NATURAL LANGUAGE PROCESSING (NLP)

Machine Learning

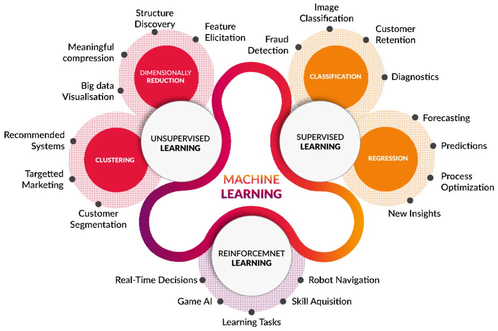
Dr. Adnan Abid

Natural Language Processing - NLP

Natural-language processing (NLP) is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages. NLP is used to apply Machine Learning models to text and language.

Example use:

Teach machines to understand what is said in spoken and written word is the focus of Natural Language Processing. Whenever you dictate something into your iPhone / Android device that is then converted to text, that's an NLP algorithm in action.



NLP History

The history of natural-language processing generally started in the 1950s, although work can be found from earlier periods. In 1950, Alan Turing published an article titled "Computing Machinery and Intelligence" which proposed what is now called the Turing test as a criterion of intelligence.

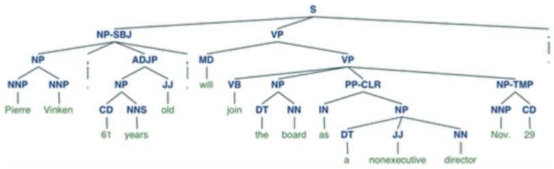
Up to the 1980s, most natural-language processing systems were based on complex sets of hand-written rules. Starting in the late 1980s, however, there was a revolution in natural-language processing with the introduction of machine learning algorithms for language processing.

https://en.wikipedia.org/wiki/Natural-language_processing

NLP Uses

- Sentiment analysis. Identifying the mood or subjective opinions within large amounts of text, including average sentiment and opinion mining.
- Use it to predict the genre of the book.
- Question Answering
- Use NLP to build a machine translator or a speech recognition system
- Document Summarization

NATURAL LANGUAGE PROCESSING - NLP



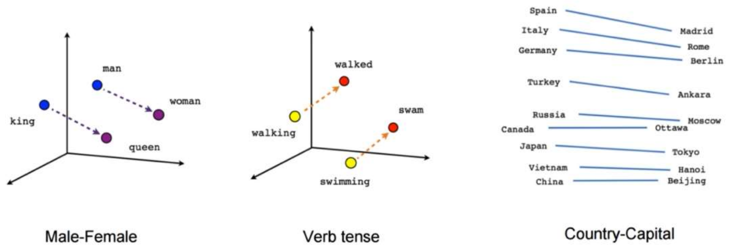
https://www.nltk.org/

Natural Language Processing - NLP

Main NLP library examples:

Natural Language Toolkit - NLTK
SpaCy
Stanford NLP
OpenNLP

Natural Language Processing - NLP



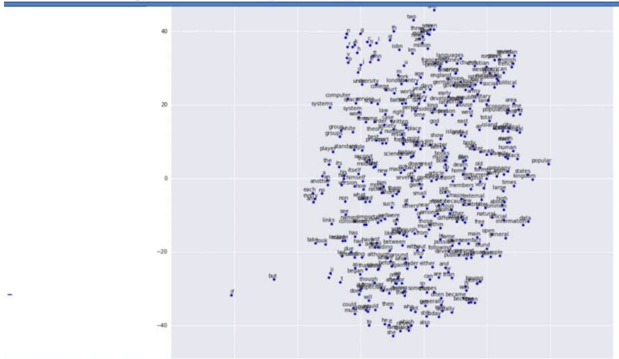
Male-Female

Verb tense

Country-Capital

https://www.tensorflow.org/tutorials/word2vec

Natural Language Processing - NLP



Natural Language Processing

In this part, you will understand and learn how to:

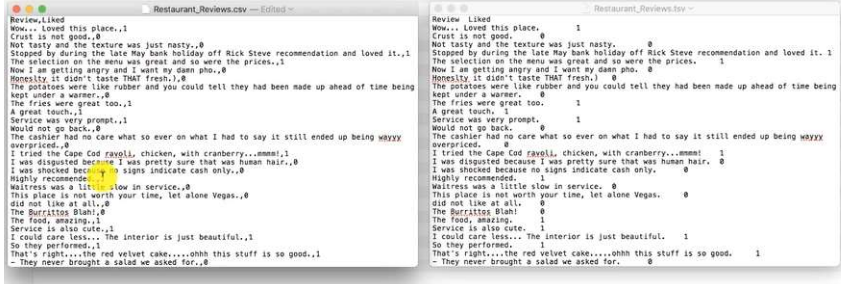
1. Clean texts to prepare them for the Machine Learning models,
2. Create a Bag of Words model,
3. Apply Machine Learning models onto this Bag of Worlds model.

NLP - Bag of Words

Very popular NLP model - It is a model used to preprocess the texts to classify before fitting the classification algorithms on the observations containing the texts.

It involves two things:

1. A vocabulary of known words.
2. A measure of the presence of known words.

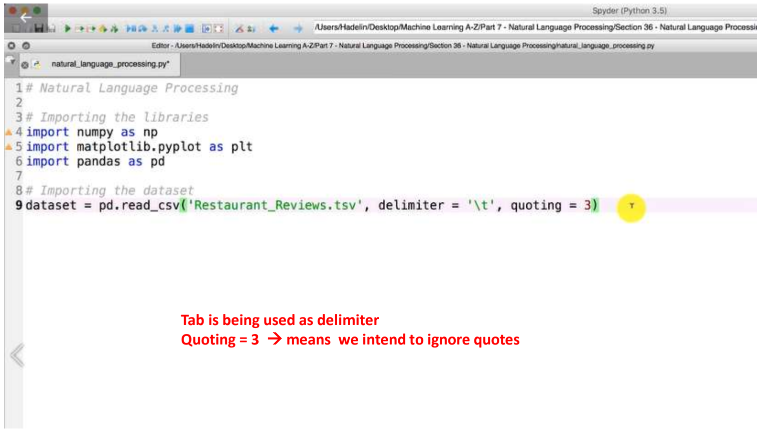


```
Review,Liked
Wow... Loved this place.,1
Crust is not good.,0
Not tasty and the texture was just nasty.,0
Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.,1
The selection on the menu was great and so were the prices.,1
Now I am getting angry and I want my damn pho.,0
Honeslty it didn't taste THAT fresh.),0
The potatoes were like rubber and you could tell they had been made up ahead of time being
kept under a warmer.,0
The fries were great too.,1
A great touch.,1
Service was very prompt.,1
Would not go back.,0
The cashier had no care what so ever on what I had to say it still ended up being wayyy
overpriced.,0
I tried the Cape Cod ravoli, chicken, with cranberry...mmmm!,1
I was disgusted because I was pretty sure that was human hair.,0
I was shocked because no signs indicate cash only.,0
Highly recommended.,1
Waitress was a little slow in service.,0
This place is not worth your time, let alone Vegas.,0
did not like at all.,0
The Burrittos Blah!,0
The food, amazing.,1
Service is also cute.,1
I could care less... The interior is just beautiful.,1
So they performed.,1
That's right....the red velvet cake.....ohhh this stuff is so good.,1
- They never brought a salad we asked for.,0
```

```
Review	Liked
Wow... Loved this place.	1
Crust is not good.	0
Not tasty and the texture was just nasty.	0
Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.	1
The selection on the menu was great and so were the prices.	1
Now I am getting angry and I want my damn pho.	0
Honeslty it didn't taste THAT fresh.)	0
The potatoes were like rubber and you could tell they had been made up ahead of time being
kept under a warmer.	0
The fries were great too.	1
A great touch.	1
Service was very prompt.	1
Would not go back.	0
The cashier had no care what so ever on what I had to say it still ended up being wayyy
overpriced.	0
I tried the Cape Cod ravoli, chicken, with cranberry...mmmm!	1
I was disgusted because I was pretty sure that was human hair.	0
I was shocked because no signs indicate cash only.	0
Highly recommended.	1
Waitress was a little slow in service.	0
This place is not worth your time, let alone Vegas.	0
did not like at all.	0
The Burrittos Blah!	0
The food, amazing.	1
Service is also cute.	1
I could care less... The interior is just beautiful.	1
So they performed.	1
That's right....the red velvet cake.....ohhh this stuff is so good.	1
- They never brought a salad we asked for.	0
```

