Data Mining using Python
— course introduction

Finn Årup Nielsen

DTU Compute
Technical University of Denmark

September 1, 2014
Data Mining using Python

DTU course 02819 Data mining using Python.

Previously called DTU course 02820 Python programming (study administration wanted another name).

Project course with a few introductory lectures, but mostly self-taught.

Deliverables: A report, a poster and an oral presentation at the poster about a Python program you write in a group.

Teacher: Finn Årup Nielsen
Tentative schedule for autumn 2014

1. September Installation

8. September. Introduction to the Python language.


22. September. Databasing, web and text processing, “natural language processing”

29. September Misc., e.g., GUI, Web serving

Project work for the rest of the time

December: Exam and report hand-in

See links to PDF on http://www.compute.dtu.dk/courses/02820/
Other courses

Introductory programming and mathematical modelling (linear algebra, statistics, machine learning)

Some overlap with 02805 (Social graphs and interaction), 02806 Social data analysis and visualization, 02821 (Web og social interaktion) and 02822 (Social data modelling).

If you take several 028xx courses be sure that you do not make a project that overlaps with projects in these courses in any way!
Project

Project: (Idea), design, implementation, testing, documentation.

Performed preferably in groups of two persons. Three is also ok.

Should preferably contain components of:

- Mathematical (numerical, computational, statistical or machine learning) modeling

- Internet/data/text mining
Poster

Construct a poster. Often A0/A1-sized.

“Defend” the poster, i.e., give a relatively short oral presentation of the poster and answer questions: Usually a ten minutes presentation for a two-person group with some questions afterwards.

Inspired from DTU course 02459 Machine Learning for Signal Processing
Why Python?

Interpreted, readable (usually clearer than Perl), interactive, many libraries, runs on many platforms, e.g., Nokia smartphones (hmmm...) and Apache Web servers.

With Python one can construct numerical programs, though with a bit more boilerplate than Matlab.

Google and Yahoo! is (has been?) using it. 2.73% of Open Source code written in Python (Black Duck Software, 2009).

“Without [Python] a project the size of Star Wars: Episode II would have been very difficult to pull off.” — http://python.org/about/quotes/

**XKCD 353**: “I wrote 20 short programs in Python yesterday. It was wonderful. Perl I’m leaving you.”
Data Mining using Python — course introduction

Interactive data analysis
Matlab

Python

PHP
Web programming
Javascript

Perl
String processing
Why Python? Interactive language!

Interactive session

$ python
Python 2.4.4 (#2, Oct 22 2008, 19:52:44)
[GCC 4.1.2 20061115 (prerelease) (Debian 4.1.1-21)] on linux2
Type "help", "copyright", "credits" or "license" for more information.

>>> 1+1
2

However, Matlab-like computation is not straightforward, e.g., what is the result of

>>> 1/2
Why Python? Interactive language!

Interactive session

```shell
$ python
Python 2.4.4 (#2, Oct 22 2008, 19:52:44)
[GCC 4.1.2 20061115 (prerelease) (Debian 4.1.1-21)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 1+1
2
```

However, Matlab-like computation is not straightforward, e.g., what is the result of

```python
>>> 1/2
0 # Integer division! (in Python2 --- not Python3)
```
Example projects for inspiration

1. Characterize external links from DTU’s Web-site.

2. Characterize internal link structure on DTU.


4. Sentiment analysis of Tweets, blogs or news articles.

5. A wiki-based database for brain activations.


7. Suggest one yourself
How do we evaluate the project?

Possible dimensions for evaluation of project?
Coding style

Bad: Variables are given incoherent names. Indentations are inconsistent.

Good: Variables are given intuitive and readable names. Code has been checked with flake8 and pylint.
Evaluation: Reusability of code

Bad: Input variable values are hard-coded. Code is repeated to make it look ‘big’.

Good: Code is in meaningful modules. It is no problem to apply the data mining on new data. Part of the code can be used in other contexts.
Evaluation: Amount of data

Bad: The system is only able to handle a small amount of prespecified data and not likely anything else.

Good: The system use a ‘large’ amount data. The system use a database or other structured way of accessing a large amount of data.
Evaluation: Data mining effort

Bad: Simple analysis is performed. No use of Numpy, Scipy or other data mining package. Data is just entered, stored and ‘copied around’.

Good: Machine learning or other complex analysis is performed.
Evaluation: Testing

Bad: No tests.

Good: A part of the code is tested.

Better: As much as feasible of the code is tested and with a variety of input and with the standard tools of Python testing. Testing coverage is computed and reported. Testing is performed on multiple versions of Python.
Evaluation: Documentation

Bad: There is no documentation. No use of docstring.

Good: Docstrings are used.

