Discussion

Spyros Samothrakis
Research Fellow, IADS
University of Essex

March 20, 2017

Last lecture

- Revision lecture based on Lecture one
- We will go again through what we have discussed already, and put everything into context
- The Beginning Is the End Is the Beginning....

Better science through data

Hey, Tony, Stewart Tansley, and Kristin M. Tolle. "Jim Gray on eScience: a transformed scientific method." (2009).

- Thousand years ago: empirical branch
  - You observed stuff and you wrote down about it
- Last few hundred years: theoretical branch
  - Equations of gravity, equations of electromagnetism
- Last few decades: computational branch
  - Modelling at the micro level, observing at the macro level
- Today: data exploration
  - Let machines create models using vast amounts of data

Mixing statistics, philosophy of science and machine learning

- Wu, C. F. J. "Statistics= data science." (1997).
- Breiman, Leo. "Statistical modeling: The two cultures (with comments and a rejoinder by the author)." Statistical Science 16.3 (2001): 199-231.
- Science is the epistemology of causation
- Data science is basically science on arbitrary data
  - But quite often we only care about predictions
- Possibly a re-branding of data mining, machine learning, artificial intelligence, statistics

Better business through data

- There was a report by Mckinsey

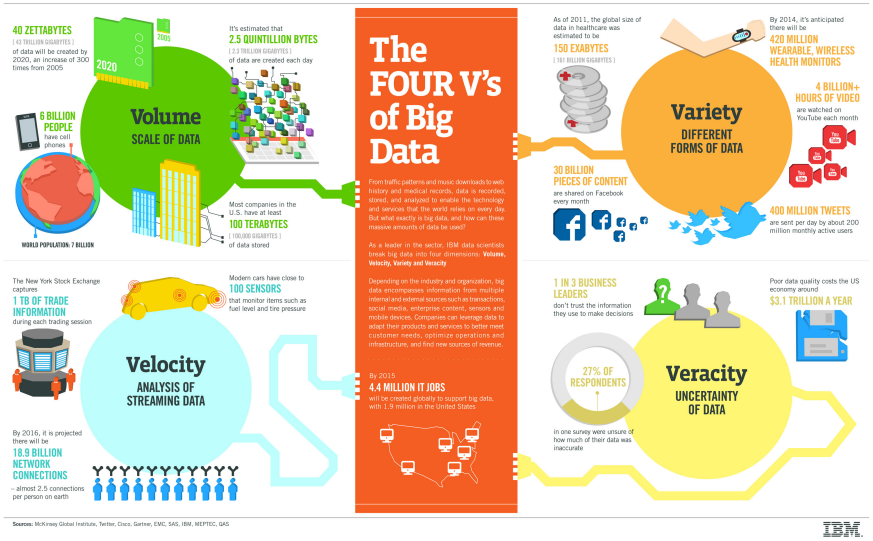
Manyika, James, et al. "Big data: The next frontier for innovation, competition, and productivity." (2011).

- Urges everyone to monetise "Big Data"
- Use the data provided within your organisation to gain insights
- Has some numbers as to how much this is worth
- Proposes a number of methods, most of them associated with machine learning and databases

More is different

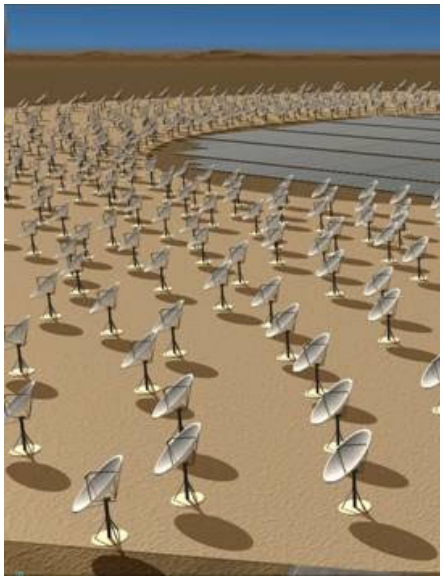
- Anderson, Philip W. "More is different." Science 177.4047 (1972): 393-396.
- The idea of emergence
- You put stuff together, you go from physics to chemistry
- . . . from chemistry to biology
- . . . from biology to psychology and zoology
- . . . from psychology to sociology
- "quantity changes into quality"

