Practical NLTK

David Arroyo Menéndez

October 28, 2021

Practical NLTK: Summary

```
$ git clone https://github.com/davidam/python-examples.git
```

1. Sentiment Analysis
2. Detect Gender
3. Sentence Similarity
4. Text Summary
5. Classify Documents
6. Manage Words

singulars/plurals, dictionary entries, stopwords

1. Gramatical Trees
2. Extract Keywords
3. Disambiguation
4. Bots

Sentiment Analysis

Tweets positives versus negatives

```
$ python3 tweet-sentyment.py
$ python3 sentiment-analysis-vader.py
$ python3 sentiment.py
```

Detect Gender

Your name is your gender

```
$ python3 sexmachine.py
$ python3 perceval_git_counter_sexmachine.py
$ python3 perceval_mbox_sexmachine.py
```

Sentence Similarity

Sentence Similarity

```
$ python3 sentence-similarity.py
$ nosetests3 test/test_sentencesimilarity.py
```

Classify Documents

Classify Documents

```
$ python3 doc-classification-ch06.py
```

Classify Newsgroups

Classify Newsgroups

```
$ python3 nltk-sklearn.py
```

Synonims and Antonims

Synonims and Antonims

```
$ python3 synonims-antonims.py
```

Singulars and Plurals

```
$ python3 stem.py
$ nosetests3 test/test_stem.py
```

Sillables

Sillables

```
$ python3 silaba.py
```

Stopwords

For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on. In this case, stop words can cause problems when searching for phrases that include them

Singulars and Plurals

```
$ python3 stopwords.py
$ nosetests3 test/test_stopwords.py
```

Lemmas: dictionary entries

```
$ python3 wordnet-lemmatizer.py
$ nosetests3 test/test_wordnet.py:TddInPythonExample.test_syno
```

Trees (I)

```
I can build a gramatic or semantic tree from a sentence
$ python3 semantic-tree.py
I can generate sentences from a gramatic
$ python3 howtos/generate.py
I can visualize a gramatic
$ python3 parse-tree.py
I can obtain bigrams, trigrams or ngrams
$ python3 bigrams-trigrams.py
$ nosetests3 test/test_bigrams_trigrams.py
```

Trees (II)

```
I can print a tree from sintactic pairs
$ python3 code-chinker.py
I can build sintactic pairs from a sentence and print a tree
$ python3 tokenizeandtag.py
$ python3 ner.py
I can find sentences verb to verb with trigrams
$ python3 code_three_word_phrase.py
```

Corpus

Corpus

```
$ python3 gutenberg.py
$ nosetests3 test/test_gutenberg.py
$ python3 inaugural.py
$ python3 corpus-howto-new-corpus.py
```

Keywords: rake algorithm

```
$ python3 nltk-rake.py
```

Disambiguation

Disambiguation

```
$ python test_all_words_wsd.py
$ python test_wsd.py

# Remember synset
$ python3 wordnet-example.py
$ nosetests3 test/test_wordnet.py
```

Sharing data models: pickle

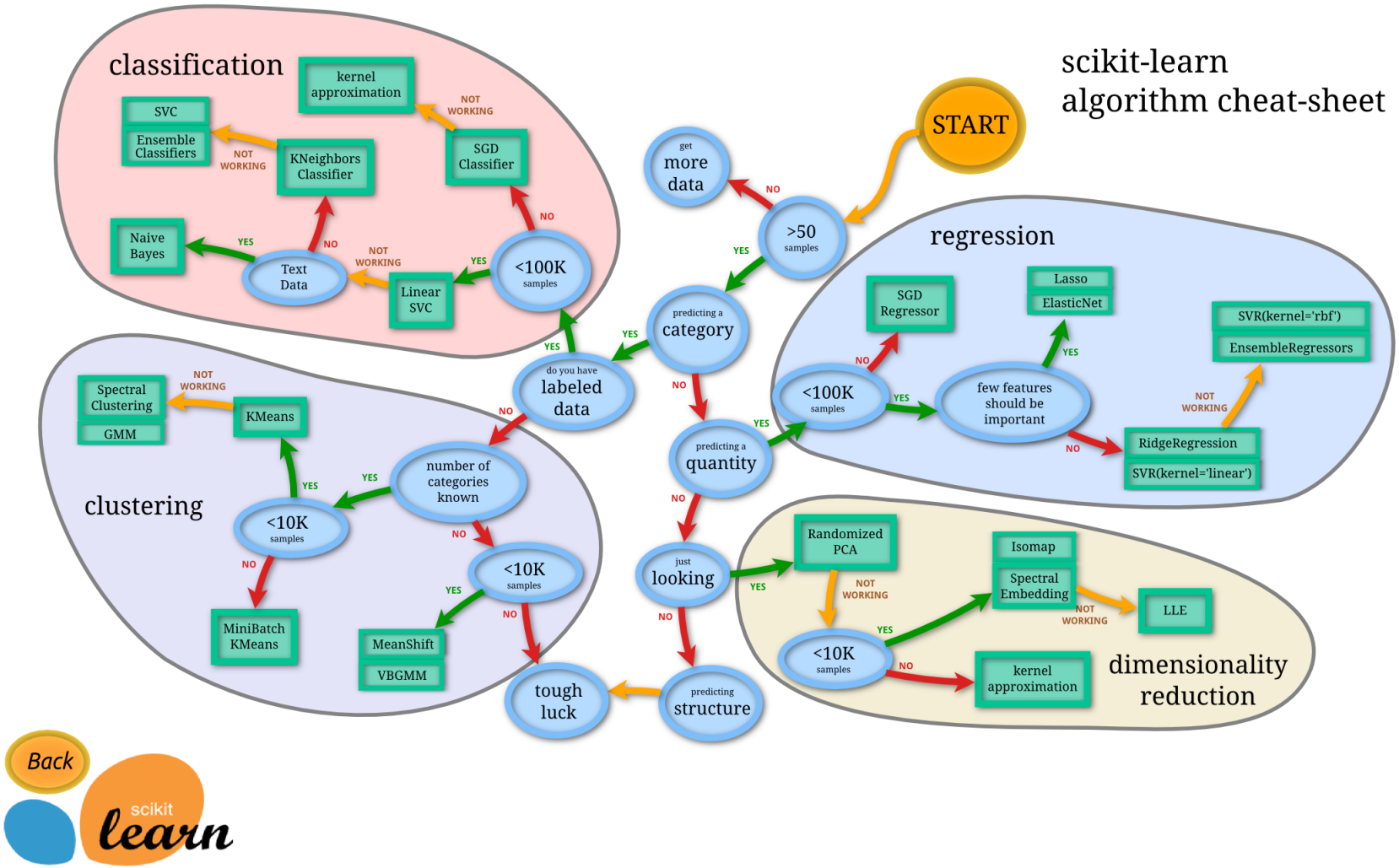
```
$ python3 nltk-pickle.py
```

Bots

Bots

```
$ python3 eliza.py
$ python3 chatbot.py
```

Using Scikit in NLTK (I)



Using Scikit in NLTK (II)

Using Scikit in NLTK (II)

```
$ pip3 install damenltk
$ cd damenltk/damenltk
$ nosetest3 test/test_svc.py
$ nosetest3 test/test_bernoulli.py
```

References

- Python Tutorial (Guido Van Rossum)
- El Tutorial de Python por Guido Van Rossum
- Natural Language Processing with Python

(Steven Bird, Ewan Klein, and Edward Loper)

License