NLTK, Scikit y Damegender como ejemplo de aplicación

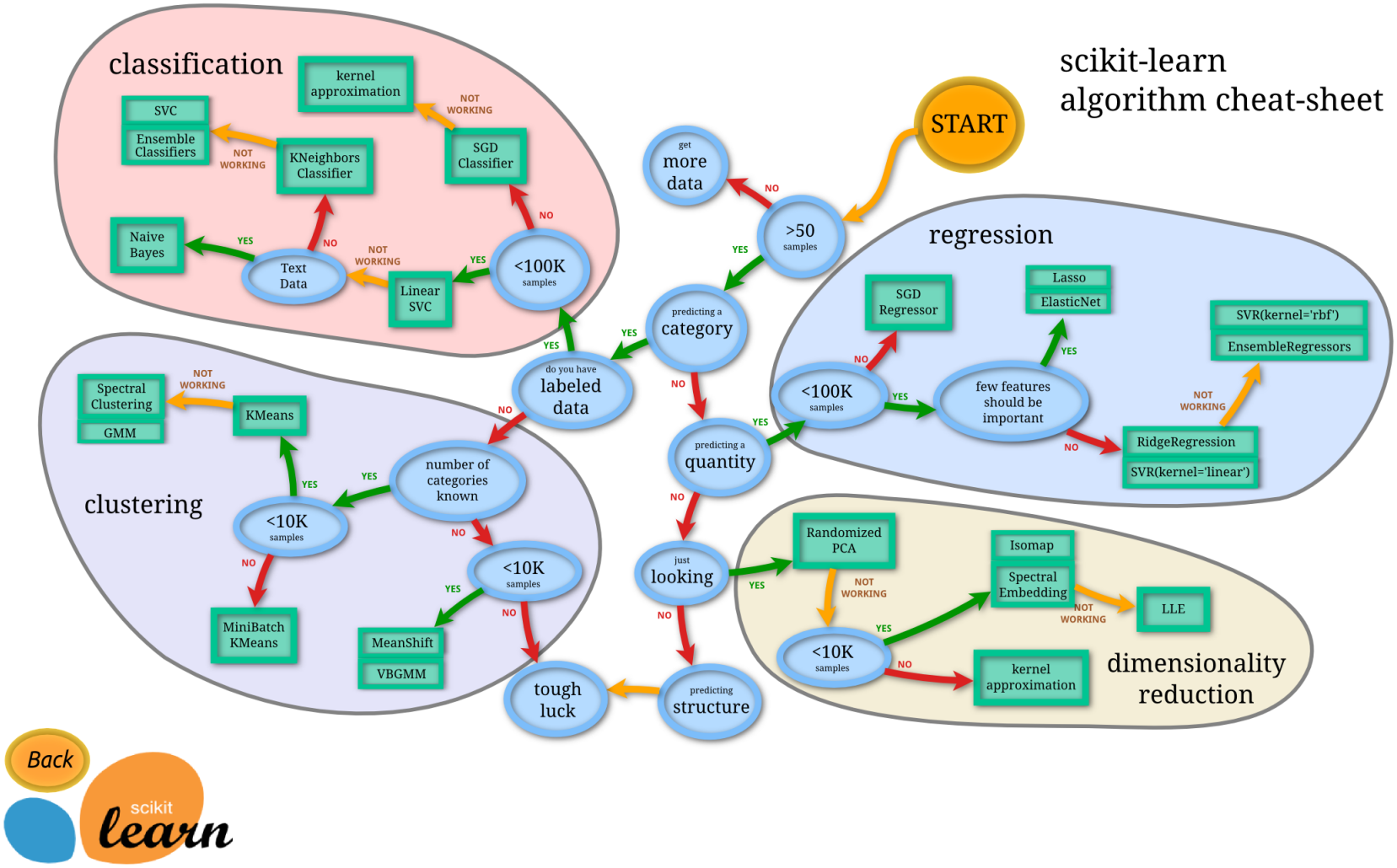
David Arroyo Menéndez

March 26, 2019

Machine Learning is for predictions

- Quantities
- Categories (with data training or not)
- What properties is predicting

Scikit Graph



In classification our data are properties and a variable for the prediction

	Predictors				Response
	Outlook	**Temperature**	**Humidity**	**Wind**	**Class** **Play=Yes** **Play=No**
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

