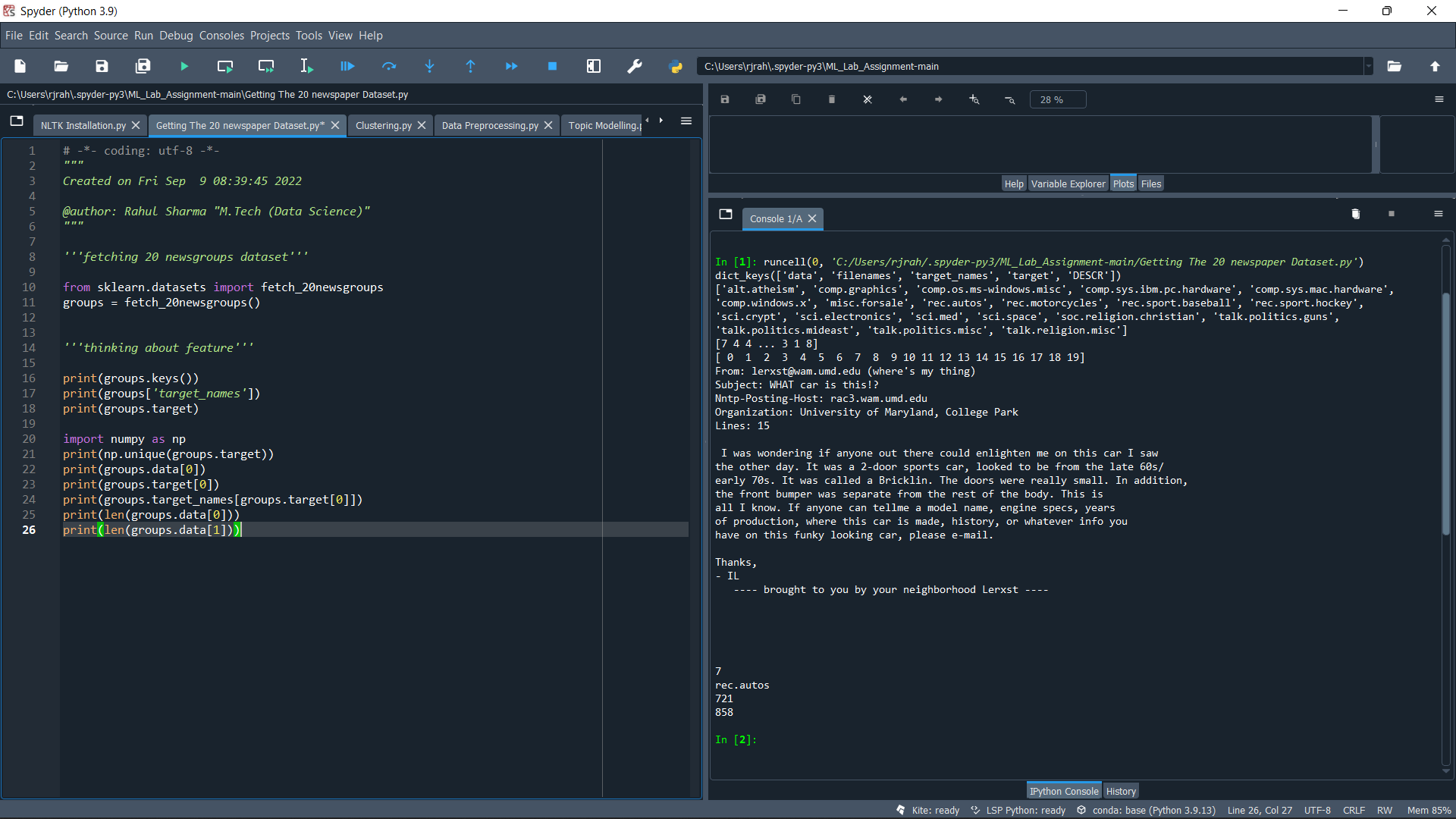
721

858

*Source Code#*



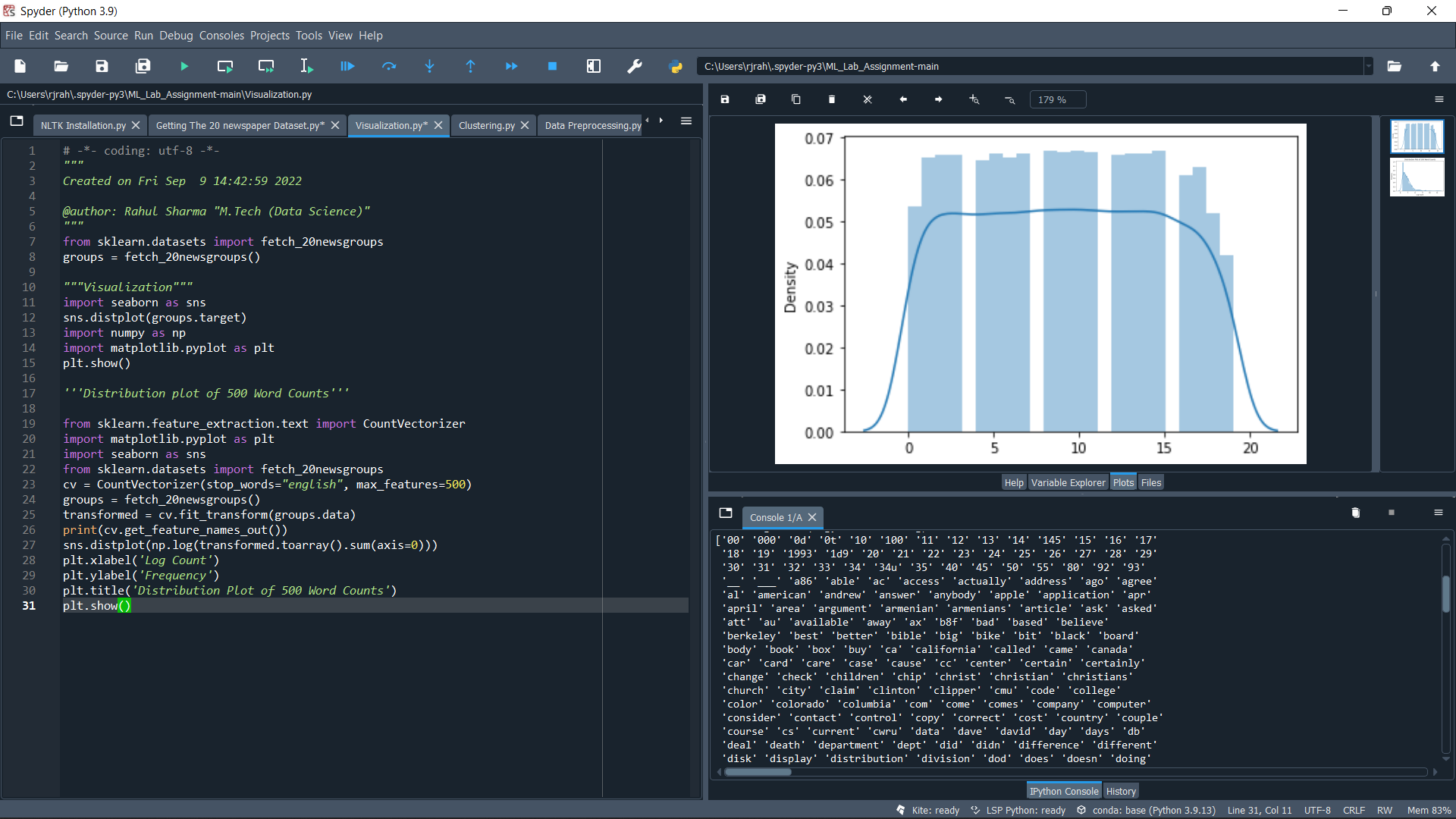
*Output:*



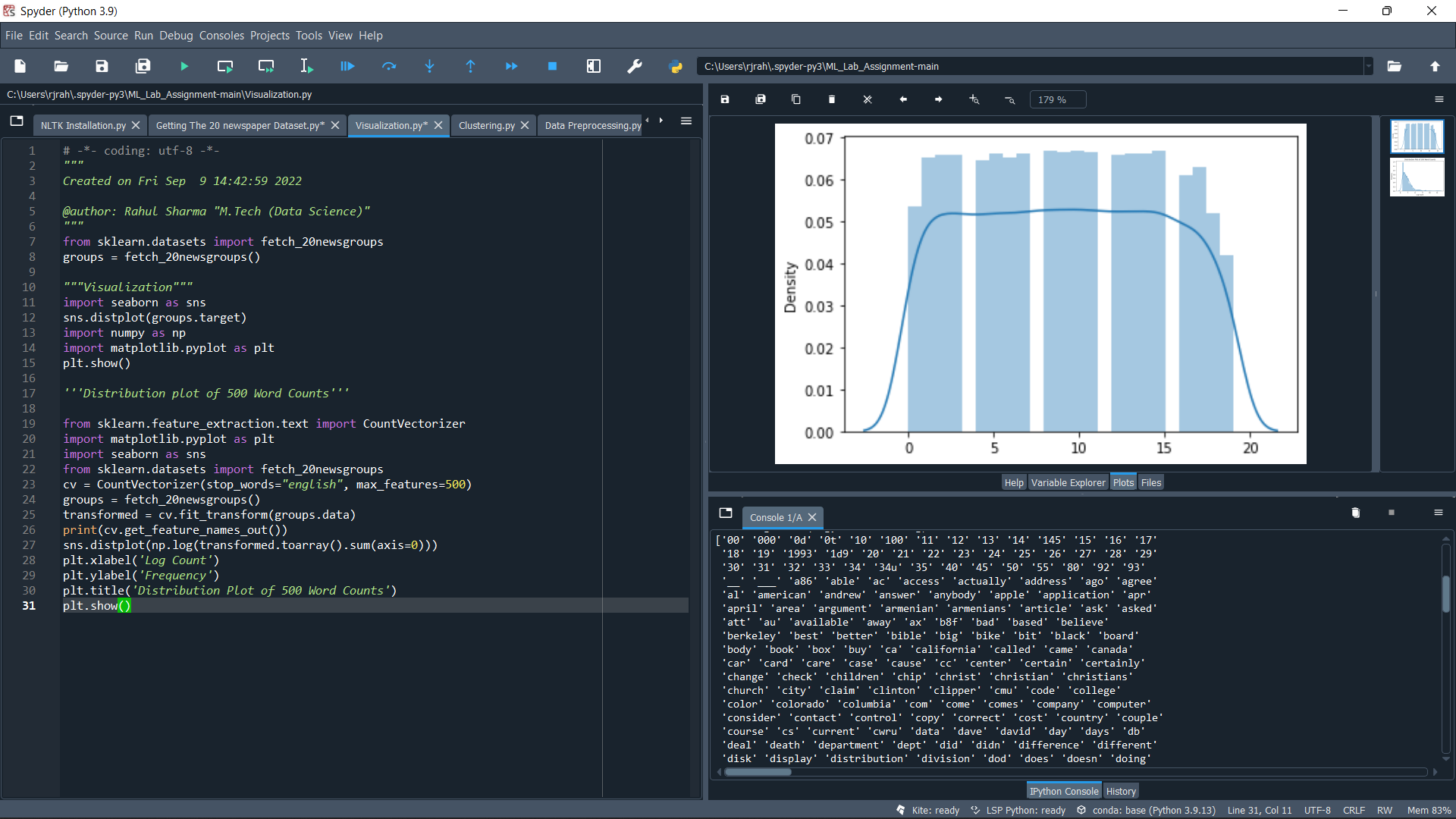
# Milestone 5: Visualization

It's good to visualize to get a general idea of how the data is structured, what possible issues may arise, and if there are any irregularities that we have to take care of:

