Introduction and overview of software

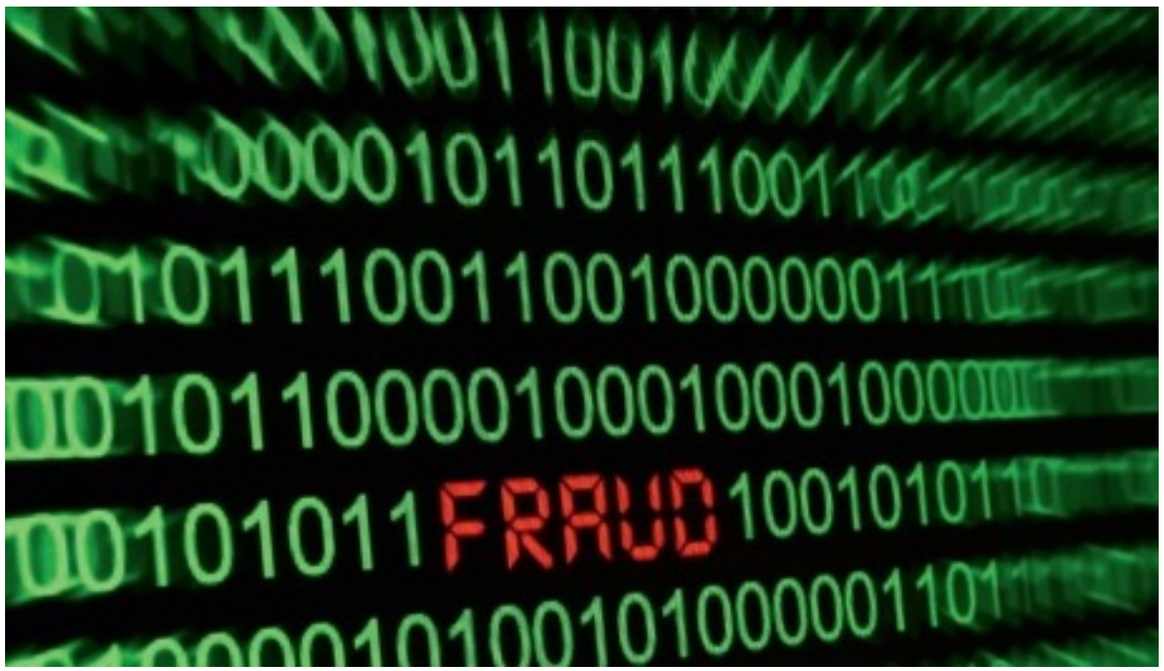
James Catmore

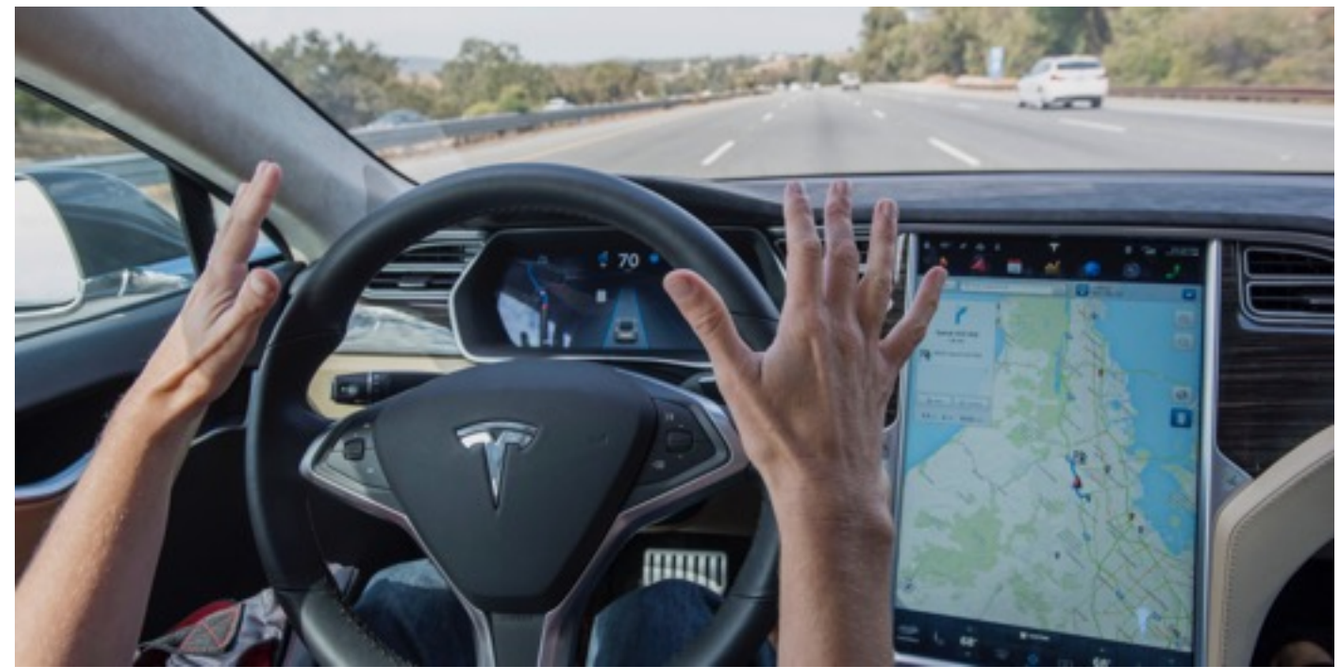
What is machine learning?

