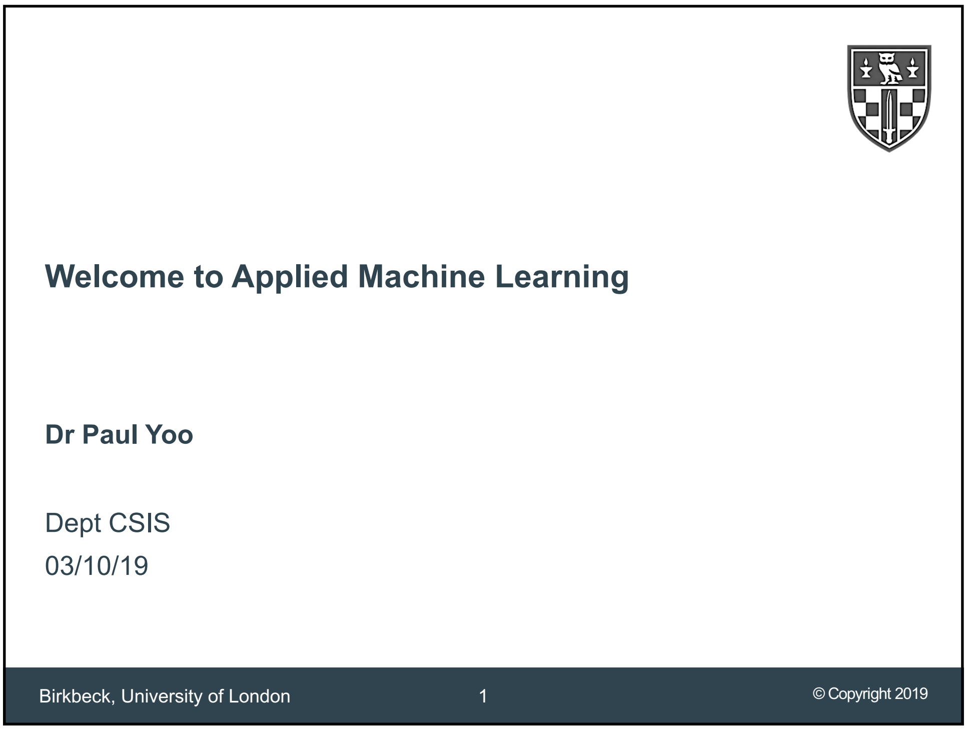
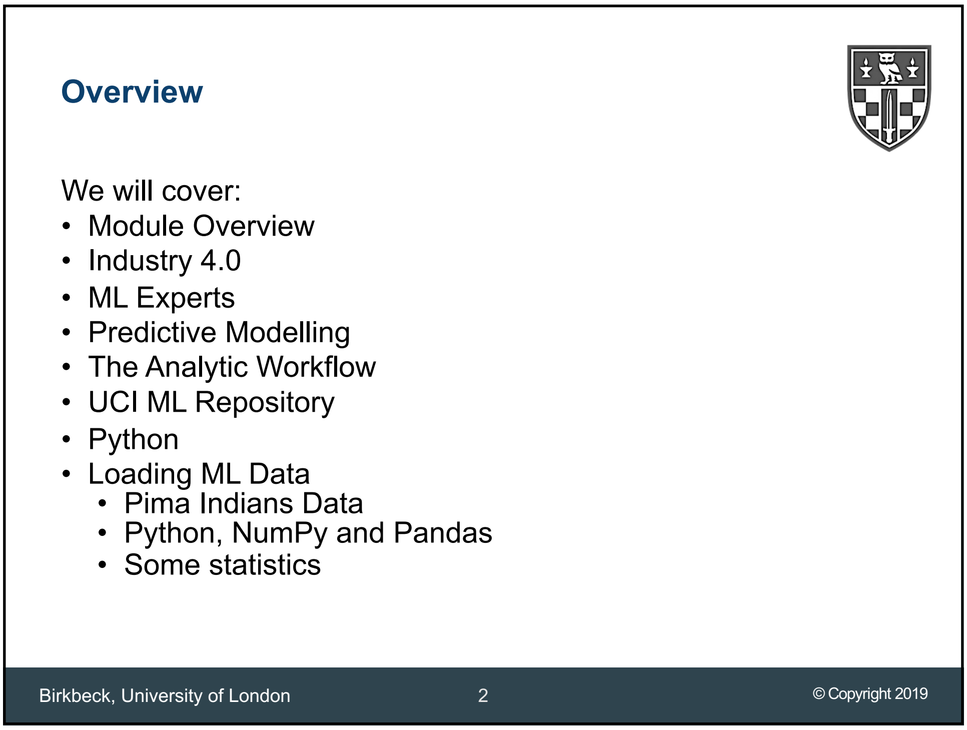