IBM's Infographic



Classic science

- The original data science field
- SKA (The Square Kilometer Array) ~ 4.6 EB expected (i.e. 4.6e+6 TB), (Zhang, Yanxia, and Yongheng Zhao. "Astronomy in the Big Data Era." Data Science Journal 14 (2015).)[1]
- Bioinformatics
- Medical science



[1] http://datascience.codata.org/article/10.5334/dsj-2015-011

Recommender Systems

- One of the most popular applications of data science
- Propose products to customers based on past history
- Almost all online vendors do it
- Made popular by the Netflix prize

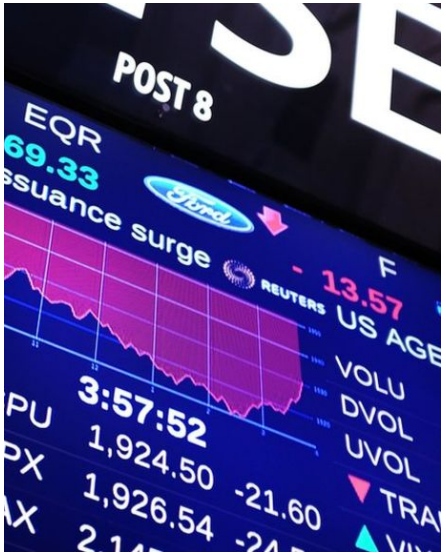
Data journalism

- Wikileak style data dumps are everywhere
- The Ashley-Madison Affair, 2015
- "Just three in every 10,000 female accounts on infidelity website are real"
- "The website claims 5.5 million of its 37 million accounts are 'female' "

[2] http://www.independent.co.uk/life-style/gadgets-and-tech/news/ashley-madison-hack-just-three-in-every-10000-female-accounts-on-infidelity-website-are-real-10475310.html

Finance & Insurance

- Predict stock prices (Hedge Funds)
- Insurance models
- Credit score
- In fact, a lot of trading that currently happens is algorithmic trading[2]
- Sudden drops in share prices often caused by defective algorithms



[3] http://www.bbc.com/news/business-34264380

Politics (current)

"...This included a) integrating data from social media, online advertising, websites, apps, canvassing, direct mail, polls, online fundraising, activist feedback, and some new things we tried such as a new way to do polling (about which I will write another time) and b) having experts in physics and machine learning do proper data science in the way only they can – i.e. far beyond the normal skills applied in political campaigns..."

Dominic Cummings's (Head of *Vote Leave*) Blog[3]

[4] https://dominiccummings.wordpress.com/2016/10/29/on-the-referendum-20-the-campaign-physics-and-data-science-vote-leaves-voter-intention-collection-system-vics-now-available-for-all/

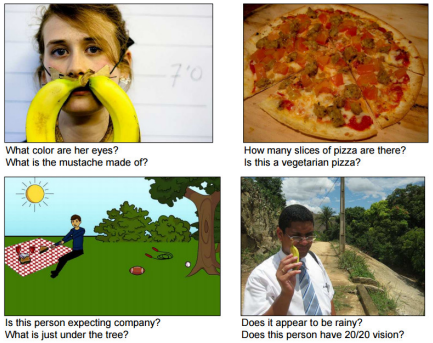
Politics (historical)

- New Yorker - THE PLANNING MACHINE: Project Cybersyn and the origins of the Big Data nation[4]
- Cybersyn / Chile during Alliente's rule, co-designed by Stafford Beer
- Plan was to use data fed directly from each industry to automate production

[5] http://www.newyorker.com/magazine/2014/10/13/planning-machine

Question answering

- e.g. Antol, Stanislaw, et al. "VQA: Visual question answering." Proceedings of the IEEE International Conference on Computer Vision. 2015.[5]
- Input can be videos, websites, et
- Think google



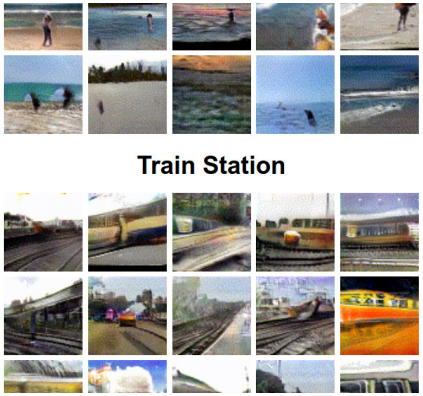
[6] http://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Antol_VQA_Visual_Question_ICCV_2015_paper.pdf

Digital Marketing

- Is a new product I just created well received by our customers?
- Is a new marketing campaign e-mail sent detrimental to our efforts?
- What is the content a chain of e-mails should have?
- Customer segmentation
- What adverts should I present to a user?