*Source Code#*



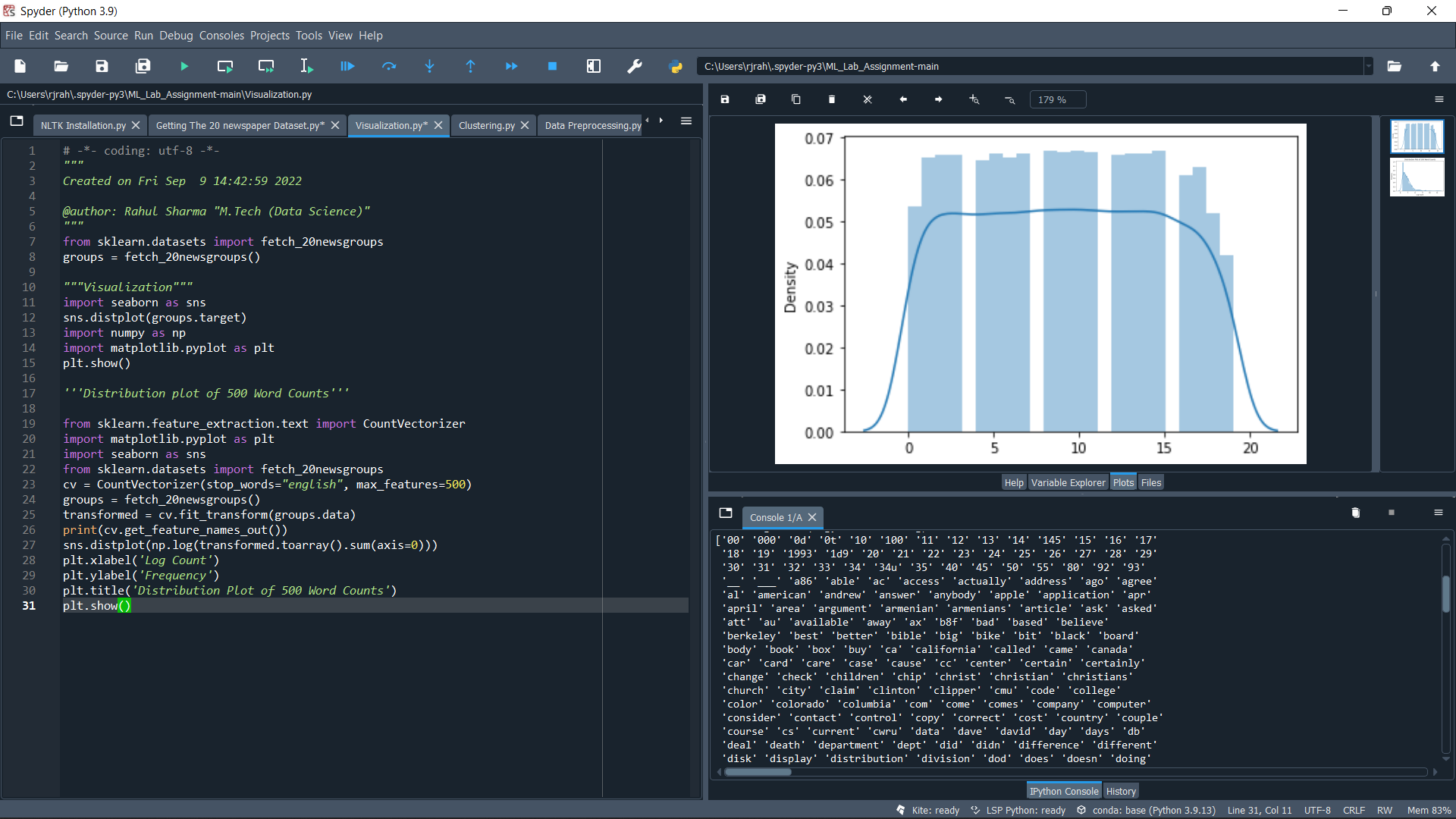
*Output:*



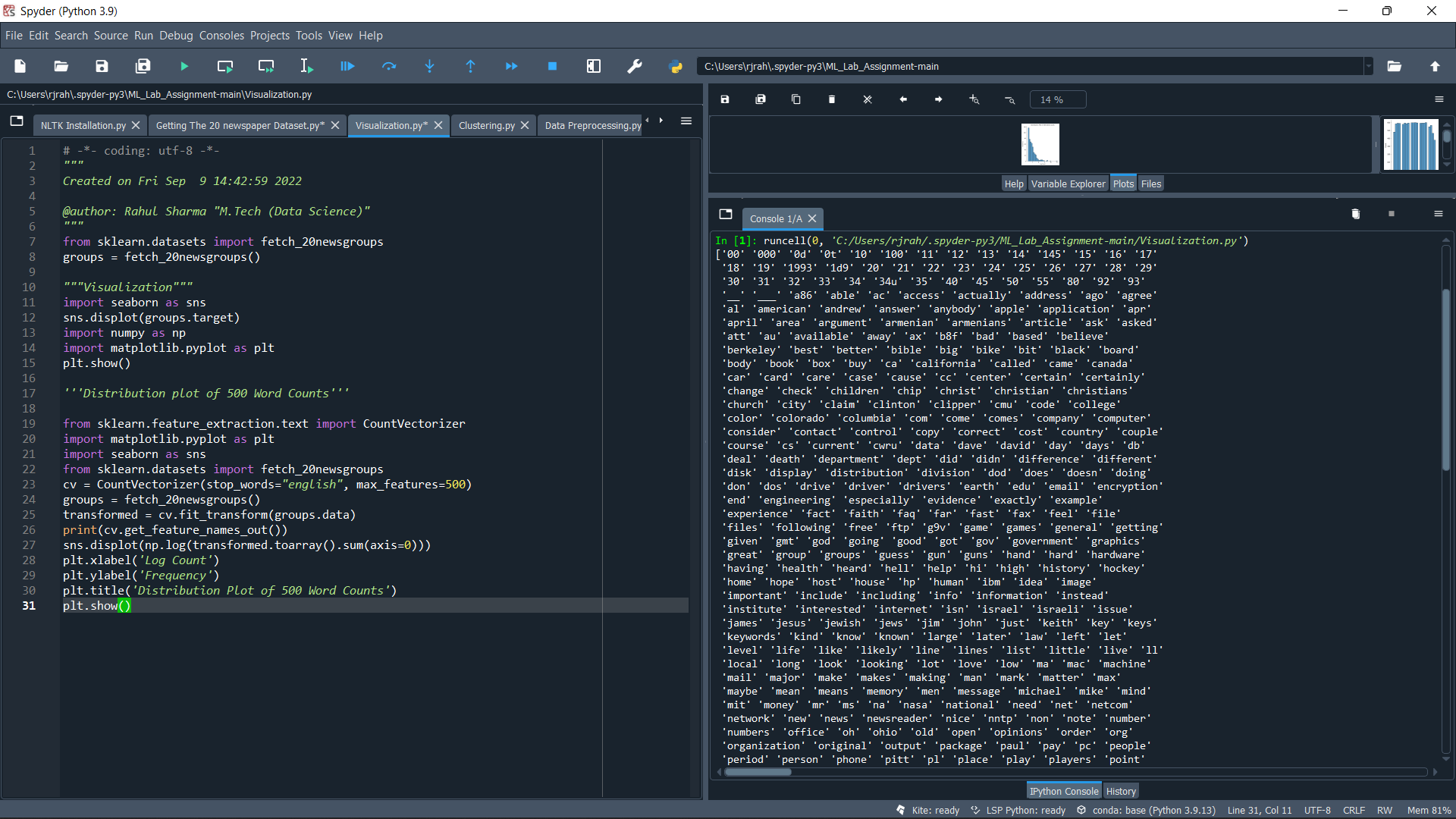
We can see, that the distribution is (approximately) uniform.

The following code displays a histogram of the 500 highest word counts:

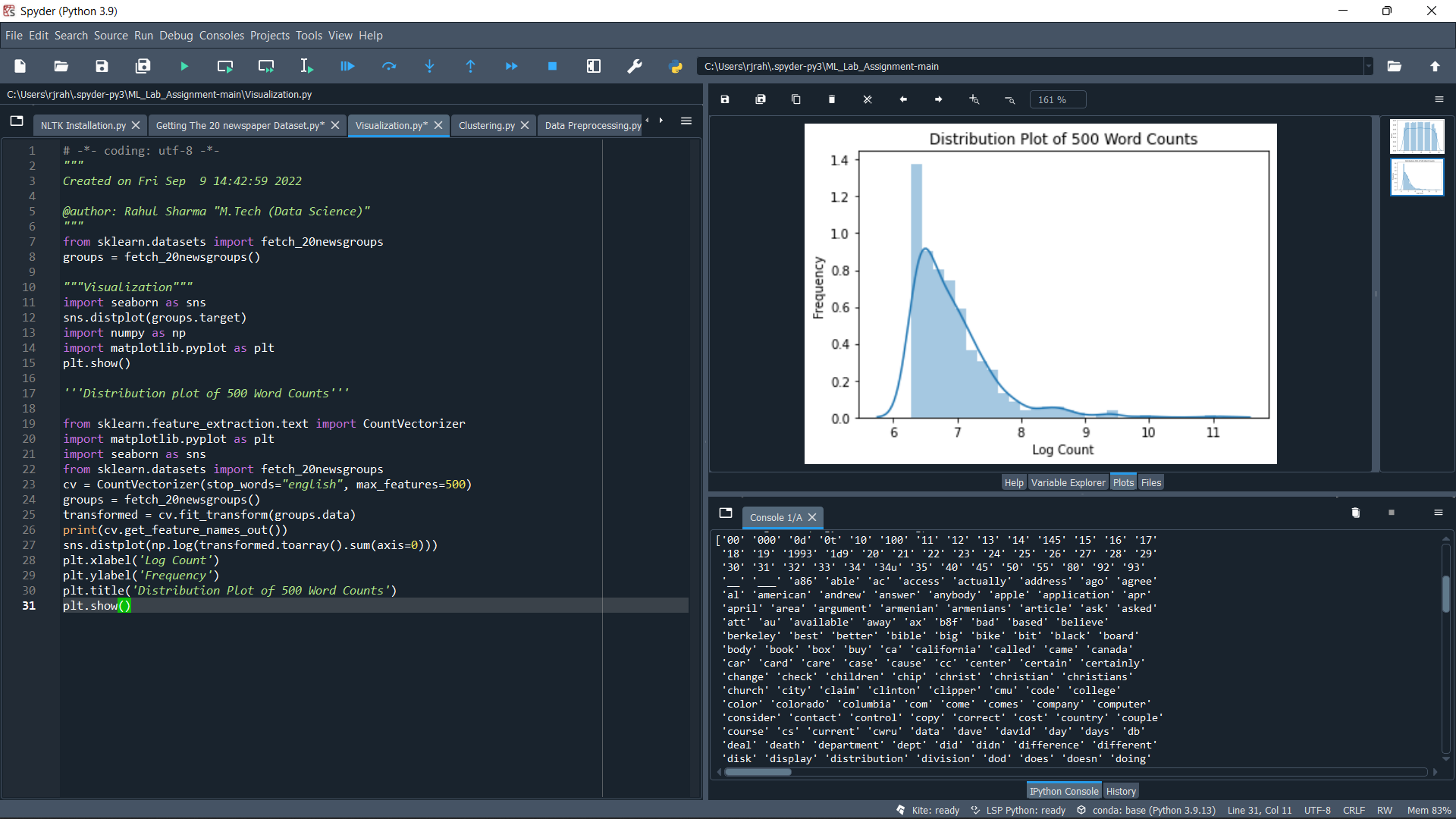
*Source code#*



*Output: list of 500 words*



*Distribution Plot of 500 Word Counts*



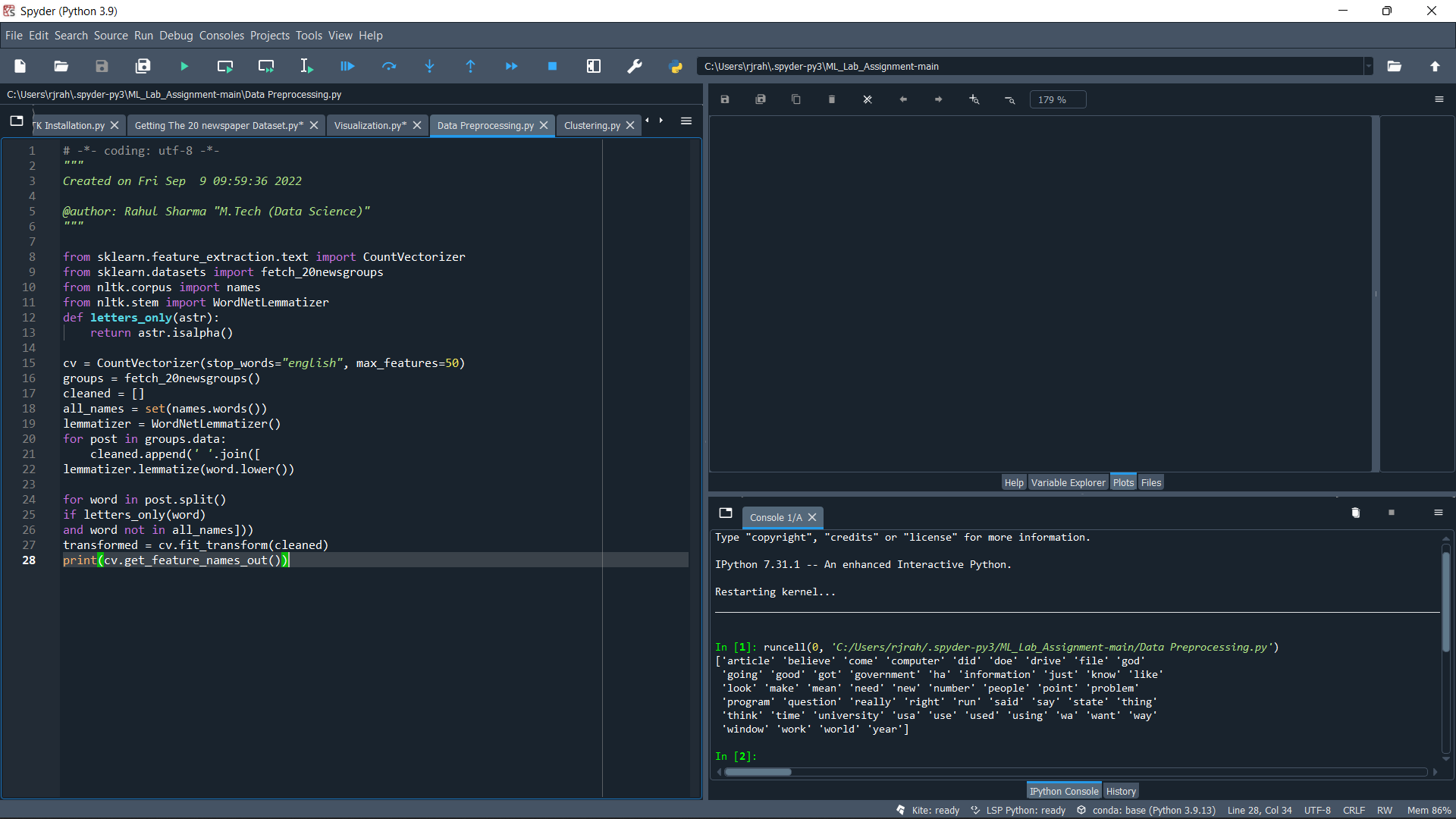
# Milestone 6: Data Pre-processing

We see items, which are not words, such as 00 and 000. Maybe we should ignore items that contain only digits. However, 0d and 0t are also not words. We also see items as \_\_, so maybe we should only allow items that consist only of letters.

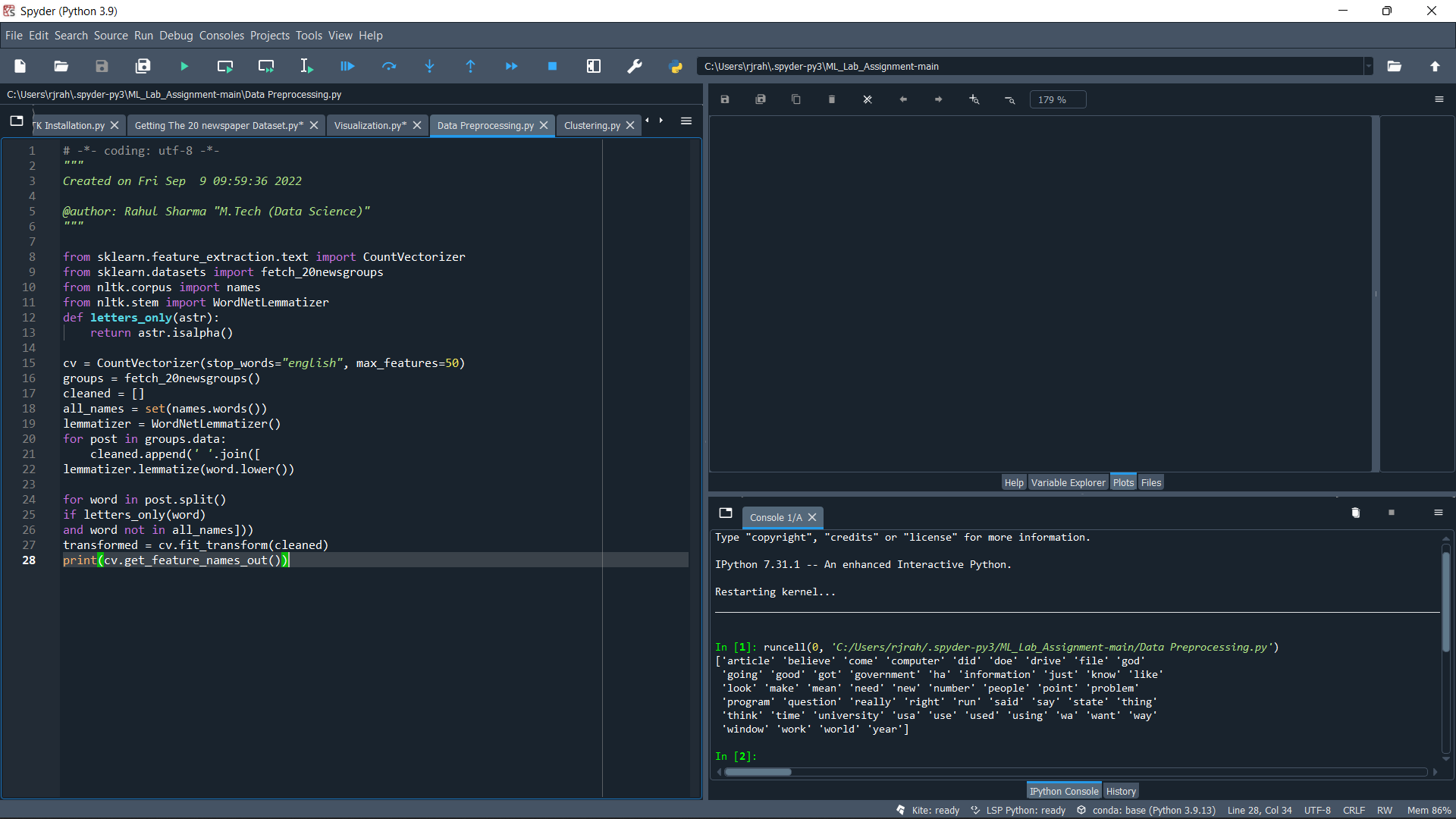
The posts contain names such as andrew as well. We can filter names with the Names corpus from NLTK.

With every filtering we apply, we have to make sure that we don't lose information. Finally, we see words that are very similar, such as include and including, and make and makes.