Better: Docstrings are used and used according to Numpy and other conventions. The documentation is checked with the pep257 program and no errors are found. Online documentation is generated with sphinx is available.
Evaluation: ‘Well-presented’ results

Bad: A plot in Excel is used with unlabeled axes.

Good: Data analysis results and other presentation with a number of Python tools, Matplotlib, etc., utilized in depth.

Better: A responsive interactive environment (perhaps web-based) is made where the user can navigate the result such zooming and panning as well as get the data results in a suitable format for further processing.
Evaluation: other dimensions

Effective and ‘good’ code. Shows a good command of Python . . .

Amount of code (but not code that is constructed to look big, by unnecessary repetitions and bad implementation).

Relevance of project: Is there a interesting (scientific) result or possibility for commercial application?

Originality of project . . . !?
More information

Learning objective: “Identify relevant learning material”. You yourself need to identify the appropriate Python documentation!

http://www.python.org/

*The Python Tutorial* http://docs.python.org/tutorial/

Internet search engines: Google, Bing or Yahoo.

Stack Overflow, ... 

Google for error messages, “Python tutorial”

*MATLAB commands in numerical Python (NumPy)* by Vidar Bronken Gundersen if you know Matlab or R.
Free books

*Dive into Python*, (Pilgrim, 2004). Free, old and good. With `sudo aptitude install diveintopython` it is available at file:///usr/share/doc/diveintopython/html/index.html

*Think Python: How to Think Like a Computer Scientist* and *How to Think Like a Computer Scientist*. Covers the basics of the Python language and Tkinter GUI. Also available as Wikibooks: *Think Python* and *How to Think Like a Computer Scientist: Learning with Python 2nd Edition*. 

Finn Årup Nielsen 21 September 1, 2014
General books

*Practical Programming. An introduction to computer science using Python*, (Campbell et al., 2009): Introductory programming. Good if you are unsure.

*Python cookbook* (Martelli et al., 2005): Short program examples for somewhat specific problems. Too specific.
Specialized books relevant for the course


*Natural language processing with Python* (Bird et al., 2009): Text mining with Python. On paper and available online from http://nltk.org

*Programming the Semantic Web* (Segaran et al., 2009)


Data analysis and numerics books

Kevin Sheppard’s *Introduction to Python for Econometrics, Statistics and Data Analysis* on 381 pages covers both Python basics and Python-based data analysis with Numpy, SciPy, Matplotlib and Pandas, — and it is not just relevant for econometrics (Sheppard, 2014).

(Langtangen, 2005; Langtangen, 2008): Python book with many examples especially for numerical processing. 2005 edition not fully up to date on numerical Python. 2008 version should be available online through DTU library

My draft *Data Mining with Python*.
Other books

Other O'Reilly titles: Python in a Nutshell, Python Pocket Reference, Learning Python, Programming Python?

Other books that I know of:


*Python Essential References* (*Beazley, 2000*): Introduction and list of Python functions with small examples. Somewhat old and not recommendable.
Example: A fielded wiki ...

Web script in Python implementing a fielded wiki for personality genetics.

Persistence with a small SQLite database.

Some of the Python libraries used: cgi, Cookie, math, pysqlite2, scipy, sha.

One Python script with 2269 lines of code.
Example: ... A fielded wiki

Computation of effect sizes (a statistical value) and comparison to statistical distributions.

Generation of interactive and hyperlinked plots in SVG (an XML format)
Structured information from Wikipedia

Get Wikipedia pages that contain a specific template, download the page, extract information from the templates and render the result on an HTML page.

Python libraries: json, re, urlib2

Around 25 Python lines to get the data, and around 120 to render the result.
Web script for Twitter annotation

CGI program that searches Twitter with a user-defined query, obtain tweets and present them in a Web form for manual annotation and stores the result in a SQL database.

Python libraries: codecs, json, re, cgi, urllib2, pysqlite2, xml.

500 Python lines.
Temporal sentiment analysis

Data Mining using Python — course introduction

DTU-forsker afkoder Twitter-beskeder med 1.200 linjer Python-kode

Twitter-beskeder og blog-indlæg har stor betydning for, hvordan virksomheder omdanner sig ud online. Danske forskere arbejder på at skabe et digitalt stemningsbarometer ud fra syndflogen af oplysninger online.

Download tweets from Twitter microblog searching on 'COP15' (United Nation climate conference in December 2009)

Compare words against a word list with valence (positive/negative) valence for each word.

Sum up positive and negative valence for each day and plot a graph.

Python libraries: SQLite, re, simplejson, ...
Online topic-sentiment mining

http://neuro.imm.dtu.dk/cgi-bin/brede_str_nmf
References