Creative artificial intelligence (recipes, music, art, text)

- e.g. Vondrick, Carl, Hamed Pirsiavash, and Antonio Torralba. "Generating videos with scene dynamics." Advances In Neural Information Processing Systems. 2016.[6]
- Generate an artefact
  - Generate videos
  - Generate text
  - Generate music



[6] http://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Antol_VQA_Visual_Question_ICCV_2015_paper.pdf

Game playing

- We recently have seen a resurgence of game playing machines
- A computer GO programme finally outperformed top humans (AlphaGO)
- No-limit heads up poker (matches still played as we speak!)
- New labs are opening from major game companies dealing with game AI
- Though directly related, game analytics

Artificial intelligence

- Everything we have seen so far are basically applications of Artificial Intelligence and Machine Learning
- Inductive reasoning from a limited amount of examples
  - Structured learning
  - One-shot models
- Deductive reasoning
  - From concepts to data
  - Platonic forms

Some sample data

- takes_off_road: owner takes the vehicle off road
- company_vehicle: it belongs to a business
- is_over_30: age of vehicle is over 30
- regular_service: is the vehicle serviced regularly?
- brake_down: will it break down within three months of our inspection date?

takes_off_road	company_vehicle	is_over_30	regular_service	brake_down
0	1	1	0	1
0	0	1	1	0
1	1	1	1	1
0	1	1	0	1
0	0	1	0	0
0	1	0	0	0
1	0	0	1	0
1	1	1	1	1
1	0	0	1	1
0	1	1	0	1
1	0	0	1	0
1	1	0	0	0
0	0	0	0	0

Predictions

- The most common data science operation
- Can you predict if a car will break down given the data, and if yes with what probability?
- Can you learn a model, that if provided with a tuple $<$ *takes_off_road*, *company_vehicle*, *is_over_30*, *regular_service* $>$ predict *break_down*?
- The tuple represents a vehicle
- Columns are called *features*
- If we call the model M, can you learn $P(C|D;M)$
- You might have seen this as *supervised learning*
- You can also try to predict if a vehicle was taken off-road, given that it broke down

Clustering

- Another very common request
- Imagine there is some hidden property in the data, another feature that we have not observed
  - This feature groups together vehicles
  - Again we are looking for $P(C|D;M)$, but C is a fictional/latent variable
- Unsupervised learning
- The probabilistic intuition I provided is not unique

Inferring what-if scenarios from the data

- Say your vehicle broke down
- What would have happened if you have not driven if off-road?
- Have a look at the data - what can you say?
- Do you have enough data of the needed type?
- Causality from observational data
  - Super hard, but super important
  - Think of smoking!

Acquiring new data

- We can't really answer what would happen to vehicle from the data collected already
- We might need to set a controlled experiment where:
  - We find vehicles of similar characteristics
  - Drive them off-road
  - See if they break down
  - What is the optimal way of doing such a procedure?
- Causality from experimental data - mostly what science is all about
  - **Science is the epistemology of causality**

Anomaly Detection

- If we are given a new vehicle, can we say if it is "special" in a way?
- Maybe it's the only vehicle with certain features
- Maybe it's a unique vehicle
- Somehow we need to find bizarre samples that do not conform to expect norm
- Multiple formal definitions

Generate new data

- Can I generate fictional vehicles and their properties?
- Mathematically, learn P(D;M), a model of the data
- You can then use your plausible, but fictional vehicles for entertainment
- "Learning to draw before learning to see"
  - $P(D, C; M) = P(C|D)P(D)$
  - $P(D|C; M)$

Dimensionality reduction

- Maybe we only need some feature combination above
- Maybe some features only carry noise with them - they are irrelevant
- For example, how important the *car_colour* feature would be?
- What happens if we learn based on irrelevant features?
- Spurious correlations are everywhere
- Kicking out useless features might make the model more interpretable

Linking with other data/collecting labels

- What if the data we have is not enough?
- In our example, model make is not provided
- Can we inquire data providers to find that?
- How expensive would that be?
- How easy is to label the data?
  - Active learning
  - Labelled data often very expensive