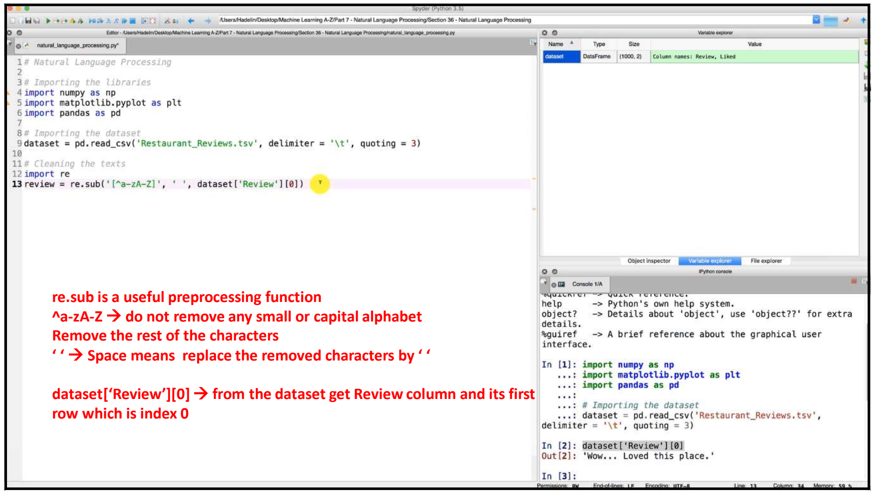
CSV OR TSV?

Commas may be a part of the review comment… which may be misleading
Reviews do not contain any tabs, so TSV is suitable.


Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*

```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)
```

Tab is being used as delimiter
Quoting = 3 → means we intend to ignore quotes



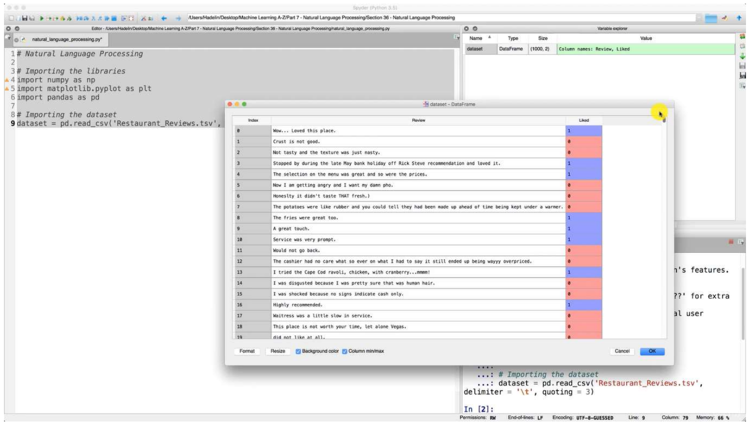
```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][0])
```

Name	Type	Size	Value
dataset	DataFrame	(1000, 2)	Column names: Review, Liked

```
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra
details.
%guiref   -> A brief reference about the graphical user
interface.

In [1]: import numpy as np
   ...: import matplotlib.pyplot as plt
   ...: import pandas as pd
   ...:
   ...: # Importing the dataset
   ...: dataset = pd.read_csv('Restaurant_Reviews.tsv',
delimiter = '\t', quoting = 3)

In [2]: dataset['Review'][0]
Out[2]: 'Wow... Loved this place.'

In [3]:
```

re.sub is a useful preprocessing function
^a-zA-Z → do not remove any small or capital alphabet
Remove the rest of the characters
' ' → Space means replace the removed characters by ' '

dataset['Review'][0] → from the dataset get Review column and its first row which is index 0

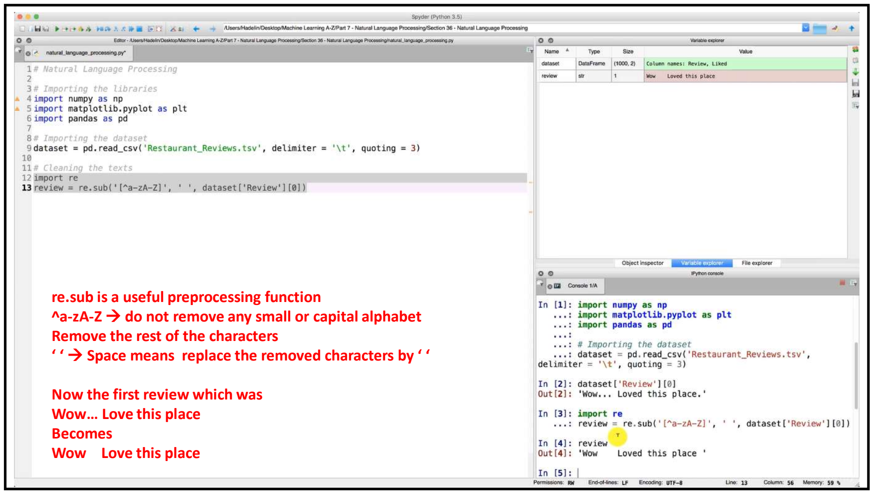


```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv',
```

Name	Type	Size	Value
dataset	DataFrame	(1000, 2)	Column names: Review, Liked

Index	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the texture was just nasty.	0
3	Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.	1
4	The selection on the menu was great and so were the prices.	1
5	Now I am getting angry and I want my damn pho.	0
6	Honeslty it didn't taste THAT fresh.)	0
7	The potatoes were like rubber and you could tell they had been made up ahead of time being kept under a warmer.	0
8	The fries were great too.	1
9	A great touch.	1
10	Service was very prompt.	1
11	Would not go back.	0
12	The cashier had no care what so ever on what I had to say it still ended up being wayyy overpriced.	0
13	I tried the Cape Cod ravoli, chicken, with cranberry...mmmm!	1
14	I was disgusted because I was pretty sure that was human hair.	0
15	I was shocked because no signs indicate cash only.	0
16	Highly recommended.	1
17	Waitress was a little slow in service.	0
18	This place is not worth your time, let alone Vegas.	0
19	did not like at all.	0

```
   ...: # Importing the dataset
   ...: dataset = pd.read_csv('Restaurant_Reviews.tsv',
delimiter = '\t', quoting = 3)

In [2]:
```



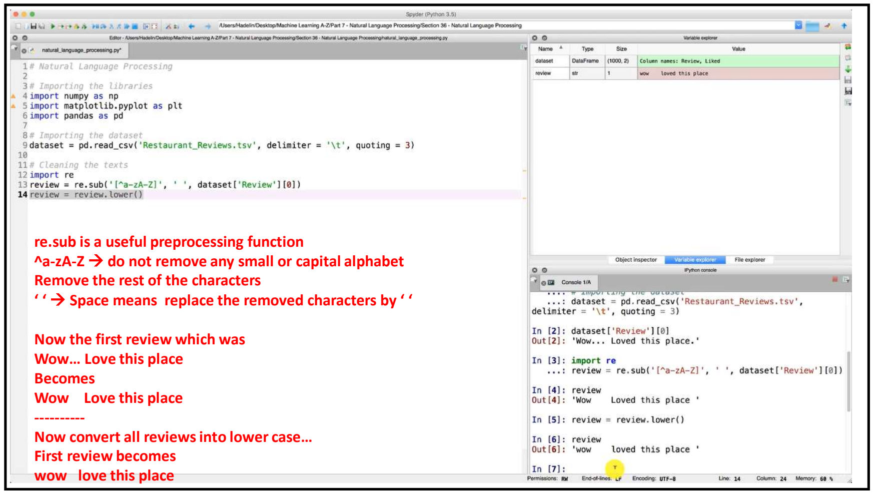
```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][0])
```