- Behaviour of a ML algorithm is determined by input data rather than explicit conditional steps mandated by the programmer, implying the building of some internal representation of the data

Type	Outcome
Supervised: algorithm is trained using labelled data, such that it is taught how to map input to output	***Classification***
	Regression
Unsupervised: algorithm discovers structure in the data without guidance	***Dimensionality reduction***
	Density estimation
	Clustering
Reinforcement: algorithm attempts to maximise a score based on a range of actions available to it	Autonomous vehicles Game-learning algorithms Animal brains?

Machine learning in the outside world





Norwegian | Russian | English | Detect language

Tenkte jeg skulle minne om denne, som kanskje har gått i glemmeboka i løpet av påsken. Foreløpig har jeg notert 0- null - innspill, noe som antakelig er positivt fordi det indikerer at folk ikke opplever tidstyver/tidspress som veldig problematisk. Men jeg ønsker allikevel ethvert innspill velkomment, innen fristen 1.4.

English | Russian | Norwegian | Translate

Just thought I'd remind you this, that might have been forgotten during Easter. Currently I have listed 0- zero - input, which is probably positive because it indicates that people do not experience time thieves / time constraints as very problematic. But I want anyway any suggestions welcome, deadline 1.4.

Suggest an edit

Points for discussion

- Where are we currently using these techniques in our respective areas?
  - In some cases we have been using them for years but by different names
  - Mainly classification/regression? Or other techniques as well?
- What software are we using?
- What challenges/limitations have we experienced?
- What do we plan to do in the future in this area?

Today’s meeting

- Short overview of software
- Round-table discussion about what we’re all doing
- Presentation from theory (Are)
- Presentation from EPF (James and Eirik)
- Opportunities for future work/collaboration

Software

- It is best not to try to code up ML algorithms if possible
  - especially not the cost function minimisation or implementation of linear-algebra
- Instead it is better to use off-the-shelf software products and modify if necessary
  - Many of the tools are open-source
  - Some are even well documented…
  - Usually written in C++, Python, or in the R or Matlab environments
  - Typically contain a wide range of different algorithms and APIs to enable easy extensions
- Two obvious examples widely used in fundamental physics: TMVA and SciKit-Learn

Software

	TMVA	SciKit Learn
Language	C++ in ROOT environment	Python
Data format	ROOT TTrees	NumPy arrays
Graphics	ROOT	Matplotlib
Documentation	http://tmva.sourceforge.net	http://scikit-learn.org/
Interoperability	Via Pythonic ROOT (pyROOT or RootPy)	

- Due to the boom in “deep learning” a number of packages have been developed to make this easier
- They can be interfaced with SKL
- TMVA also has a built-in NN implementation

High-level examples (building, configuration)	Lasagne Keras Caffe
Low-level examples (minimization, linear algebra)	Theano TensorFlow

Software challenges

- Platforms: most of the software works on Linux and Mac (with some patience); no idea about Windows
    - Software updates often break the installations - especially with Macs
    - For Python, virtualenv is the way to go - allows a “private” Python installation separate from the system installation
- Resources
    - Can only do small/medium scale tasks on a laptop/desktop; larger computing facilities may be needed for some jobs
    - Making use of multiple CPUs or GPUs can increase performance significantly but often not easy to set up
    - Sensible memory management needed for larger scale tasks
- Interoperability
    - Thanks to Pythonic ROOT it is trivial to combine ROOT/TMVA and the various Python products
    - A lot of work going into making R compatible with ROOT (will it be called TR?)
    - Not necessarily the case with other tools/environments