Practical NLTK

```
$ git clone https://github.com/davidam/GAPLEN.git
$ sudo pip3 install GAPLEN
```

In NLP the machine learning is a classification task:

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

Classification: SGD

We want predict a category and we've labeled data with <100k samples

```
$ python3 plot_sgd_iris.py
$ python3 sgd.py
```

Classification: Kernel Approximation

We want predict a category and we've labeled data with <100k samples and SGD is not working

```
$ python3 kernel-approximation.py
```

Classification: Naive Bayes Examples

```
$ python3 gaussiannb.py
$ python3 bernoullinb.py
$ python3 multinomialnb.py
$ python3 nltk/sexmachine.py
```

Sentiment Analysis

Tweets positives versus negatives

```
$ python3 tweet-sentyment.py
$ python3 tweepy-example.py
```

Sentence Similarity

Your name is your gender

```
$ 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

Singulars and Plurals

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

Corpus

Corpus

```
$ python3 gutenberg.py
$ nosetests3 test/test_gutenberg.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
```

Detect Gender

Detect your gender from your name with nltk

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

Execute damegender program

```
# Detect gender from a name (INE is the dataset used by default
$ python3 main.py David
David gender is male
363559  males for David from INE.es
0 females for David from INE.es

# Detect gender from a name from multiple dataset
$ python3 main.py David --total="all"
David gender is male
375099 males and 9 females from all census (INE + Uk census + U

# Detect gender from a name only using machine learning (experi
$ python3 main.py Mesa --ml=nltk
Mesa gender is female
0 males for Mesa from INE.es
0 females for Mesa from INE.es
```

Counting males and females in mailing lists and repositories

```
# Count gender from a git repository
$ python3 git2gender.py https://github.com/chaoss/grimoirelab-p
The number of males sending commits is 15
The number of females sending commits is 7

# Count gender from a mailing list
$ cd files/mbox
$ wget -c http://mail-archives.apache.org/mod_mbox/httpd-annour
$ cd ..
$ python3 mail2gender.py http://mail-archives.apache.org/mod_mb
```

Use an api to detect gender

```
# Use an api to detect the gender
$ python3 api2gender.py Leticia --surname="Martin" --api=namsor
female
scale: 0.99

# Google popularity for a name
$ python3 gendergoogle.py Leticia
Google results of Leticia as male: 42300
Google results of Leticia as female: 63400
```

Informative Features

```
# Give me informative features
$ python3 infofeatures.py
Females with last letter a: 0.4705246078961601
Males with last letter a: 0.04867256637168141 5
Females with last letter consonant: 0.2735841767750908
Males with last letter consonant: 0.6355328972681801
Females with last letter vocal: 0.7262612995441552
Males with last letter vocal: 0.3640823393612928
```

To measure success

```
$ python3 accuracy.py --csv=files/names/min.csv
#################### NLTK!!
Gender list: [1, 1, 1, 1, 2, 1, 0, 0]
Guess list:  [1, 1, 1, 1, 0, 1, 0, 0]
Dame Gender accuracy: 0.875

$ python3 accuracy.py --api="genderize" --csv=files/names/min.c
#################### Genderize!!
Gender list: [1, 1, 1, 1, 2, 1, 0, 0]
Guess list:  [1, 1, 1, 1, 2, 1, 0, 0]
Genderize accuracy: 1
```

Where is the confusion? Guessing males or females?

```
$ python3 confusion.py
A confusion matrix C is such that Ci,j is equal to the number o
If the classifier is nice, the diagonal is high because there a
Namsor confusion matrix:
 [[2 0 0]
 [0 5 0]
 [0 1 0]]
Genderize confusion matrix:
 [[2 0 0]
 [0 5 0]
 [0 0 1]]
Gender Guesser confusion matrix:
 [[2 0 0]
 [0 5 0]
 [0 1 0]]
Sexmachine confusion matrix:
 [[2 0 0]
```

To analyze errors guessing names from a csv

```
$ python3 errors.py --csv="files/names/all.csv" --api="gendergu
Gender Guesser with files/names/all.csv has:
+ The error code: 0.22564457518601835
+ The error code without na: 0.026539047204698716
+ The na coded: 0.2045336563419276 6
+ The error gender bias: 0.0026103980857080703
```

Support Python and Libremanuals

El Tutorial de Python por Guido Van Rossum