Name	Type	Size	Value
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
review	str	1	Wow Loved this place

```
In [1]: import numpy as np
   ...: import matplotlib.pyplot as plt
   ...: import pandas as pd
   ...:
   ...: # Importing the dataset
   ...: dataset = pd.read_csv('Restaurant_Reviews.tsv',
delimiter = '\t', quoting = 3)

In [2]: dataset['Review'][0]
Out[2]: 'Wow... Loved this place.'

In [3]: import re
   ...: review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][0])

In [4]: review
Out[4]: 'Wow    Loved this place '

In [5]:
```

re.sub is a useful preprocessing function
^a-zA-Z → do not remove any small or capital alphabet
Remove the rest of the characters
' ' → Space means replace the removed characters by ' '

Now the first review which was
Wow... Love this place
Becomes
Wow   Love this place


Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*
Variable explorer
Object inspector
Variable explorer
File explorer
IPython console
Console 1/A
Permissions: RW End-of-lines: LF Encoding: UTF-8 Line: 14 Column: 24 Memory: 60 %

```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][0])
review = review.lower()
```

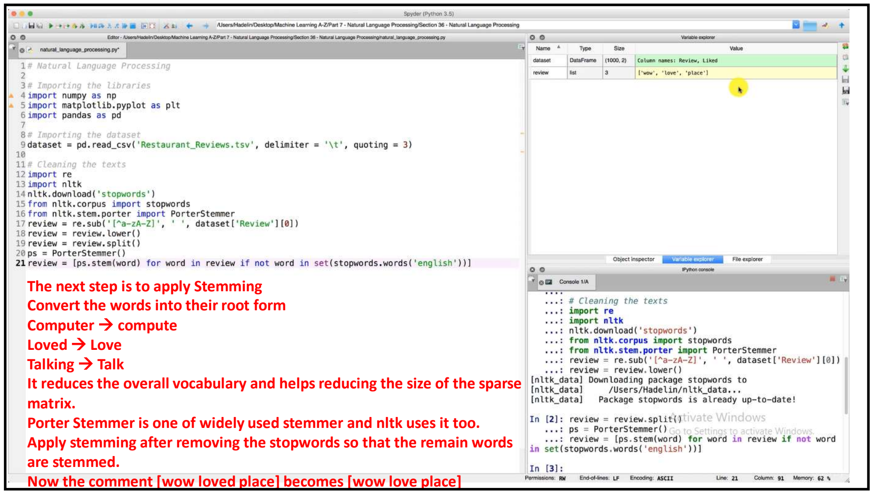
re.sub is a useful preprocessing function
^a-zA-Z → do not remove any small or capital alphabet
Remove the rest of the characters
' ' → Space means replace the removed characters by ' '

Now the first review which was
Wow… Love this place
Becomes
Wow   Love this place

Now convert all reviews into lower case…
First review becomes
wow   love this place

Name	Type	Size	Value
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
review	str	1	wow loved this place

```
   ...: dataset = pd.read_csv('Restaurant_Reviews.tsv',
delimiter = '\t', quoting = 3)

In [2]: dataset['Review'][0]
Out[2]: 'Wow... Loved this place.'

In [3]: import re
   ...: review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][0])

In [4]: review
Out[4]: 'Wow    Loved this place '

In [5]: review = review.lower()

In [6]: review
Out[6]: 'wow    loved this place '

In [7]:
```

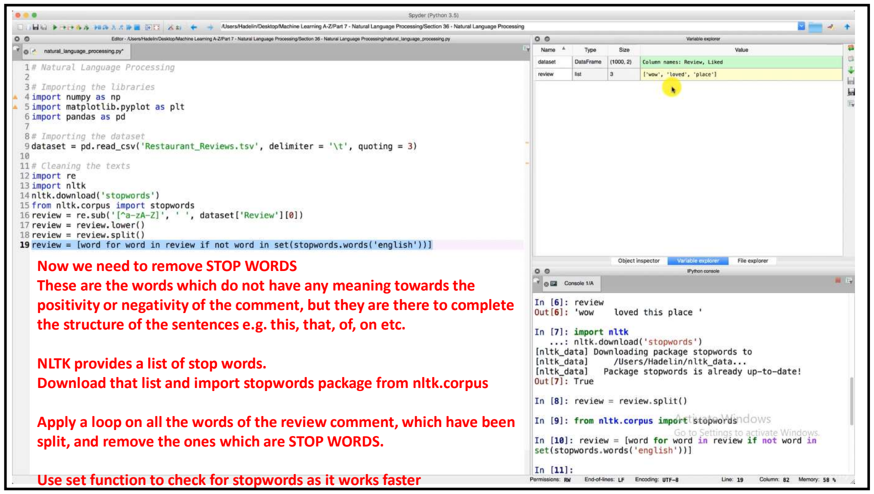

Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*
Variable explorer
Object inspector
Variable explorer
File explorer
IPython console
Console 1/A
Permissions: RW End-of-lines: LF Encoding: ASCII Line: 21 Column: 91 Memory: 62 %

```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][0])
review = review.lower()
review = review.split()
ps = PorterStemmer()
review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
```

The next step is to apply Stemming
Convert the words into their root form
Computer → compute
Loved → Love
Talking → Talk
It reduces the overall vocabulary and helps reducing the size of the sparse matrix.
Porter Stemmer is one of widely used stemmer and nltk uses it too.
Apply stemming after removing the stopwords so that the remain words are stemmed.
Now the comment [wow loved place] becomes [wow love place]

Name	Type	Size	Value
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
review	list	3	['wow', 'love', 'place']

```
   ...: # Cleaning the texts
   ...: import re
   ...: import nltk
   ...: nltk.download('stopwords')
   ...: from nltk.corpus import stopwords
   ...: from nltk.stem.porter import PorterStemmer
   ...: review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][0])
   ...: review = review.lower()
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

In [2]: review = review.split()
   ...: ps = PorterStemmer()
   ...: review = [ps.stem(word) for word in review if not word
in set(stopwords.words('english'))]

In [3]:
```


Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*
Variable explorer
Object inspector
Variable explorer
File explorer
IPython console
Console 1/A
Permissions: RW End-of-lines: LF Encoding: UTF-8 Line: 19 Column: 82 Memory: 58 %

```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][0])
review = review.lower()
review = review.split()
review = [word for word in review if not word in set(stopwords.words('english'))]
```

Now we need to remove STOP WORDS
These are the words which do not have any meaning towards the positivity or negativity of the comment, but they are there to complete the structure of the sentences e.g. this, that, of, on etc.