*Source Code#*



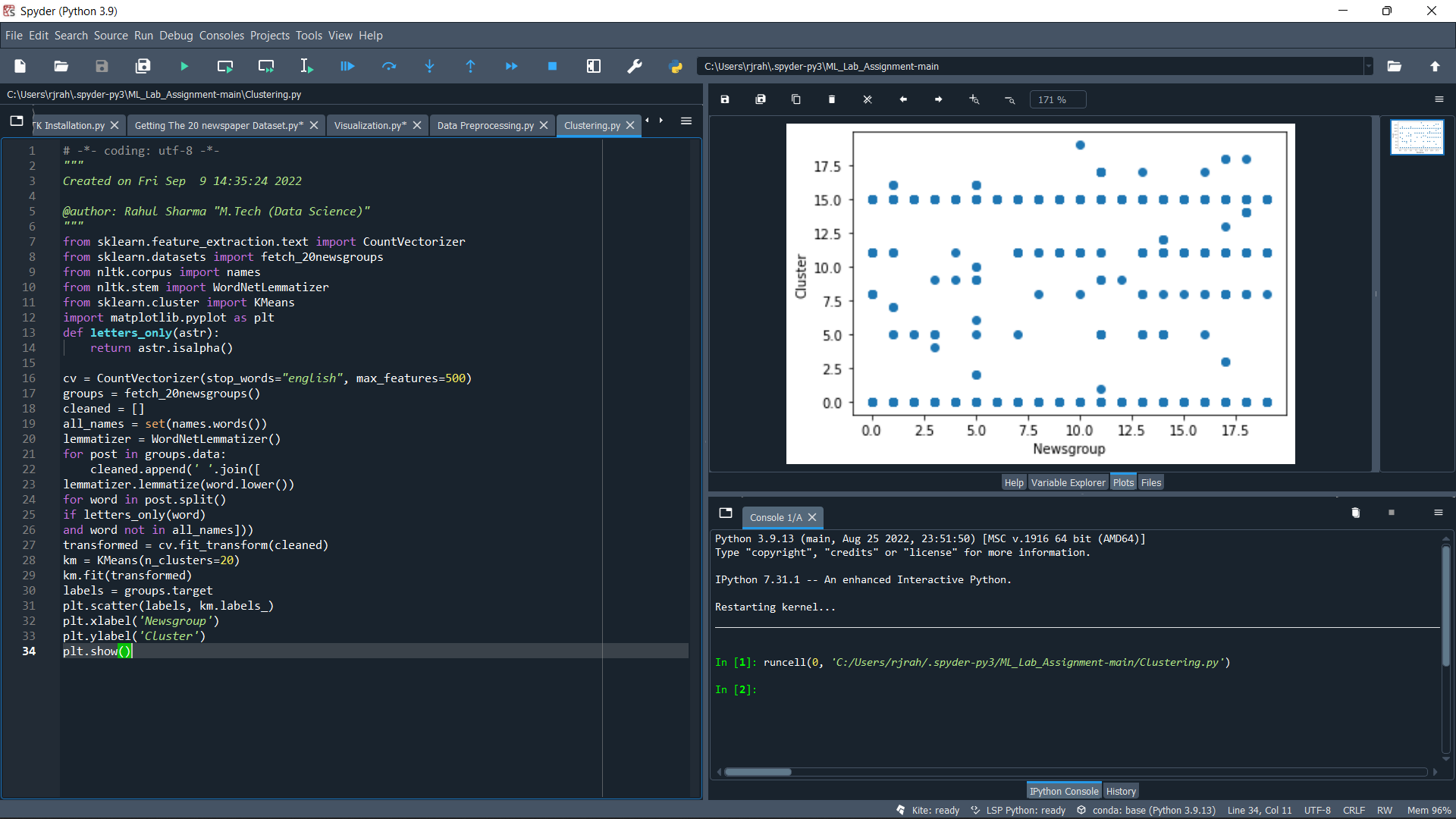
*Output*:



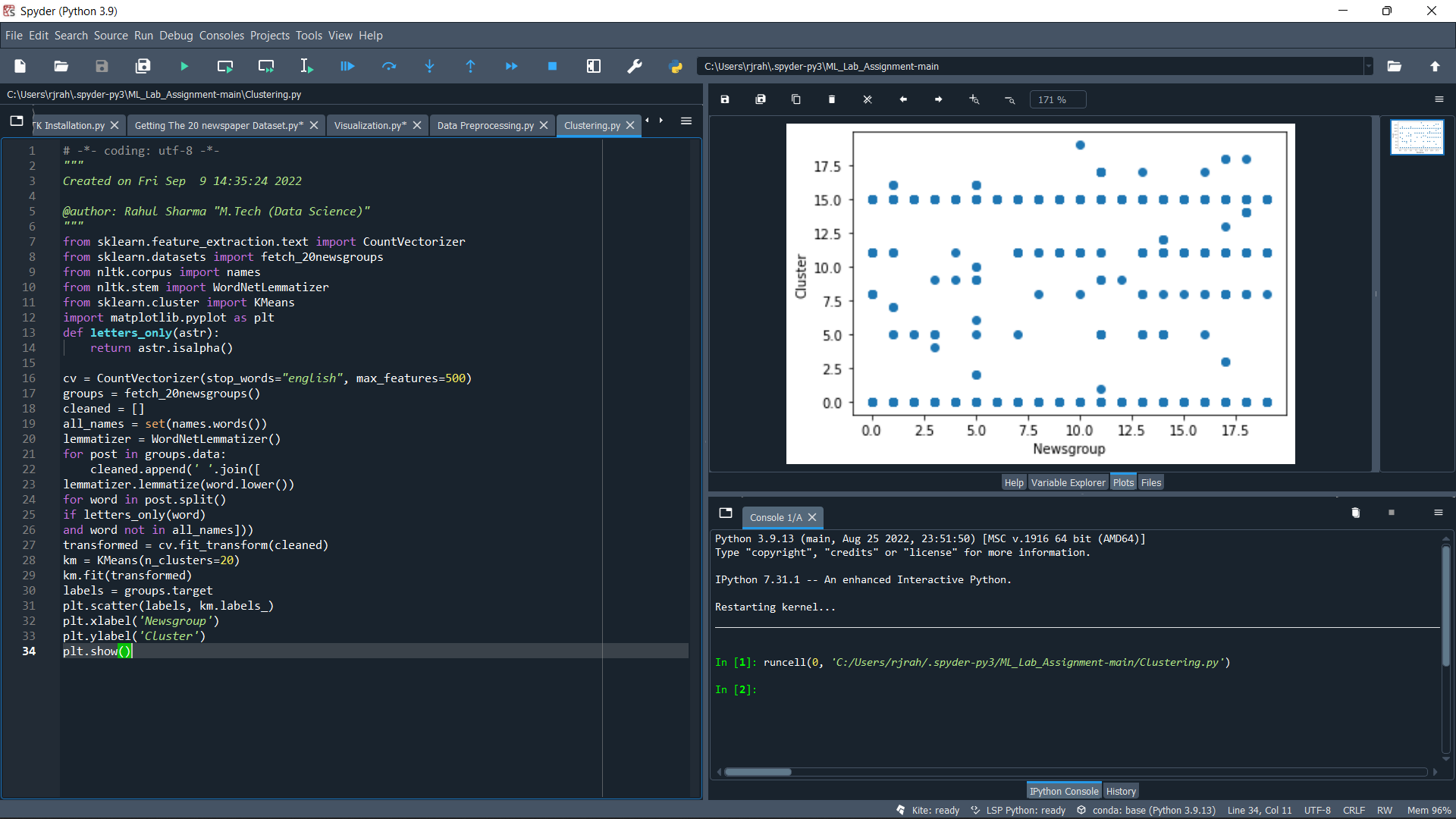
# Milestone 7: Clustering

Clustering is the task of dividing the population or data points into several groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is a collection of objects based on similarity and dissimilarity between them.

*Source Code#*



*Output/Plot:*



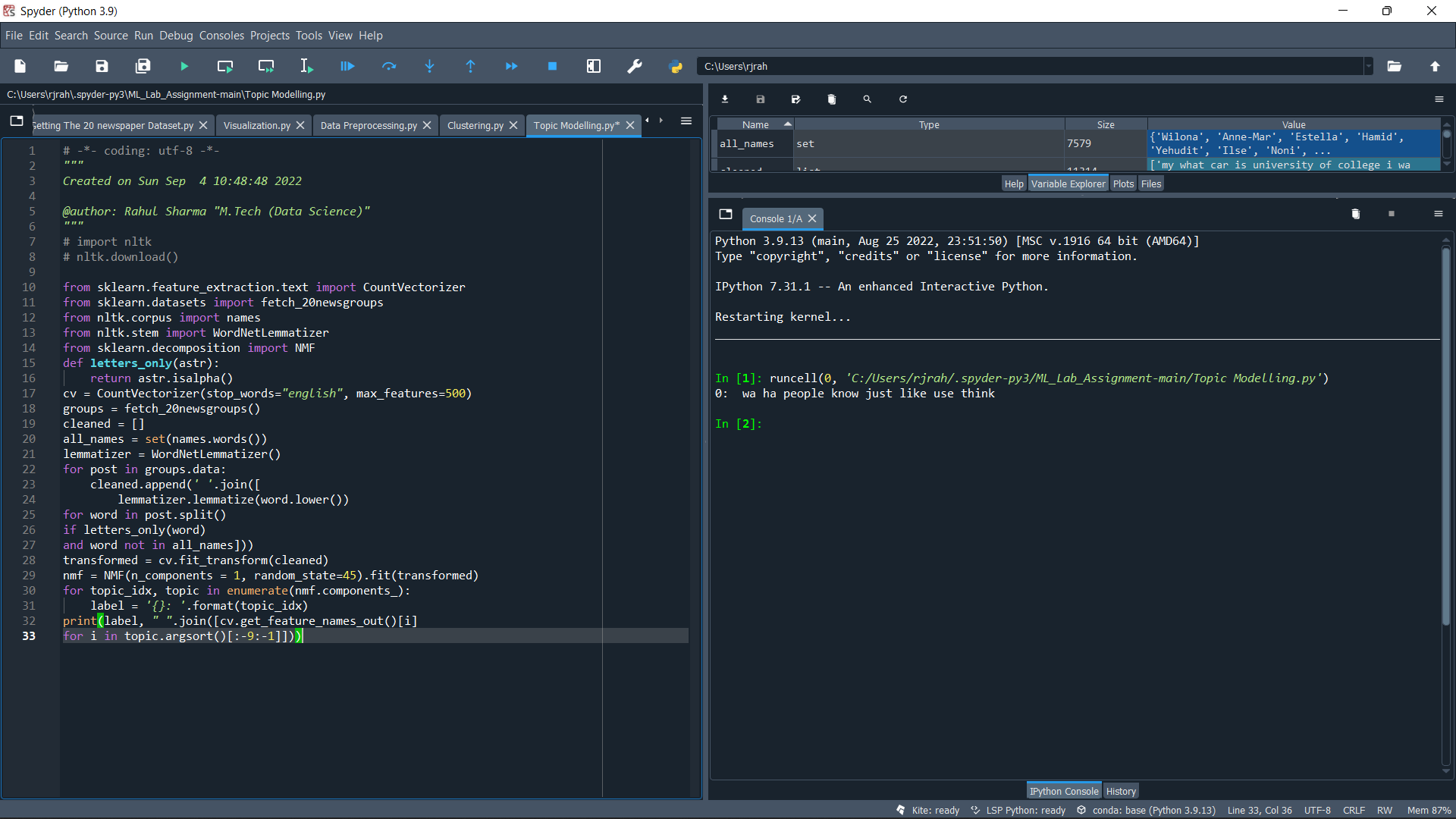
# Milestone 8: Topic Modeling

Topic Modeling falls under unsupervised machine learning where the documents are processed to obtain the relative topics. It is a very important concept of the traditional Natural Processing Approach because of its potential to obtain a semantic relationship between words in the document clusters. In addition, it has numerous other applications in NLP.