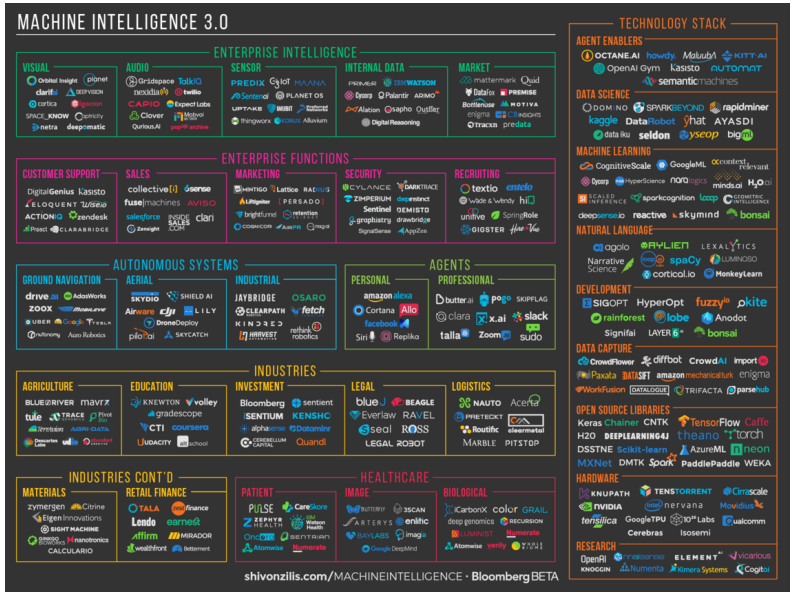
Making decisions from data

- Now that we have a model
- Let's say you know that a vehicle will break down after three months with a certain probability
  - How much do we charge for insurance on it?
  - Should we even sell insurance to the owner?
  - What is the risk of actually selling insurance?
- We are missing another model (that of the customer)
  - Do we actually need the model?
  - Do customer preferences change over time?
- Bandits, reinforcement learning

Some notes

- *"If you torture the data enough, nature will always confess."*
  - *Disputed*
- *"If you torture the data long enough, it will confess to anything."*
  - Huff, D. "How to lie with statistics (illust. I. Geis)." NY: Norton (1954).
- *Lies, damned lies, and statistics*
  - *Disputed*

Startup mayhem



The law

"We summarize the potential impact that the European Union's new General Data Protection Regulation will have on the routine use of machine learning algorithms. Slated to take effect as law across the EU in 2018, it will restrict automated individual decision-making (that is, algorithms that make decisions based on user-level predictors) which"significantly affect" users. The law will also effectively create a **right to explanation**, whereby a user can ask for an explanation of an algorithmic decision that was made about them. We argue that while this law will pose large challenges for industry, it highlights opportunities for computer scientists to take the lead in designing algorithms and evaluation frameworks which avoid discrimination and enable explanation"

Goodman, Bryce, and Seth Flaxman. "European Union regulations on algorithmic decision-making and a" right to explanation"." arXiv preprint arXiv:1606.08813 (2016).

The social impact of AI/Machine Learning

"We examine how susceptible jobs are to computerisation. To assess this, we begin by implementing a novel methodology to estimate the probability of computerisation for 702 detailed occupations, using a Gaussian process classifier. Based on these estimates, we examine expected impacts of future computerisation on US labour market outcomes, with the primary objective of analysing the number of jobs at risk and the relationship between an occupation's probability of computerisation, wages and educational attainment. According to our estimates, about 47 percent of total US employment is at risk. We further provide evidence that wages and educational attainment exhibit a strong negative relationship with an occupation's probability of computerisation"

- Not sure I believe them, but read the article

Frey, Carl Benedikt, and Michael A. Osborne. "The future of employment: how susceptible are jobs to computerisation." Technological Forecasting and Social Change (2014).

Overall on Data and Society

- Think about how much of your life you spend online
  - Not just on a computer, but mobile phones, GPS signals etc., car sensors
  - Soon your fridge and coffee machine (IoT)
- Tons of data flying around
  - They are being used to make decisions on a micro level (i.e. about you)
- Regulations are set in place
- New El-Dorado?