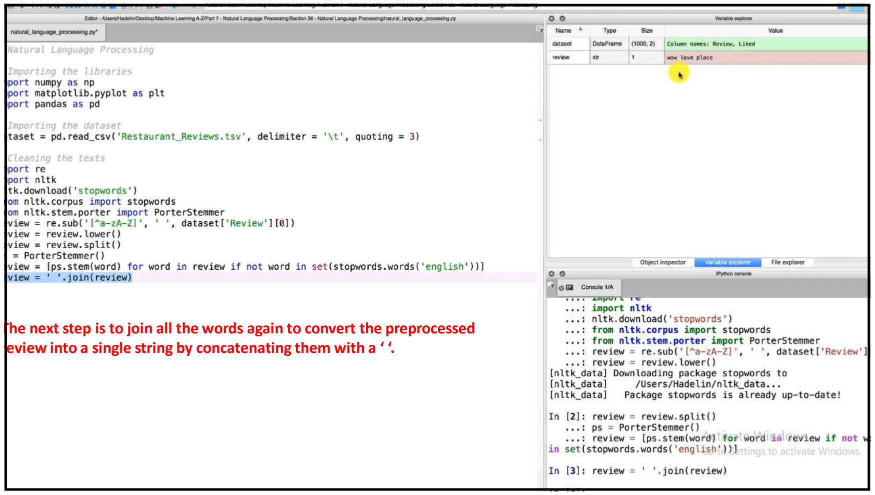
NLTK provides a list of stop words.
Download that list and import stopwords package from nltk.corpus

Apply a loop on all the words of the review comment, which have been split, and remove the ones which are STOP WORDS.

Use set function to check for stopwords as it works faster

Name	Type	Size	Value
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
review	list	3	['wow', 'loved', 'place']

```
In [6]: review
Out[6]: 'wow    loved this place '

In [7]: import nltk
   ...: nltk.download('stopwords')
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Out[7]: True

In [8]: review = review.split()

In [9]: from nltk.corpus import stopwords

In [10]: review = [word for word in review if not word in
set(stopwords.words('english'))]

In [11]:
```

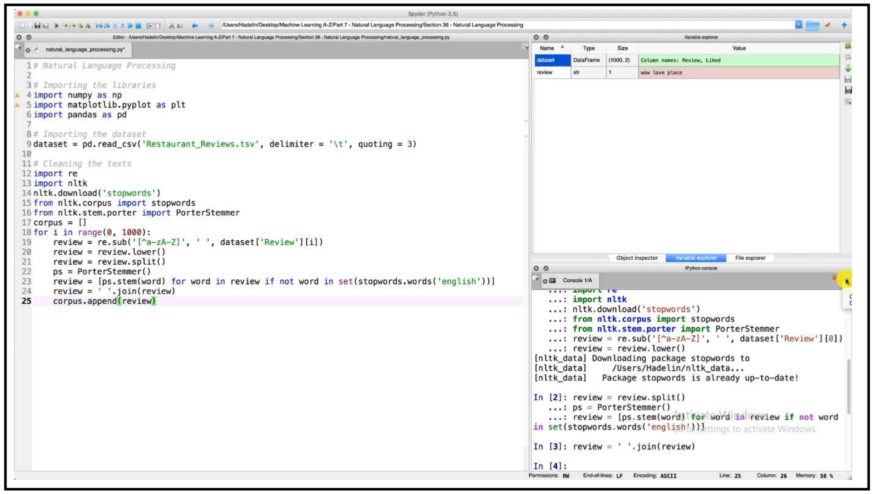

Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*
Variable explorer
Object inspector
Variable explorer
File explorer
IPython console
Console 1/A

```
Natural Language Processing

Importing the libraries
port numpy as np
port matplotlib.pyplot as plt
port pandas as pd

Importing the dataset
taset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

Cleaning the texts
port re
port nltk
tk.download('stopwords')
om nltk.corpus import stopwords
om nltk.stem.porter import PorterStemmer
view = re.sub('[^a-zA-Z]', ' ', dataset['Review'][0])
view = review.lower()
view = review.split()
 = PorterStemmer()
view = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
view = ' '.join(review)
```

he next step is to join all the words again to convert the preprocessed eview into a single string by concatenating them with a ' '.

Name	Type	Size	Value
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
review	str	1	wow love place

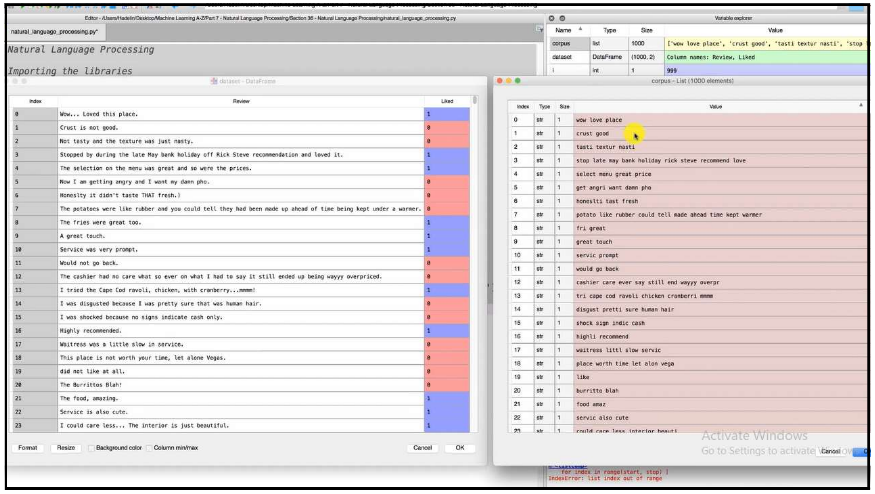
```
   ...: import nltk
   ...: nltk.download('stopwords')
   ...: from nltk.corpus import stopwords
   ...: from nltk.stem.porter import PorterStemmer
   ...: review = re.sub('[^a-zA-Z]', ' ', dataset['Review']
   ...: review = review.lower()
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

In [2]: review = review.split()
   ...: ps = PorterStemmer()
   ...: review = [ps.stem(word) for word in review if not w
in set(stopwords.words('english'))]

In [3]: review = ' '.join(review)
```


Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*
Variable explorer
Object inspector
Variable explorer
File explorer
IPython console
Console 1/A
Permissions: RW
End-of-lines: LF
Encoding: ASCII
Line: 25
Column: 26
Memory: 58 %

```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []
for i in range(0, 1000):
    review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)
```

Name	Type	Size	Value
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
review	str	1	wow love place

```
   ...: import nltk
   ...: nltk.download('stopwords')
   ...: from nltk.corpus import stopwords
   ...: from nltk.stem.porter import PorterStemmer
   ...: review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][0])
   ...: review = review.lower()
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

In [2]: review = review.split()
   ...: ps = PorterStemmer()
   ...: review = [ps.stem(word) for word in review if not word
in set(stopwords.words('english'))]

In [3]: review = ' '.join(review)

In [4]:
```

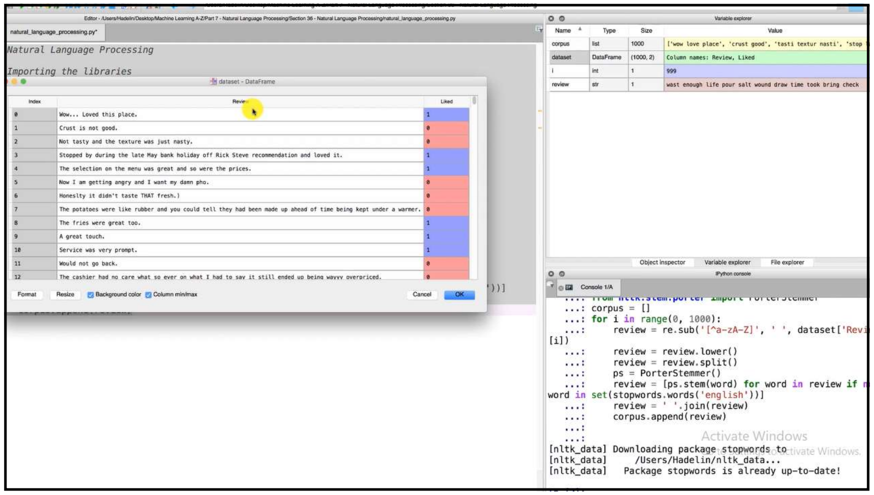

Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*
Natural Language Processing
Importing the libraries
dataset - DataFrame
Format
Resize
Background color
Column min/max
Cancel
OK
Variable explorer
corpus - List (1000 elements)
Cancel