Now let us have a look at the Non-Negative Matrix Factorization. NMP is a statistical method to reduce the dimension of the input corpora. It uses the factor analysis method to provide comparatively less weightage to the words with less coherence.

*Source Code#*

***Note:*** “By changing the value n\_component with a positive integer”



*Output:*

0: wa ha people know just like use think

1: db bit data stuff place add time line

2: file use ha program image available version information

3: file image program available use entry version window

4: drive disk hard controller support card scsi head

5: ha think people new make president doe just

6: new anonymous ha use information computer hockey service

7: image file available version data color program software

8: window widget use available application version set server

9: anonymous use information computer internet user service posting

10: people god ha doe just article like make

11: gun state use law new united control government

12: gun state use law new control united national

13: space nasa data ha center national research technology

14: anonymous posting information internet user computer email service

15: people like just think want good make know

16: people like just think know want thing right

17: widget application value data set type use return

18: health medical center number university child research use

19: ha article new like time year doe wa

20: key encryption government technology law use security chip

21: president think going ha wa know said russian

22: use need ground power box usually used doe

23: russian government new american program state way support

24: ibm scsi memory color use card hardware program

25: anonymous posting internet user information email computer service

26: la win san year game period power shot

27: new military war state time united world attack

28: encryption technology government law new use access data

29: drive scsi tape single using driver mode data

30: god atheist believe doe belief religion religious say

31: god atheist believe doe belief religion religious say

32: god atheist believe doe belief religion religious say

33: university center research new medical science article institute

34: university center research new medical institute study science

35: water city new car division san list high

36: good like think just make time really better

37: article university know post read doe usa science

38: good car like better make time year just

39: drive scsi single head mode set using model

40: israeli arab question jew human true attack state

41: like just dod make time look thing really

42: know just like going went come woman came

43: graphic send mail message server support computer package

44: graphic send mail message server support computer package

45: article university post read doe usa discussion world

46: right israeli state human government arab second law

47: power period second game shot got play goal

48: dod time question number computer member list bike

49: article university post read usa world discussion opinion

50: doe say argument make question word point mean

51: just time going maybe make come start got

52: information san said police group league political including

53: israeli arab jew question human attack policy true

54: make thing little le difference better way sure

55: university computer science general department engineering thanks state

56: article university post read world usa discussion opinion

57: university science computer state general department thanks engineering

58: gun control crime rate weapon house american united

59: people say said come tell government live woman

60: graphic send mail message server support package line

61: just start maybe going want really thought woman

62: information san said police group league political including

63: want thing need work help really going school

64: san said information police group league political including

65: game team got win play shot year run

66: article university post read usa world discussion opinion

67: new york year question sale change service old

68: article usa read world post opinion discussion sure

69: san information police said group league political government

70: say word believe mean thing did christian point

71: point way human case child different line better

72: car driver speed change better number high point

73: want need help let really thing better point

74: chip unit clipper serial number disk algorithm bit

75: article post usa world read opinion discussion sure

76: university science general thanks department computer engineering texas

77: say word believe mean christian did act point

78: believe belief christian truth evidence church claim different

79: need help phone kind able needed thanks bike

80: israeli arab human attack policy true state fact

81: university science general department thanks engineering computer texas

82: say word believe christian mean did act day

83: wa did thought used later order seen early

84: want need help let really life better reason

85: university science general thanks department engineering computer texas

86: power period second san special le play shot

87: time long lot better having able order actually

88: card driver video support mouse mode board bus

89: did death let fact money body place man

90: woman men world muslim life religion man great

91: believe belief christian truth evidence claim different church

92: like look really sound better going long little

93: gun rate crime city death study control difference

94: really better lot player probably little best big

95: going come way mean sure working look kind

96: point way different line better algorithm issue view

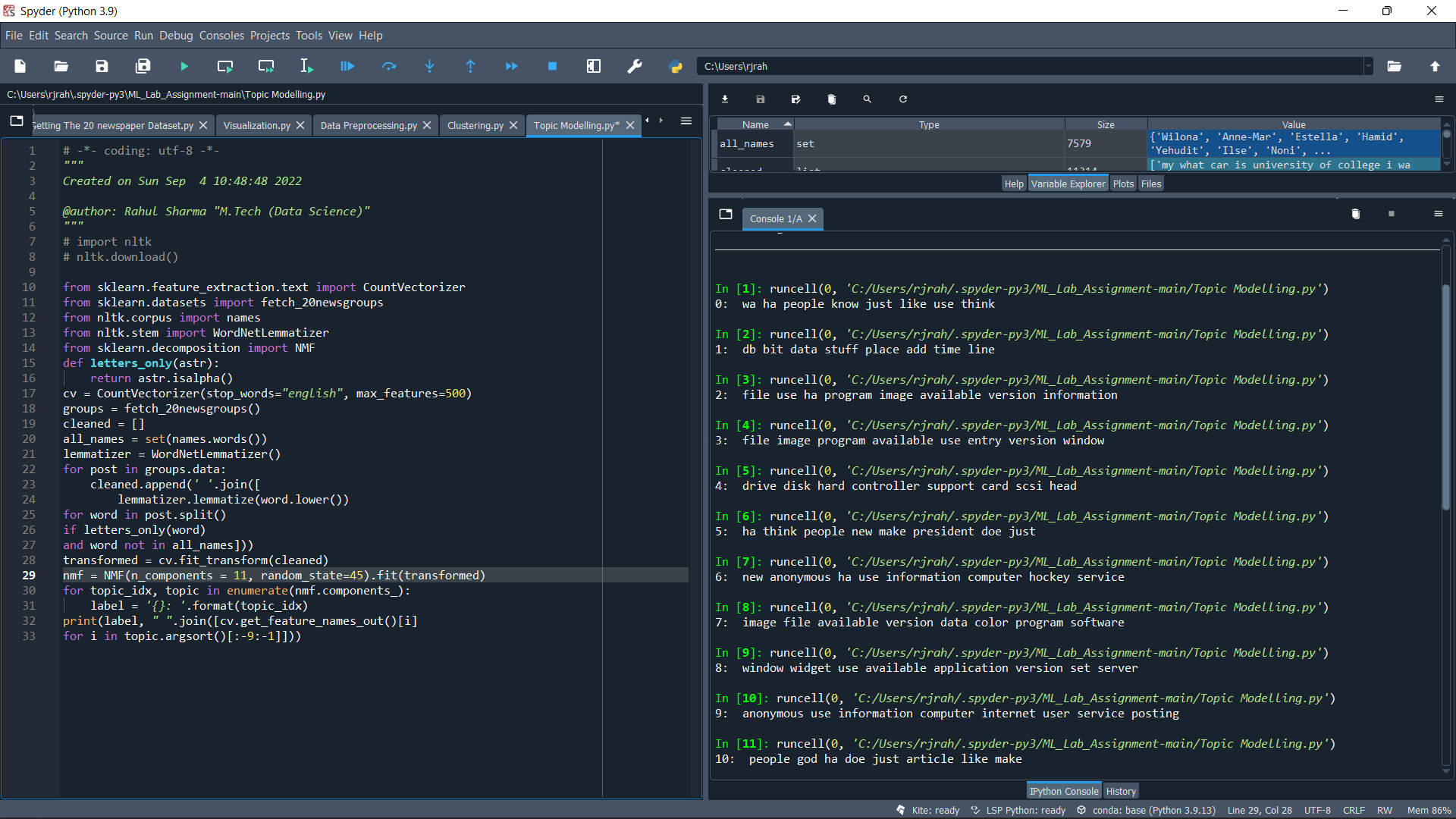
97: just maybe start woman big thought probably getting

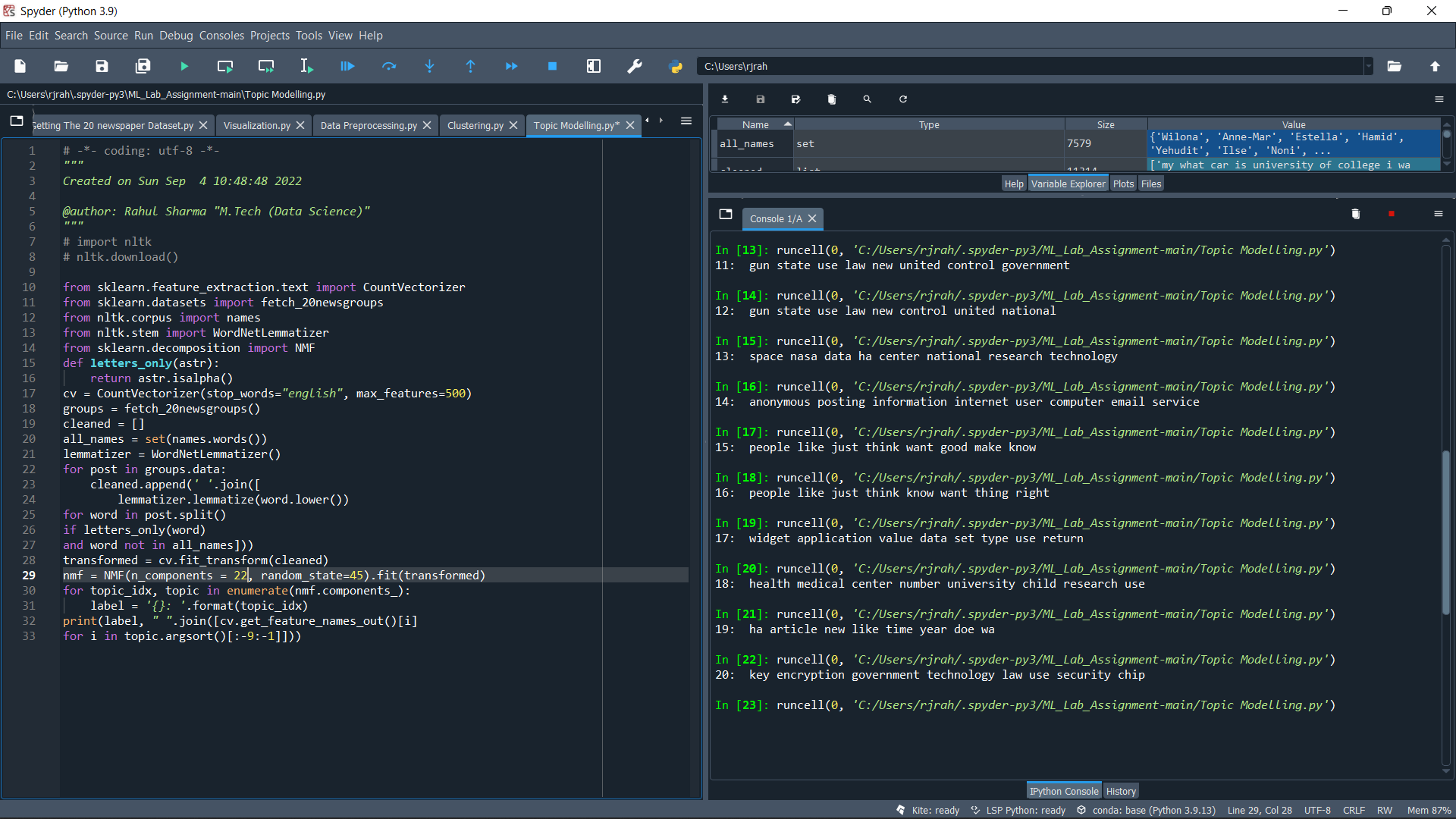
98: government political federal free sure private local country