Linux VM

- Download the VM for this module
- External link `https://docs.google.com/uc?id=0B_kDfEzMuWD6ZGJFU1VfeEY3TnM&export=download`
- The VM contains all (or most) of what you need if you are to create a successful python project
- Username/password is `mlvm/mlvm`
- You will have a USB stick were you should copy the VM folder (after you un-rar the archive)
- More about this on the labs

Python

- Python is the language of this module
- You are expected to be competent python programmers (or willing to put the extra effort)
- Python has evolved to be one of the two "data science" languages (the other is **R**)
- Python has/is:
  - An excellent list of features coming from functional programming
  - A huge number of related libraries
  - Easy to learn
  - Object oriented programming capabilities
  - Can be extended via *C* trivially
  - A massive amount of related libraries

IPython/Jupiter

- A better command line interface to python
- Has something called a "notebook"
  - A notebook combines code + natural language
- See here for a very nice example

PyCharm shortcuts

- Double shift - meta-shortcut!

PyCharm Default Keymap

Editing

Ctrl + Space	Basic code completion (the name of any class, method or variable)
Ctrl + Alt + Space	Class name completion (the name of any project class independently of current imports)
Ctrl + Shift + Enter	Complete statement
Ctrl + P	Parameter info (within method call arguments)
Ctrl + Q	Quick documentation lookup
Shift + F1	External Doc
Ctrl + mouse over code	Brief info
Ctrl + F1	Show descriptions of error or warning at caret
Alt + Insert	Generate code...
Ctrl + O	Override methods
Ctrl + Alt + T	Surround with...
Ctrl + /	Comment/uncomment with line comment
Ctrl + Shift + /	Comment/uncomment with block comment
Ctrl + W	Select successively increasing code blocks
Ctrl + Shift + W	Decrease current selection to previous state
Ctrl + Shift +]/[	Select till code block end/start
Alt + Enter	Show intention actions and quick-fixes
Ctrl + Alt + L	Reformat code
Ctrl + Alt + O	Optimize imports
Ctrl + Alt + I	Auto-indent line(s)
Tab / Shift + Tab	Indent/unindent selected lines
Ctrl + X or Shift + Delete	Cut current line or selected block to clipboard
Ctrl + C or Ctrl + Insert	Copy current line or selected block to clipboard
Ctrl + V or Shift + Insert	Paste from clipboard
Ctrl + Shift + V	Paste from recent buffers...
Ctrl + D	Duplicate current line or selected block
Ctrl + Y	Delete line at caret
Ctrl + Shift + J	Smart line join
Ctrl + Enter	Smart line split
Shift + Enter	Start new line
Ctrl + Shift + U	Toggle case for word at caret or selected block
Ctrl + Delete	Delete to word end
Ctrl + Backspace	Delete to word start

PyCharm Default Keymap

Running

Alt + Shift + F10	Select configuration and run
Alt + Shift + F9	Select configuration and debug
Shift + F10	Run
Shift + F9	Debug
Ctrl + Shift + F10	Run context configuration from editor
Ctrl+Alt+R	Run manage.py task

Debugging

F8	Step over
F7	Step into
Shift + F8	Step out
Alt + F9	Run to cursor
Alt + F8	Evaluate expression
Ctrl + Alt + F8	Quick evaluate expression
F9	Resume program
Ctrl + F8	Toggle breakpoint
Ctrl + Shift + F8	View breakpoints

Navigation

Ctrl + N	Go to class
Ctrl + Shift + N	Go to file
Ctrl + Alt + Shift + N	Go to symbol
Alt + Right/Left	Go to next/previous editor tab
F12	Go back to previous tool window
Esc	Go to editor (from tool window)
Shift + Esc	Hide active or last active window
Ctrl + Shift + F4	Close active run/messages/find/... tab
Ctrl + G	Go to line
Ctrl + E	Recent files popup
Ctrl + Alt + Left/Right	Navigate back/forward
Ctrl + Shift + Backspace	Navigate to last edit location
Alt + F1	Select current file or symbol in any view
Ctrl + B or Ctrl + Click	Go to declaration
Ctrl + Alt + B	Go to implementation(s)
Ctrl + Shift + I	Open quick definition lookup

(From jetbrains blog)

Jupiter/IPython notebook shortcuts

The Jupyter Notebook has two different keyboard input modes. **Edit mode** allows you to type code/text into a cell and is indicated by a green cell border. **Command mode** binds the keyboard to notebook level actions and is indicated by a grey cell border.

