

# Tutorial Task

Name – Swati Rai

Reg No – 20BCE0996

Date – 13<sup>th</sup> Jan 2023

## Task 1

### # Task 1

```
In [ ]: #SWATI_20BCE0996
```

```
In [1]: def gender_features(word):  
        return {'last_letter': word[-1]}
```

```
In [2]: gender_features('Swati')
```

```
Out[2]: {'last_letter': 'i'}
```

```
In [3]: from nltk.corpus import names
```

```
In [6]: labeled_names = [(name, 'male') for name in names.words('male.txt')] + [(name, 'female') for name in names.words('female.txt')]
```

```
In [7]: import random
```

```
In [9]: random.shuffle(labeled_names)
```

```
In [10]: labeled_names
```

```
('Maren', 'female'),  
( 'Porter', 'male'),  
( 'Wilmette', 'female'),  
( 'Wyn', 'male'),  
( 'Tedra', 'female'),  
( 'Vijay', 'male'),  
( 'Tudor', 'male'),  
( 'Bethena', 'female'),  
( 'Smitty', 'male'),  
( 'Randee', 'female'),  
( 'Hiralal', 'male'),  
( 'Giffer', 'male'),  
( 'Haywood', 'male'),  
( 'Katti', 'female'),  
( 'Pail', 'male'),  
( 'Andie', 'female'),  
( 'Glynn', 'male'),  
( 'Hasty', 'male'),  
( 'Michalina', 'female'),  
( 'Helena', 'female')
```

```
In [11]: featuresets = [(gender_features(n), gender) for (n,gender) in labeled_names]
```

```
In [12]: featuresets
```

```
Out[12]: [(('last_letter': 'e'), 'female'),  
          (('last_letter': 'y'), 'male'),  
          (('last_letter': 'e'), 'female'),  
          (('last_letter': 'n'), 'male'),  
          (('last_letter': 'a'), 'female'),  
          (('last_letter': 'n'), 'female'),  
          (('last_letter': 'l'), 'male'),  
          (('last_letter': 'h'), 'female'),  
          (('last_letter': 'd'), 'male'),  
          (('last_letter': 'a'), 'female'),  
          (('last_letter': 'h'), 'male'),  
          (('last_letter': 'd'), 'male'),  
          (('last_letter': 'a'), 'female'),  
          (('last_letter': 'l'), 'male'),  
          (('last_letter': 'n'), 'male'),  
          (('last_letter': 'a'), 'female'),  
          (('last_letter': 's'), 'female'),  
          (('last_letter': 'n'), 'female'),  
          (('last_letter': 'a'), 'female')]
```

```
In [36]: train_set, test_set = featuresets[500:], featuresets[:500]
```

## Task 2

### # Task 2

```
In [37]: import nltk
```

```
In [41]: classifier = nltk.NaiveBayesClassifier.train(train_set)
```

```
In [47]: classifier.classify(gender_features('Swati'))
```

```
Out[47]: 'female'
```

```
In [ ]: print(nltk.classify.accuracy(classifier, test_set))
```

```
In [44]: import nltk
from nltk.tokenize import TweetTokenizer
text = 'The party was sooo fun :D #superfun'
twtkn = TweetTokenizer()
twtkn.tokenize(text)
```

```
Out[44]: ['The', 'party', 'was', 'sooo', 'fun', ':D', '#superfun']
```

```
In [ ]:
```












## Task 3

### Explore COCA

Demonstrate the following features

#### 3.1 Find the frequency count of a word

Word searched : Machine

 **Corpus of Contemporary American English**          

SEARCH

FREQUENCY

CONTEXT

ACCOUNT

List

Chart

Word Browse

Collocates

Compare KWIC


Machine

[POS]?

Find matching strings

Reset

☐ Sections ☐ Texts/Virtual ☐ Sort/Limit ☐ Options

 (HIDE HELP)

NO LICENSE

LIST display

Find single words like *mysterious*, all forms of a word like *JUMP*, words matching patterns like *\*break\**, phrases like *more \* than* or *rough NOUN*. You can also search by synonyms (e.g. *gorgeous*), and customized wordlists like *clothes*. In each case, you see each individual matching string.

More information: [basic syntax](#), [part of speech](#), [lemmas](#) (forms of words), [synonyms](#), [customized word lists](#), and [combining words](#).

Frequency:

Corpus of Contemporary American English

SEARCH FREQUENCY CONTEXT ACCOUNT

ON CLICK: [CONTEXT](#) [TRANSLATE \(??\)](#) [ENTIRE PAGE](#) [GOOGLE](#) [IMAGE](#) [PRON/VIDEO](#) [BOOK](#) (HELP)

HELP	1	★	ALL FORMS (SAMPLE): 100 200 500	FREQ
1	i	★	MACHINE	63558

0.238 seconds

### 3.2 Chart – word – frequency; section; sub-section;

Corpus of Contemporary American English

SEARCH CHART CONTEXT ACCOUNT

CLICK TO SEE CONTEXT

SECTION	ALL	BLOG	WEB	TV/M	SPOK	FIC	MAG	NEWS	ACAD	1990-94	1995-99	2000-04	2005-09	2010-14	2015-19
FREQ	63578	7686	8510	8994	4855	10950	11512	6117	4954	8160	8610	9087	7411	7067	7047
WORDS (M)	993	128.6	124.3	128.1	126.1	118.3	126.1	121.7	119.8	121.1	125.2	124.6	123.1	123.3	122.8
PER MIL	64.02	59.76	68.49	70.22	38.49	92.54	91.30	50.25	41.36	67.38	68.76	72.92	60.23	57.29	57.41

SEE ALL SUB-SECTIONS AT ONCE

### 3.3 Collocate – Display three collocations on the left and right each of the word

Set to 3 collocations

Corpus of Contemporary American English

SEARCH CHART CONTEXT ACCOUNT

List Chart Word Browse **Collocates** Compare KWIC -

Machine Word/phrase [POS]?  
 Collocates [POS]  
 + 4 3 2 1 0 0 1 2 3 4 +  
 Find collocates Reset

☐ Sections Texts/Virtual Sort/Limit Options

(HIDE HELP) NO LICENSE

**COLLOCATES display**

To find collocates in COCA, you would normally input a word via **Word**, and then select Collocates on the next page. In COCA, the collocates display (via **Word**) is much better than with the other corpora from English-Corpora.org, such as automatically grouping collocates by part of speech. Some examples: *bread*, *kiss (v)*, *rough*, or *naturally*.

The only time that you'd want to use the form to the left is when you want to find collocates for a string of words (e.g. *put away* or *fire station*), or when you absolutely need to limit the number of words left or right.

Note however that using the form to the left for individual words (especially high frequency words) will be *much* slower, and in many cases the search will "time out", resulting in an error. In nearly all cases, it is much better to search for collocates via **Word**.

See what words occur near other words, which provides great insight into



Corpus of Contemporary American English

SEARCH

FREQUENCY

CONTEXT

ACCOUNT

76	NO	602	
77	COULD	592	
78	SHE	585	
79	'	574	
80	ITS	571	
81	USE	571	
82	DO	562	
83	OFF	506	
84	ALL	505	
85	:	496	
86	THEN	491	
87	THROUGH	489	
88	GET	465	
89	USING	445	
90	ABOUT	422	
91	BIG	422	
92	SOME	420	
93	OUR	416	
94	USED	408	
95	MAN	405	
96	AGAINST	404	
97	GOT	403	
98	THAN	403	
99	POLITICAL	390	
100	MORE	389	
TOTAL		228815	

SEE MORE THAN 100 WORDS/PHRASES

0.930 seconds