Index	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the texture was just nasty.	0
3	Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.	1
4	The selection on the menu was great and so were the prices.	1
5	Now I am getting angry and I want my damn pho.	0
6	Honeslty it didn't taste THAT fresh.)	0
7	The potatoes were like rubber and you could tell they had been made up ahead of time being kept under a warmer.	0
8	The fries were great too.	1
9	A great touch.	1
10	Service was very prompt.	1
11	Would not go back.	0
12	The cashier had no care what so ever on what I had to say it still ended up being wayyy overpriced.	0
13	I tried the Cape Cod ravoli, chicken, with cranberry...mmmm!	1
14	I was disgusted because I was pretty sure that was human hair.	0
15	I was shocked because no signs indicate cash only.	0
16	Highly recommended.	1
17	Waitress was a little slow in service.	0
18	This place is not worth your time, let alone Vegas.	0
19	did not like at all.	0
20	The Burrittos Blah!	0
21	The food, amazing.	1
22	Service is also cute.	1
23	I could care less... The interior is just beautiful.	1

Name	Type	Size	Value
corpus	list	1000	['wow love place', 'crust good', 'tasti textur nasti', 'stop
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
i	int	1	999

Index	Type	Size	Value
0	str	1	wow love place
1	str	1	crust good
2	str	1	tasti textur nasti
3	str	1	stop late may bank holiday rick steve recommend love
4	str	1	select menu great price
5	str	1	get angri want damn pho
6	str	1	honeslti tast fresh
7	str	1	potato like rubber could tell made ahead time kept warmer
8	str	1	fri great
9	str	1	great touch
10	str	1	servic prompt
11	str	1	would go back
12	str	1	cashier care ever say still end wayyy overpr
13	str	1	tri cape cod ravoli chicken cranberri mmmm
14	str	1	disgust pretti sure human hair
15	str	1	shock sign indic cash
16	str	1	highli recommend
17	str	1	waitress littl slow servic
18	str	1	place worth time let alon vega
19	str	1	like
20	str	1	burritto blah
21	str	1	food amaz
22	str	1	servic also cute
23	str	1	could care less interior beauti

```
    for index in range(start, stop) ]
IndexError: list index out of range
```


Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*
Natural Language Processing
Importing the libraries
dataset - DataFrame
Format
Resize
☑ Background color
☑ Column min/max
Cancel
OK
Variable explorer
Object inspector
Variable explorer
File explorer
IPython console
Console 1/A

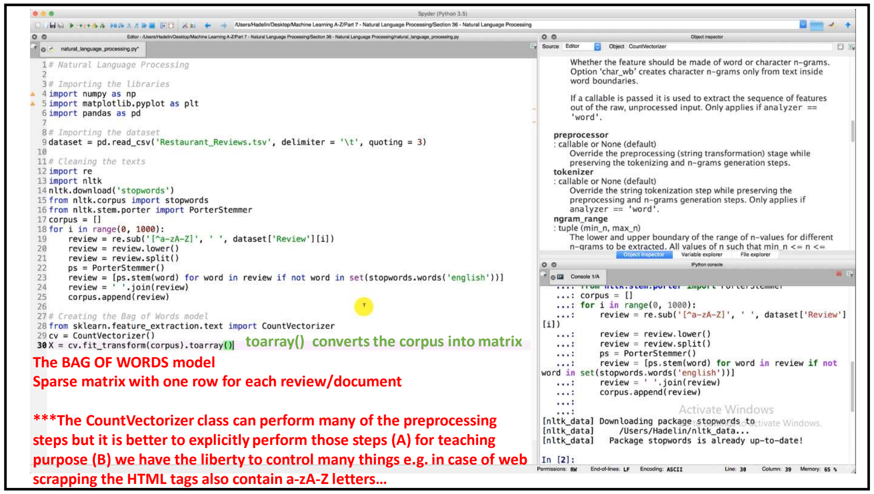
Index	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the texture was just nasty.	0
3	Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.	1
4	The selection on the menu was great and so were the prices.	1
5	Now I am getting angry and I want my damn pho.	0
6	Honeslty it didn't taste THAT fresh.)	0
7	The potatoes were like rubber and you could tell they had been made up ahead of time being kept under a warmer.	0
8	The fries were great too.	1
9	A great touch.	1
10	Service was very prompt.	1
11	Would not go back.	0
12	The cashier had no care what so ever on what I had to say it still ended up being wayyy overpriced.	0

Name	Type	Size	Value
corpus	list	1000	['wow love place', 'crust good', 'tasti textur nasti', 'stop
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
i	int	1	999
review	str	1	wast enough life pour salt wound draw time took bring check

```
   ...: corpus = []
   ...: for i in range(0, 1000):
   ...:     review = re.sub('[^a-zA-Z]', ' ', dataset['Revi
[i])
   ...:     review = review.lower()
   ...:     review = review.split()
   ...:     ps = PorterStemmer()
   ...:     review = [ps.stem(word) for word in review if n
word in set(stopwords.words('english'))]
   ...:     review = ' '.join(review)
   ...:     corpus.append(review)
   ...:
   ...:
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Text Cleaning

- Load Text
- CSV vs TSV
- Convert into lower case
- Split the review comment/doc into words
- Remove stop words
- Stemming
- Combining all reviews/documents


Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*
Object inspector
Source Editor Object CountVectorizer
Object inspector Variable explorer File explorer
IPython console
Console 1/A

```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []
for i in range(0, 1000):
    review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer()
X = cv.fit_transform(corpus).toarray()
```

toarray() converts the corpus into matrix

Whether the feature should be made of word or character n-grams. Option 'char_wb' creates character n-grams only from text inside word boundaries.

If a callable is passed it is used to extract the sequence of features out of the raw, unprocessed input. Only applies if analyzer == 'word'.

preprocessor
: callable or None (default)
Override the preprocessing (string transformation) stage while preserving the tokenizing and n-grams generation steps.

tokenizer
: callable or None (default)
Override the string tokenization step while preserving the preprocessing and n-grams generation steps. Only applies if analyzer == 'word'.

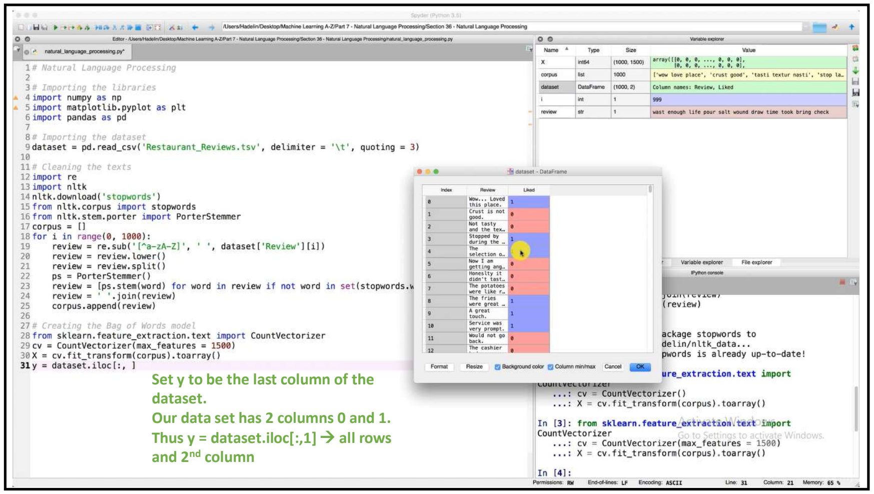
ngram_range
: tuple (min_n, max_n)
The lower and upper boundary of the range of n-values for different n-grams to be extracted. All values of n such that min_n <= n <=