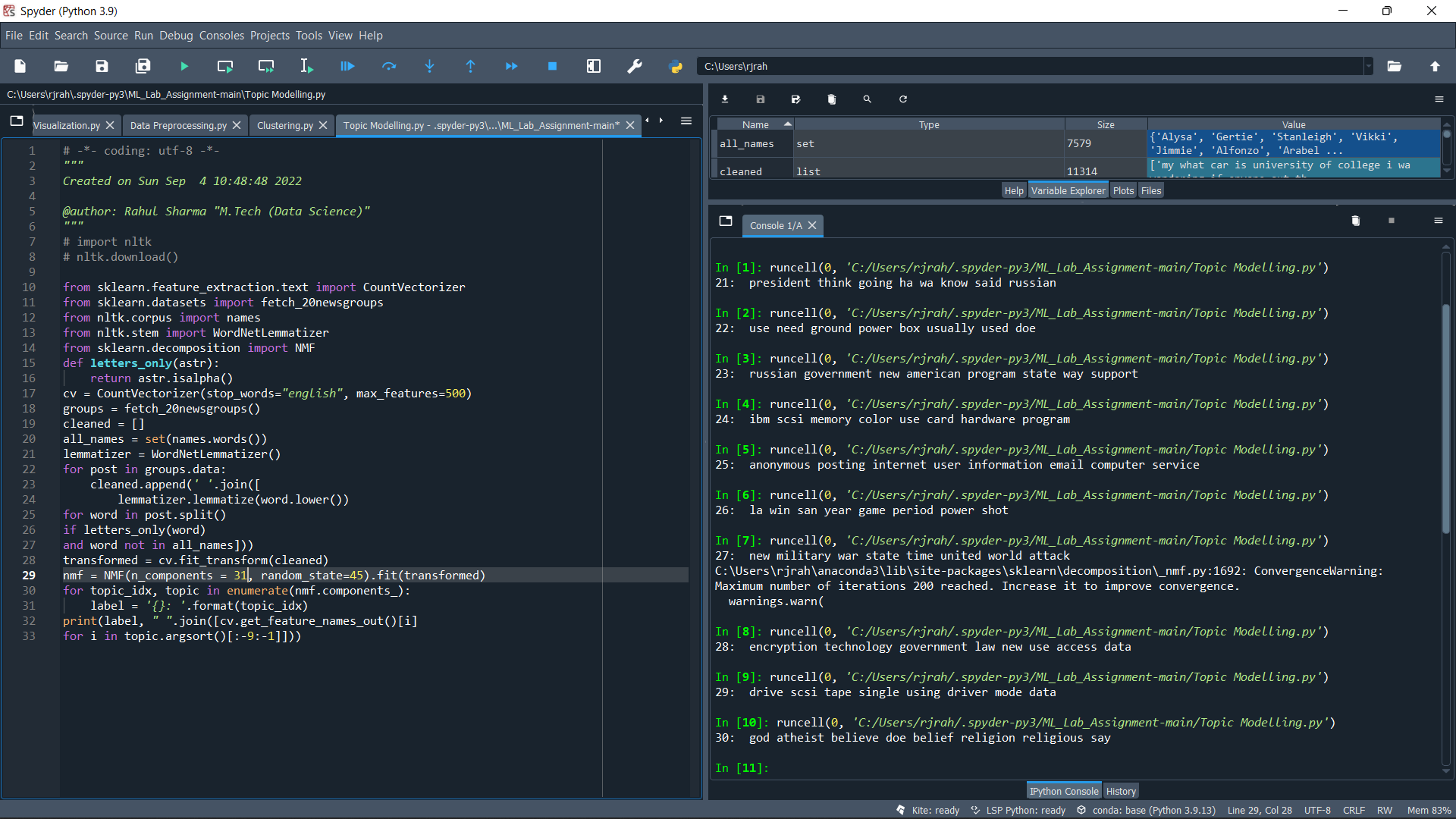
99: article read world usa opinion post discussion sure

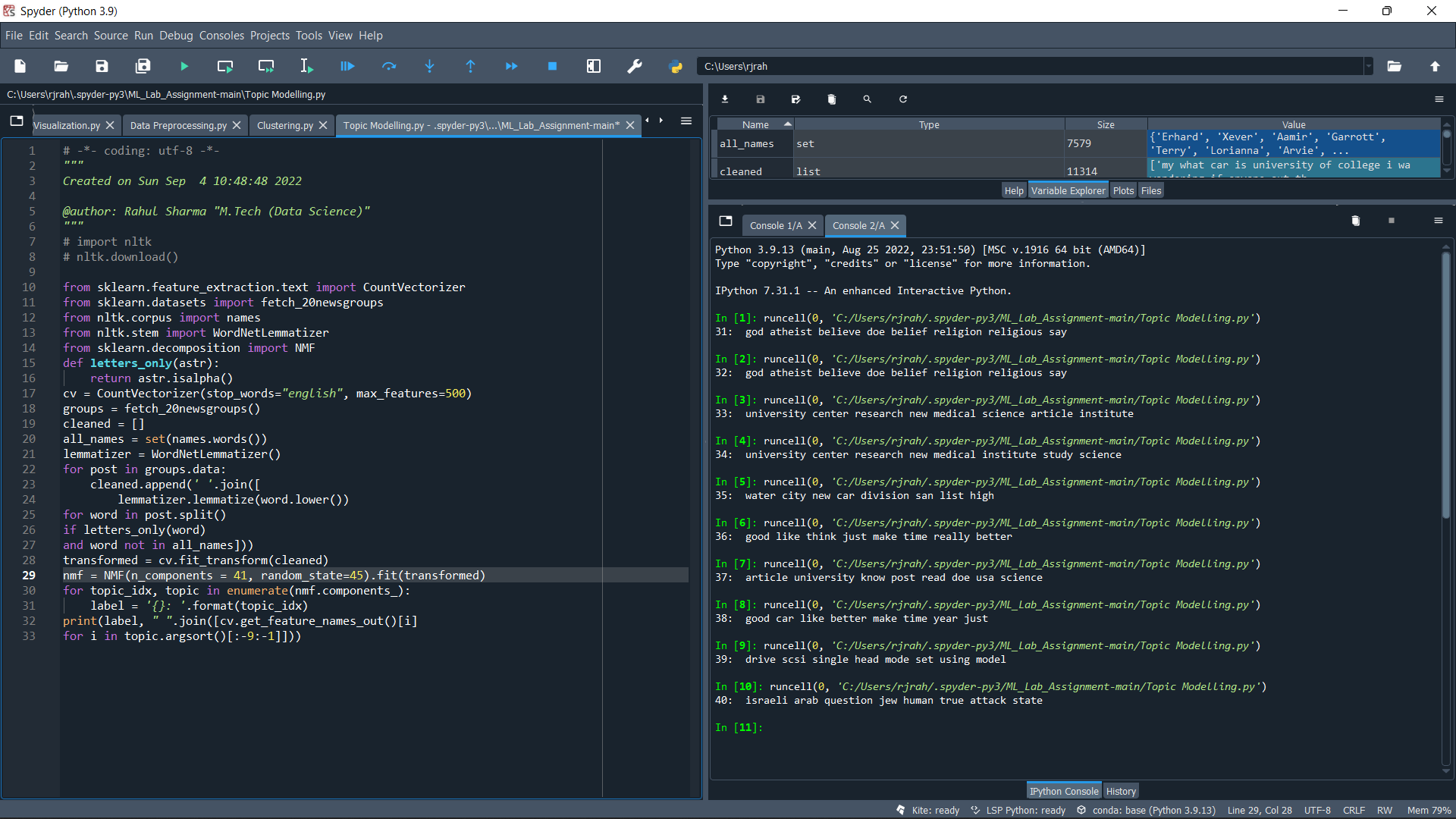
100: state united public national political federal member local