Command Mode (press `Esc` to enable)

- `Enter` : enter edit mode
- `Shift` - `Enter` : run cell, select below
- `Ctrl` - `Enter` : run cell
- `Alt` - `Enter` : run cell, insert below
- `Y` : to code
- `M` : to markdown
- `R` : to raw
- `1` : to heading 1
- `2` : to heading 2
- `3` : to heading 3
- `4` : to heading 4
- `5` : to heading 5
- `6` : to heading 6
- `K` : select cell above
- `Up` : select cell above
- `J` : select cell below
- `Down` : select cell below
- `Shift` - `K` : extend selection above
- `Shift` - `J` : extend selection below
- `A` : insert cell above
- `B` : insert cell below
- `X` : cut selected cell
- `C` : copy selected cell
- `Shift` - `V` : paste cell above
- `V` : paste cell below
- `Z` : undo last cell deletion
- `D,D` : delete selected cell
- `Shift` - `M` : merge selected cells
- `S` : Save and Checkpoint
- `Ctrl` - `S` : Save and Checkpoint
- `L` : toggle line numbers
- `O` : toggle output
- `Shift` - `O` : toggle output scrolling
- `Esc` : close pager
- `Q` : close pager
- `H` : show keyboard shortcut help dialog
- `I,I` : interrupt kernel
- `0,0` : restart kernel
- `Shift` - `Space` : scroll up

(From stackoverflow)

https://github.com/rhiever/Data-Analysis-and-Machine-Learning-Projects/blob/master/example-data-science-notebook/Example%20Machine%20Learning%20Notebook.ipynb

Numpy

- Numpy is possibly the most important library in Python for numerical computing
- Provides vector and matrix operations on top of *arrays*
- Almost every other library manipulates numpy arrays underneath

Scipy

- A scientific computing framework
- Linear Algebra
- Optimisation
- Statistics
- Clustering

Scikit-learn

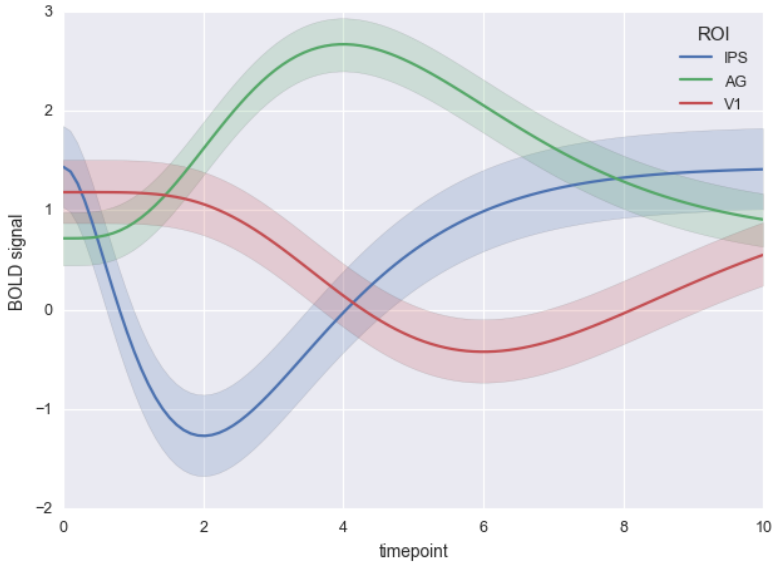
- A machine learning framework
- Includes almost everything, apart from neural networks
- We are going to use it extensively
- Super-fast trees
- Excellent documentation
- `http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html`
  - Cross validation for time series

Keras

- A neural networks framework
- Very popular
- Uses theano or tensorflow underneath
- We will use this as well
- Though notice this is not a module on neural networks
  - But you can delve into this if you want
  - Not trivial, but not super hard either
  - Again, a lot of examples and online tutorials

Matplotlib, seaborn

- Standard visualisation tools



Pandas

- *R* had dataframes
  - Essentially, a very SQL-like table-like data structure
- "DataFrame is a 2-dimensional labeled data structure with columns of potentially different types. You can think of it like a spreadsheet or SQL table, or a dict of Series objects. It is generally the most commonly used pandas object"
- You can manipulate these, and it helps a lot with cleaning up and re-shaping your data
- This is a big part of data science!
  - Data munging/data wrangling

Apache Spark

- The clustering framework
- You need it when you have tons of data to process
- Has its own machine learning library (mlib), which we are not going to use
  - But it makes sense to use it if your data doesn't fit in memory
  - Can be used with 3rd party modules in conjuction with sk-learn
- Sits on top of HDFS (which we are going to install and use later on)

Github

- All your code for your project will need to be publicly available
- Create a github account if you don't have one
- Two directories (`/src`, `/pdf`)
  - One for the pdf of the project
  - One for the code
  - If you have an ipython ipnb it should go here
- Add a README.md as well!