```
   ...: corpus = []
   ...: for i in range(0, 1000):
   ...:     review = re.sub('[^a-zA-Z]', ' ', dataset['Review']
[i])
   ...:     review = review.lower()
   ...:     review = review.split()
   ...:     ps = PorterStemmer()
   ...:     review = [ps.stem(word) for word in review if not
word in set(stopwords.words('english'))]
   ...:     review = ' '.join(review)
   ...:     corpus.append(review)
   ...:
   ...:
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

In [2]:
```

Permissions: RW End-of-lines: LF Encoding: ASCII Line: 30 Column: 39 Memory: 65 %

The BAG OF WORDS model
Sparse matrix with one row for each review/document

***The CountVectorizer class can perform many of the preprocessing steps but it is better to explicitly perform those steps (A) for teaching purpose (B) we have the liberty to control many things e.g. in case of web scrapping the HTML tags also contain a-zA-Z letters…


Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*
Variable explorer
dataset - DataFrame
Format
Resize
Cancel
OK
Variable explorer File explorer
IPython console

```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []
for i in range(0, 1000):
    review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review if not word in set(stopwords.w
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, ]
```

Name	Type	Size	Value
X	int64	(1000, 1500)	array([[0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0],
corpus	list	1000	['wow love place', 'crust good', 'tasti textur nasti', 'stop la…
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
i	int	1	999
review	str	1	wast enough life pour salt wound draw time took bring check

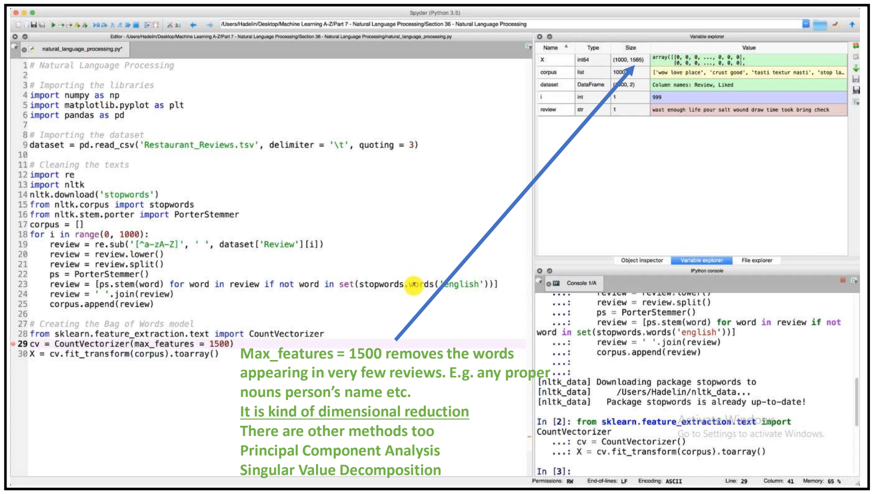
Index	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the tex…	0
3	Stopped by during the …	1
4	The selection o…	1
5	Now I am getting ang…	0
6	Honeslty it didn't tast…	0
7	The potatoes were like r…	0
8	The fries were great …	1
9	A great touch.	1
10	Service was very prompt.	1
11	Would not go back.	0
12	The cashier …	0

☑ Background color ☑ Column min/max

```
j.join(review)
(review)

ackage stopwords to
delin/nltk_data...
pwords is already up-to-date!

ure_extraction.text import
CountVectorizer
   ...: cv = CountVectorizer()
   ...: X = cv.fit_transform(corpus).toarray()

In [3]: from sklearn.feature_extraction.text import
CountVectorizer
   ...: cv = CountVectorizer(max_features = 1500)
   ...: X = cv.fit_transform(corpus).toarray()

In [4]:
```

Permissions: RW End-of-lines: LF Encoding: ASCII Line: 31 Column: 21 Memory: 65 %

Set y to be the last column of the dataset.
Our data set has 2 columns 0 and 1.
Thus y = dataset.iloc[:,1] → all rows and 2nd column


Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*
Variable explorer
Object inspector Variable explorer File explorer
IPython console
Console 1/A

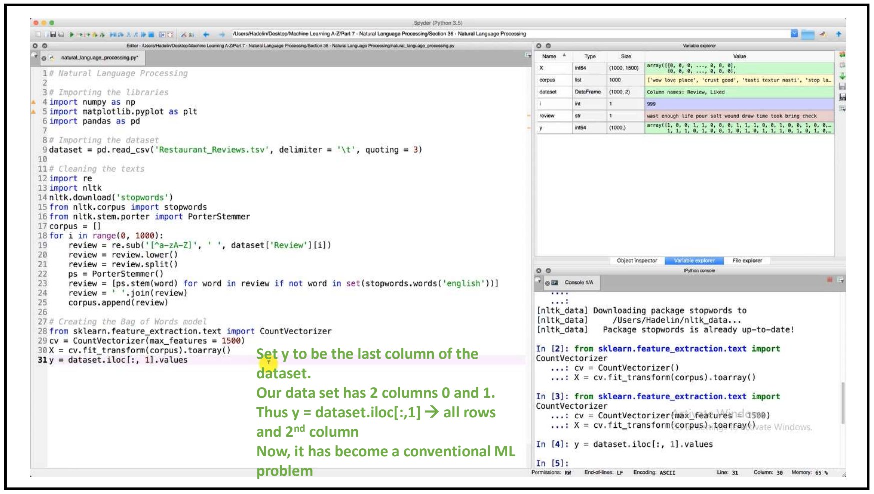
```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []
for i in range(0, 1000):
    review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
```

Name	Type	Size	Value
X	int64	(1000, 1565)	array([[0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0],
corpus	list	1000	['wow love place', 'crust good', 'tasti textur nasti', 'stop la…
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
i	int	1	999
review	str	1	wast enough life pour salt wound draw time took bring check

```
   ...:     review = review.split()
   ...:     ps = PorterStemmer()
   ...:     review = [ps.stem(word) for word in review if not
word in set(stopwords.words('english'))]
   ...:     review = ' '.join(review)
   ...:     corpus.append(review)
   ...:
   ...:
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

In [2]: from sklearn.feature_extraction.text import
CountVectorizer
   ...: cv = CountVectorizer()
   ...: X = cv.fit_transform(corpus).toarray()

In [3]:
```

Permissions: RW End-of-lines: LF Encoding: ASCII Line: 29 Column: 41 Memory: 65 %

Max_features = 1500 removes the words appearing in very few reviews. E.g. any proper nouns person's name etc.
It is kind of dimensional reduction
There are other methods too
Principal Component Analysis
Singular Value Decomposition


Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*
Variable explorer
Object inspector Variable explorer File explorer
IPython console
Console 1/A

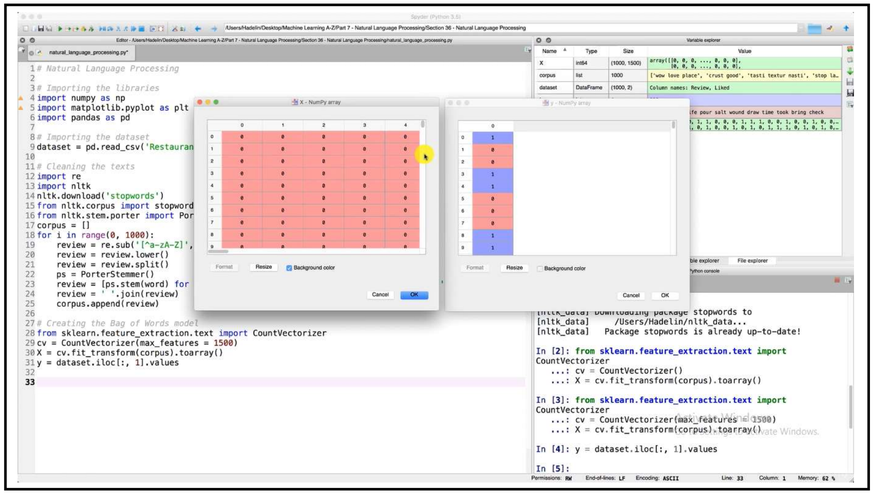
```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []
for i in range(0, 1000):
    review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values
```

Name	Type	Size	Value
X	int64	(1000, 1500)	array([[0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0],
corpus	list	1000	['wow love place', 'crust good', 'tasti textur nasti', 'stop la…
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
i	int	1	999
review	str	1	wast enough life pour salt wound draw time took bring check
y	int64	(1000,)	array([1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0,… 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0,…