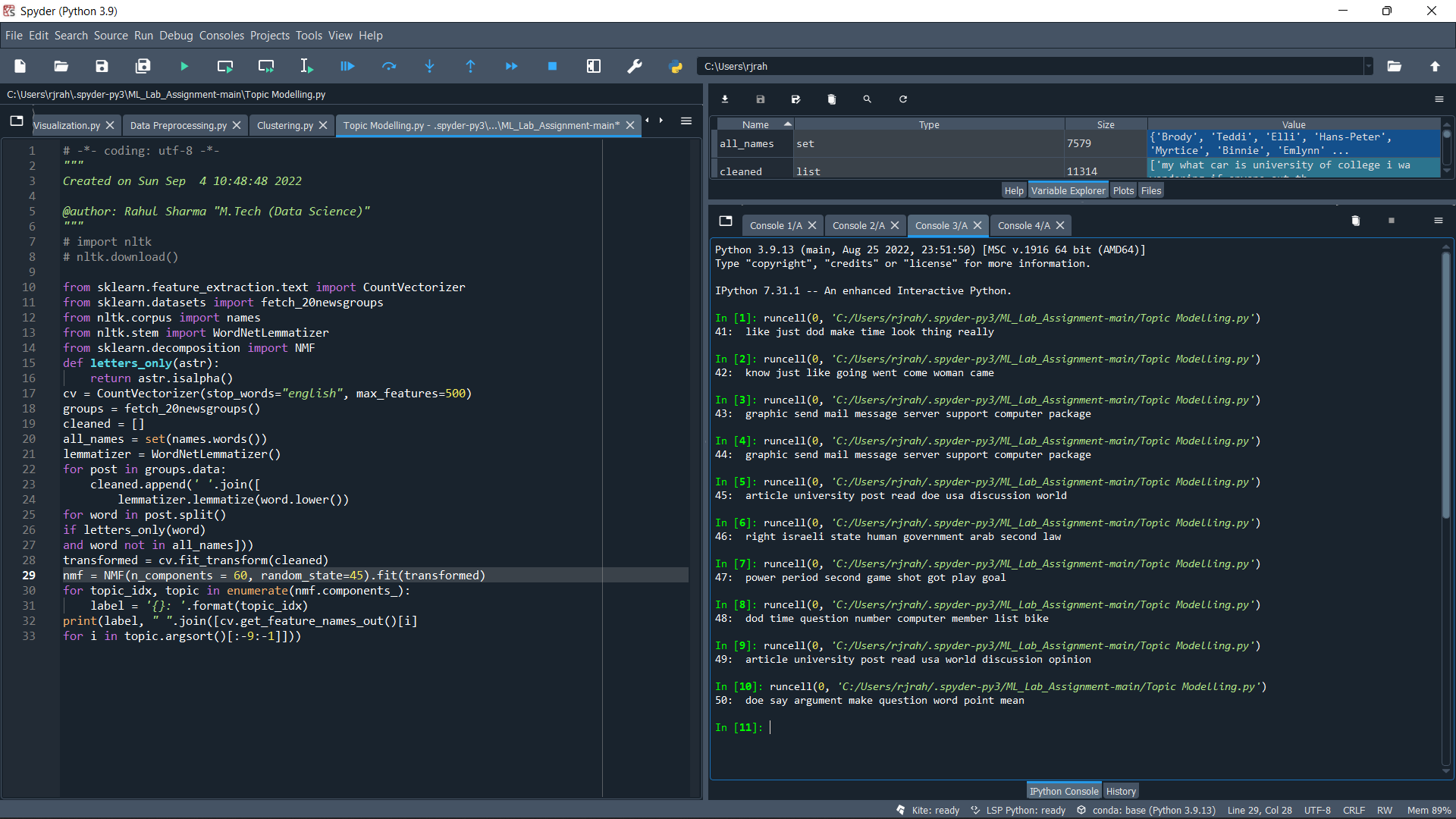
*Screenshots of the above given Output*

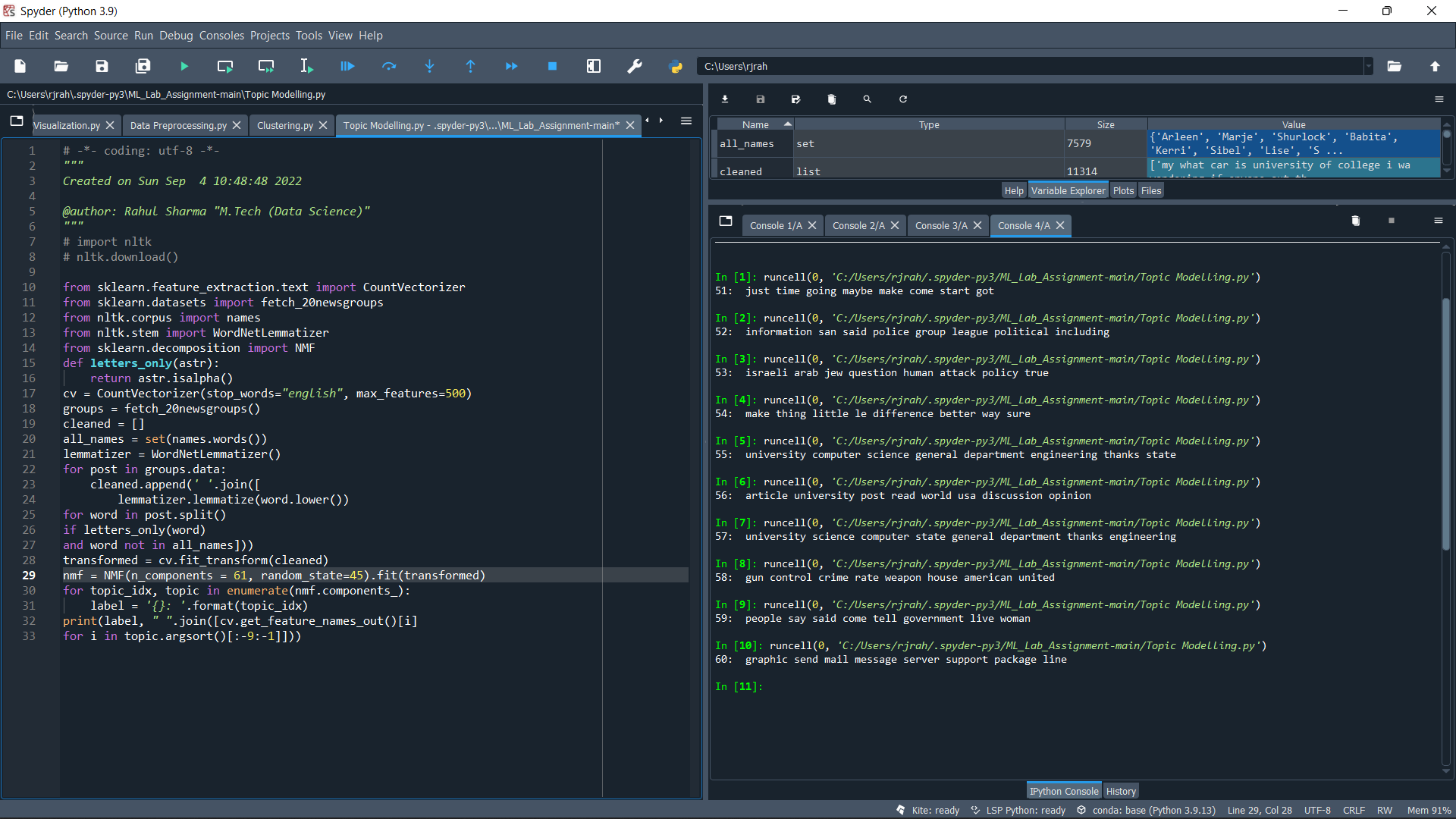


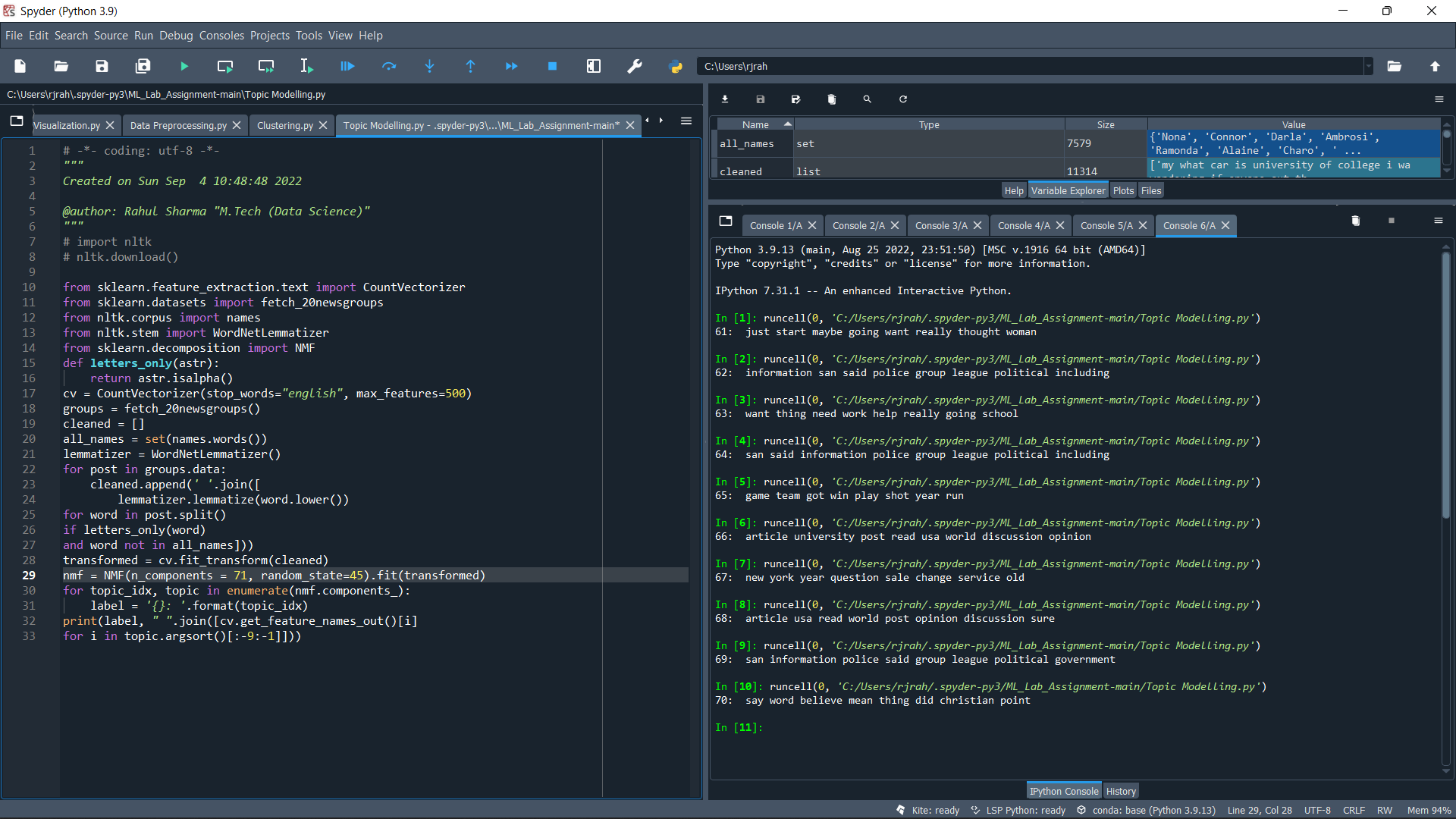


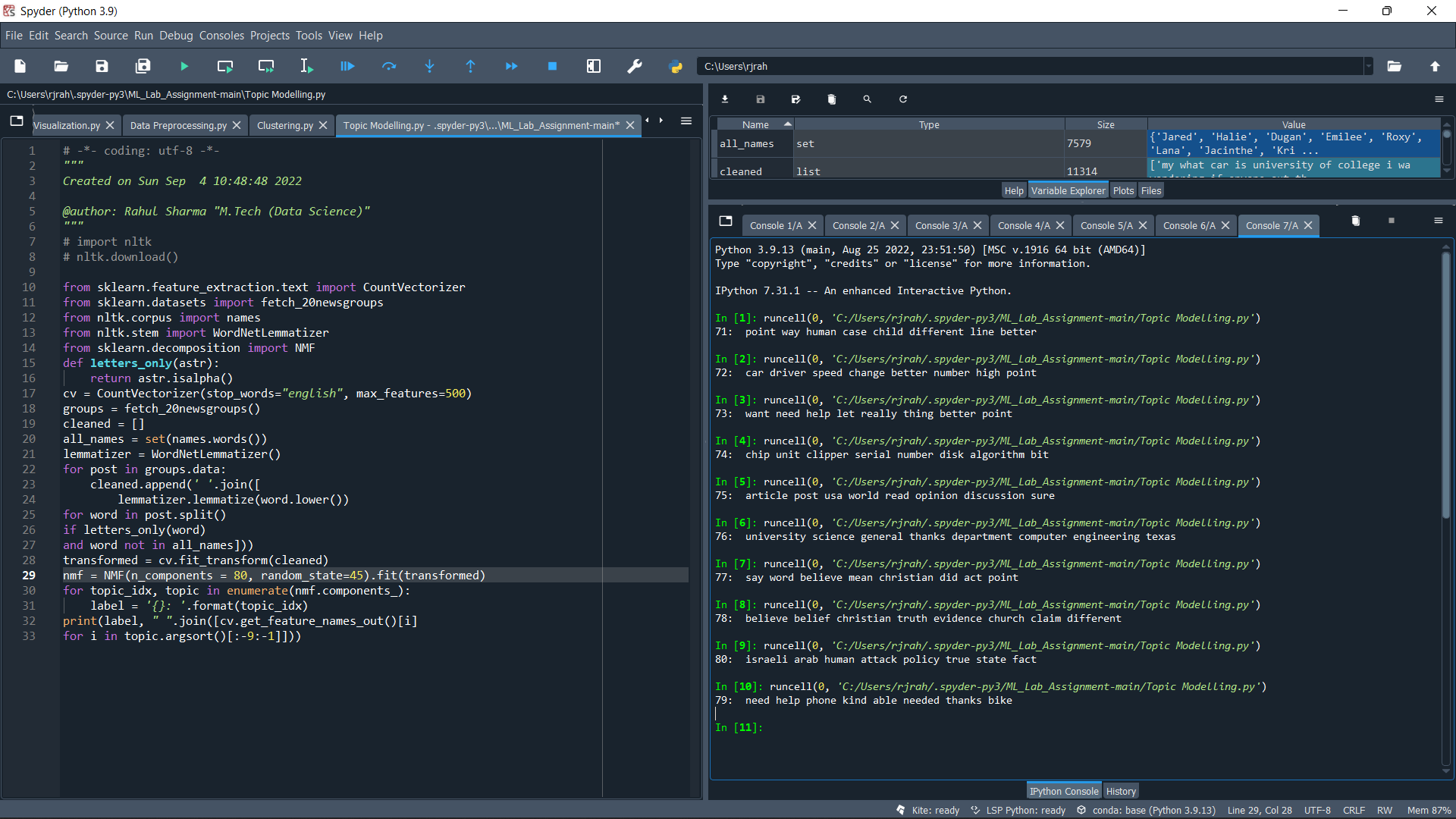


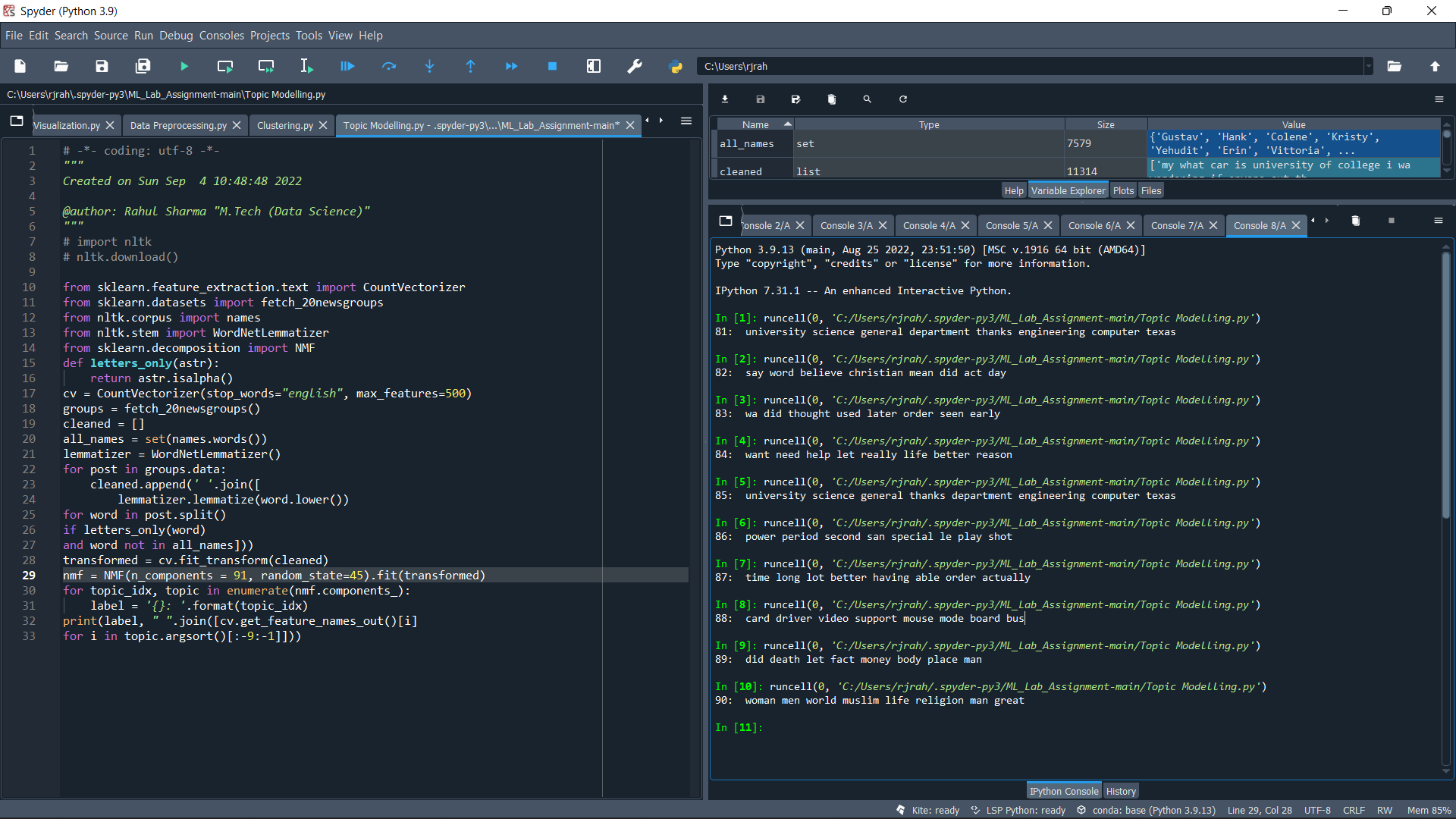