```
   ...:
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

In [2]: from sklearn.feature_extraction.text import
CountVectorizer
   ...: cv = CountVectorizer()
   ...: X = cv.fit_transform(corpus).toarray()

In [3]: from sklearn.feature_extraction.text import
CountVectorizer
   ...: cv = CountVectorizer(max_features = 1500)
   ...: X = cv.fit_transform(corpus).toarray()

In [4]: y = dataset.iloc[:, 1].values

In [5]:
```

Permissions: RW End-of-lines: LF Encoding: ASCII Line: 31 Column: 30 Memory: 65 %

Set y to be the last column of the dataset.
Our data set has 2 columns 0 and 1.
Thus y = dataset.iloc[:,1] → all rows and 2nd column
Now, it has become a conventional ML problem


Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
natural_language_processing.py*
X - NumPy array
y - NumPy array
Format
Resize
Background color
Cancel
OK
Variable explorer
File explorer
IPython console

```python
# Natural Language Processing

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restauran

# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopword
from nltk.stem.porter import Por
corpus = []
for i in range(0, 1000):
    review = re.sub('[^a-zA-Z]',
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values
```

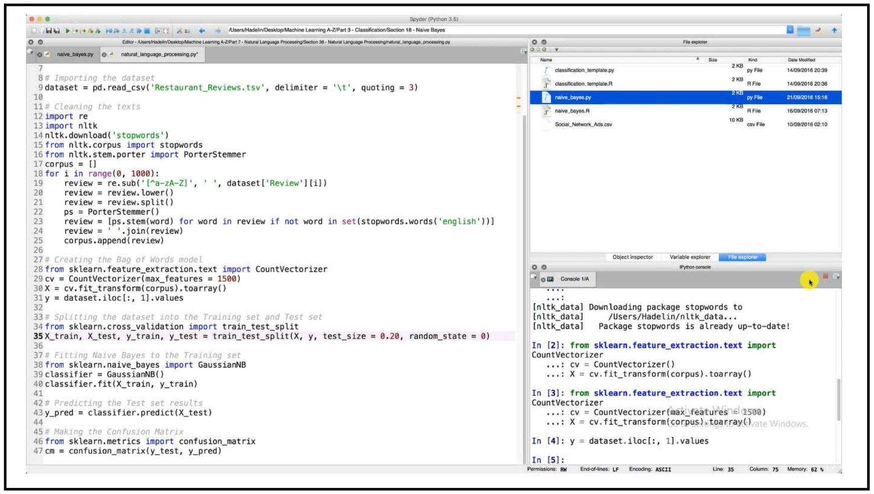
	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0

	0
0	1
1	0
2	0
3	1
4	1
5	0
6	0
7	0
8	1
9	1

Name	Type	Size	Value
X	int64	(1000, 1500)	array([[0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0],
corpus	list	1000	['wow love place', 'crust good', 'tasti textur nasti', 'stop la…
dataset	DataFrame	(1000, 2)	Column names: Review, Liked

```
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

In [2]: from sklearn.feature_extraction.text import
CountVectorizer
   ...: cv = CountVectorizer()
   ...: X = cv.fit_transform(corpus).toarray()

In [3]: from sklearn.feature_extraction.text import
CountVectorizer
   ...: cv = CountVectorizer(max_features = 1500)
   ...: X = cv.fit_transform(corpus).toarray()

In [4]: y = dataset.iloc[:, 1].values

In [5]:
```

Permissions: RW End-of-lines: LF Encoding: ASCII Line: 33 Column: 1 Memory: 62 %

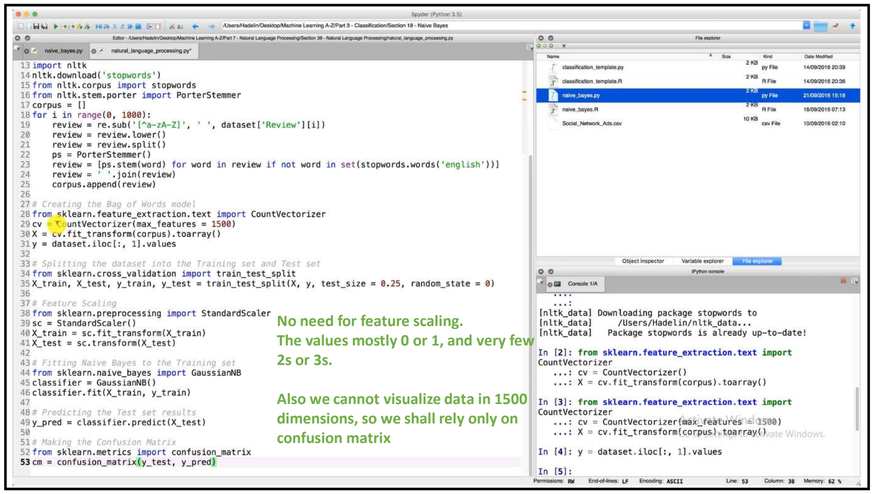

Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 3 - Classification/Section 18 - Naive Bayes
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
naive_bayes.py
natural_language_processing.py*
File explorer
Object inspector
Variable explorer
File explorer
IPython console
Console 1/A

```python

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []
for i in range(0, 1000):
    review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values

# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)

# Fitting Naive Bayes to the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

Name	Size	Kind	Date Modified
classification_template.py	2 KB	py File	14/09/2016 20:39
classification_template.R	2 KB	R File	14/09/2016 20:36
naive_bayes.py	2 KB	py File	21/09/2016 15:16
naive_bayes.R	2 KB	R File	16/09/2016 07:13
Social_Network_Ads.csv	10 KB	csv File	10/09/2016 02:10

```
   ...:
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

In [2]: from sklearn.feature_extraction.text import
CountVectorizer
   ...: cv = CountVectorizer()
   ...: X = cv.fit_transform(corpus).toarray()

In [3]: from sklearn.feature_extraction.text import
CountVectorizer
   ...: cv = CountVectorizer(max_features = 1500)
   ...: X = cv.fit_transform(corpus).toarray()

In [4]: y = dataset.iloc[:, 1].values

In [5]:
```

Permissions: RW End-of-lines: LF Encoding: ASCII Line: 35 Column: 75 Memory: 62 %


Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 3 - Classification/Section 18 - Naive Bayes
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
naive_bayes.py
natural_language_processing.py*
File explorer
Object inspector
Variable explorer
File explorer
IPython console
Console 1/A

```python
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []
for i in range(0, 1000):
    review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values

# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Fitting Naive Bayes to the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

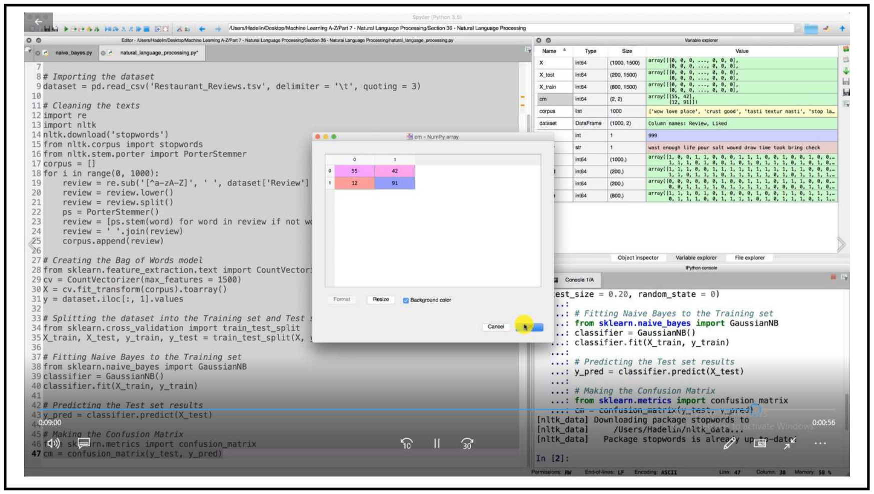
No need for feature scaling.
The values mostly 0 or 1, and very few 2s or 3s.

Also we cannot visualize data in 1500 dimensions, so we shall rely only on confusion matrix

Name	Size	Kind	Date Modified
classification_template.py	2 KB	py File	14/09/2016 20:39
classification_template.R	2 KB	R File	14/09/2016 20:36
naive_bayes.py	2 KB	py File	21/09/2016 15:16
naive_bayes.R	2 KB	R File	16/09/2016 07:13
Social_Network_Ads.csv	10 KB	csv File	10/09/2016 02:10

```
   ...:
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

In [2]: from sklearn.feature_extraction.text import
CountVectorizer
   ...: cv = CountVectorizer()
   ...: X = cv.fit_transform(corpus).toarray()

In [3]: from sklearn.feature_extraction.text import
CountVectorizer
   ...: cv = CountVectorizer(max_features = 1500)
   ...: X = cv.fit_transform(corpus).toarray()

In [4]: y = dataset.iloc[:, 1].values

In [5]:
```

Permissions: RW End-of-lines: LF Encoding: ASCII Line: 53 Column: 38 Memory: 62 %


Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
naive_bayes.py
natural_language_processing.py*
cm - NumPy array
Format
Resize
Background color
Cancel
Variable explorer
Object inspector
Variable explorer
File explorer
IPython console
Console 1/A
0:09:00
0:00:56
10
30

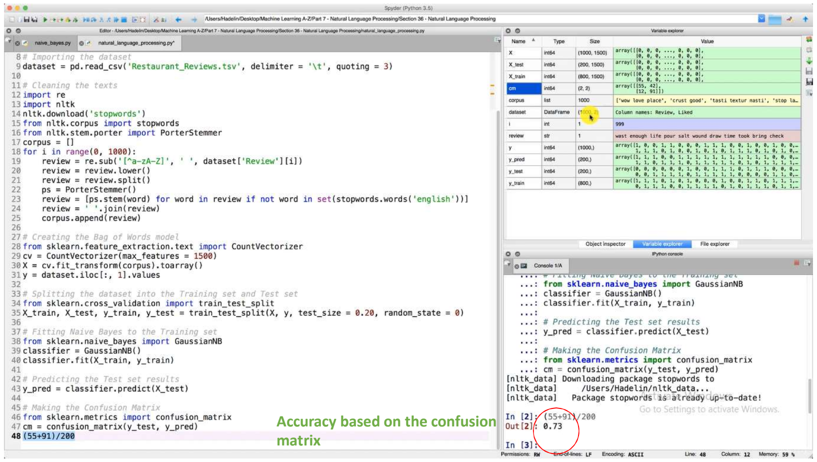
```python

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []
for i in range(0, 1000):
    review = re.sub('[^a-zA-Z]', ' ', dataset['Review']
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review if not wo
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVector
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values

# Splitting the dataset into the Training set and Test
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,

# Fitting Naive Bayes to the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

	0	1
0	55	42
1	12	91

Name	Type	Size	Value
X	int64	(1000, 1500)	array([[0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0],
X_test	int64	(200, 1500)	array([[0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0],
X_train	int64	(800, 1500)	array([[0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0],
cm	int64	(2, 2)	array([[55, 42], [12, 91]])
corpus	list	1000	['wow love place', 'crust good', 'tasti textur nasti', 'stop la…
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
	int	1	999
	str	1	wast enough life pour salt wound draw time took bring check
	int64	(1000,)	array([1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0,… 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0,…
	int64	(200,)	array([1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,… 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1,…
	int64	(200,)	array([0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0,… 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0,…
	int64	(800,)	array([1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1,… 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1,…

```
est_size = 0.20, random_state = 0)
   ...:
   ...: # Fitting Naive Bayes to the Training set
   ...: from sklearn.naive_bayes import GaussianNB
   ...: classifier = GaussianNB()
   ...: classifier.fit(X_train, y_train)
   ...:
   ...: # Predicting the Test set results
   ...: y_pred = classifier.predict(X_test)
   ...:
   ...: # Making the Confusion Matrix
   ...: from sklearn.metrics import confusion_matrix
   ...: cm = confusion_matrix(y_test, y_pred)
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

In [2]:
```

Permissions: RW End-of-lines: LF Encoding: ASCII Line: 47 Column: 38 Memory: 58 %


Spyder (Python 3.5)
/Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing
Editor - /Users/Hadelin/Desktop/Machine Learning A-Z/Part 7 - Natural Language Processing/Section 36 - Natural Language Processing/natural_language_processing.py
naive_bayes.py
natural_language_processing.py*
Variable explorer
Object inspector
Variable explorer
File explorer
IPython console
Console 1/A

```python
# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []
for i in range(0, 1000):
    review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
    review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values

# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)

# Fitting Naive Bayes to the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
(55+91)/200
```

Name	Type	Size	Value
X	int64	(1000, 1500)	array([[0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0],
X_test	int64	(200, 1500)	array([[0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0],
X_train	int64	(800, 1500)	array([[0, 0, 0, ..., 0, 0, 0], [0, 0, 0, ..., 0, 0, 0],
cm	int64	(2, 2)	array([[55, 42], [12, 91]])
corpus	list	1000	['wow love place', 'crust good', 'tasti textur nasti', 'stop la…
dataset	DataFrame	(1000, 2)	Column names: Review, Liked
i	int	1	999
review	str	1	wast enough life pour salt wound draw time took bring check
y	int64	(1000,)	array([1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0,… 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0,…
y_pred	int64	(200,)	array([1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,… 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1,…
y_test	int64	(200,)	array([0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0,… 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0,…
y_train	int64	(800,)	array([1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1,… 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1,…

```
   ...: from sklearn.naive_bayes import GaussianNB
   ...: classifier = GaussianNB()
   ...: classifier.fit(X_train, y_train)
   ...:
   ...: # Predicting the Test set results
   ...: y_pred = classifier.predict(X_test)
   ...:
   ...: # Making the Confusion Matrix
   ...: from sklearn.metrics import confusion_matrix
   ...: cm = confusion_matrix(y_test, y_pred)
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/Hadelin/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!

In [2]: (55+91)/200
Out[2]: 0.73

In [3]:
```

Permissions: RW End-of-lines: LF Encoding: ASCII Line: 48 Column: 12 Memory: 59 %

Activate Windows
Go to Settings to activate Windows.

Accuracy based on the confusion matrix

NLP Classification

- Text Cleaning
- Bag of Words or Vectorization
- Apply Classification Models
- Compare Models
  - Confusion Matrix
  - Accuracy