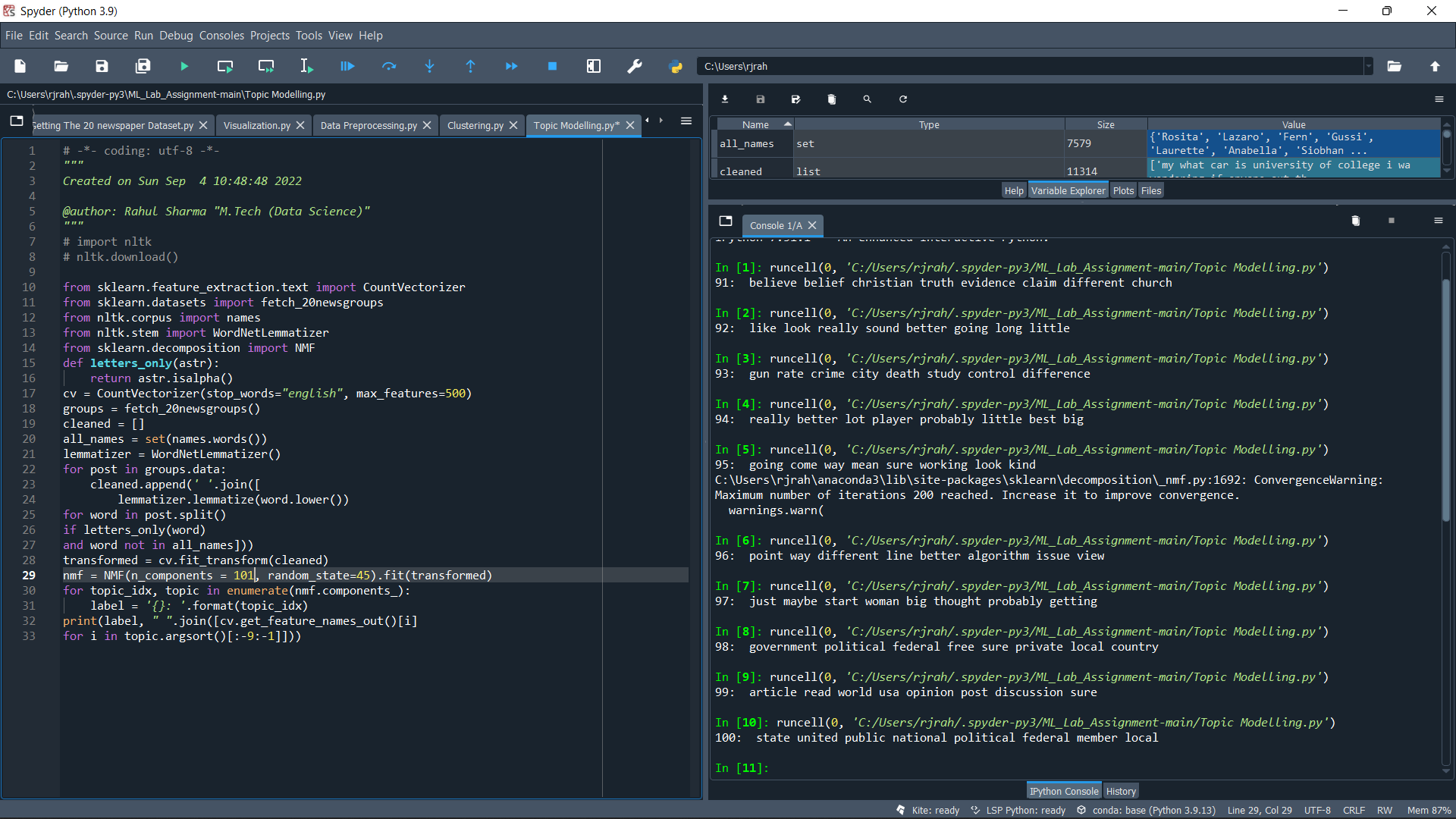












***GitHub-Repository:***

This link redirect to my all-ML Lab Assignment’s source code repository on GitHub

<https://github.com/cse-rahulsharma/ML_Lab_Assignment.git>