Welcome to Applied Machine Learning

Dr Paul Yoo

Dept CSIS

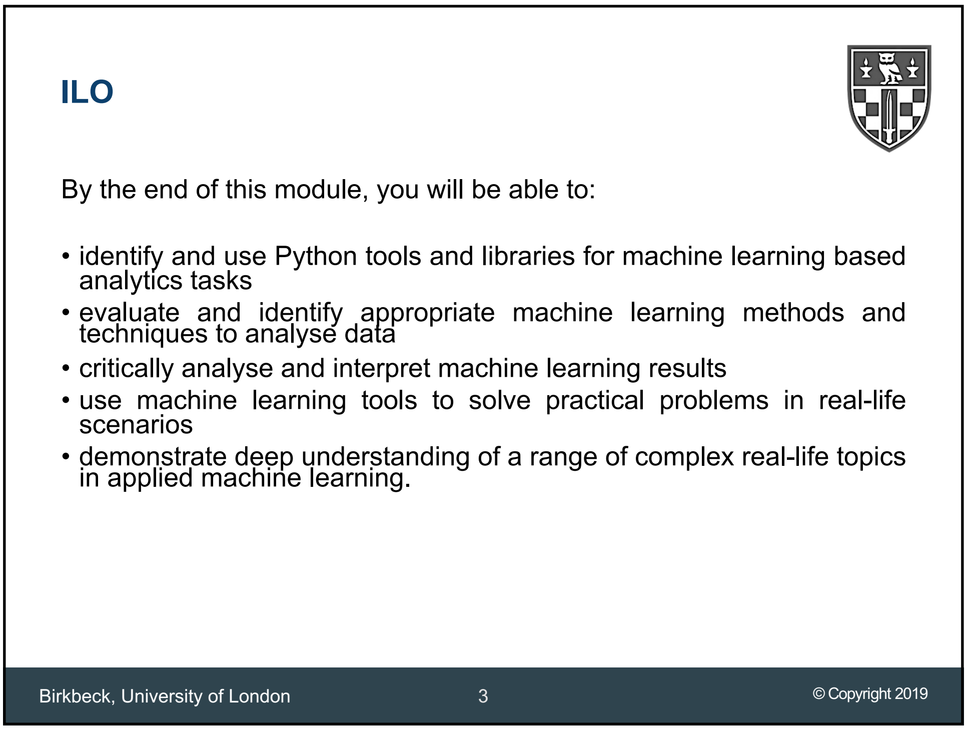
03/10/19



Overview

We will cover:

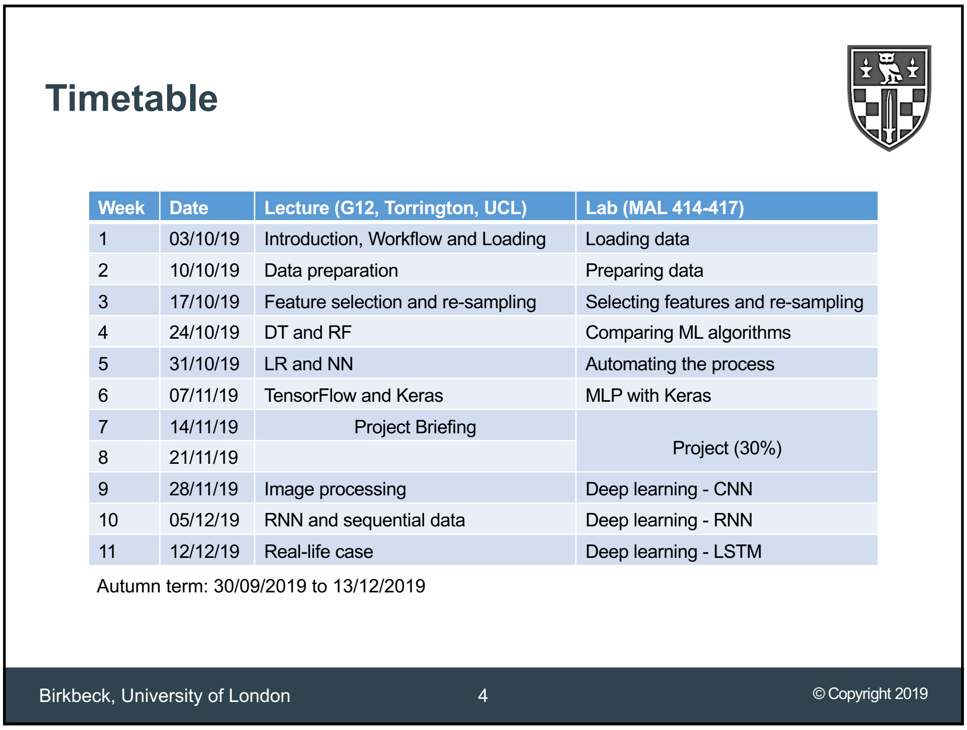
- Module Overview
- Industry 4.0
- ML Experts
- Predictive Modelling
- The Analytic Workflow
- UCI ML Repository
- Python
- Loading ML Data
  - Pima Indians Data
  - Python, NumPy and Pandas
  - Some statistics



ILO

By the end of this module, you will be able to:

- identify and use Python tools and libraries for machine learning based analytics tasks
- evaluate and identify appropriate machine learning methods and techniques to analyse data
- critically analyse and interpret machine learning results
- use machine learning tools to solve practical problems in real-life scenarios
- demonstrate deep understanding of a range of complex real-life topics in applied machine learning.



Timetable

Week	Date	Lecture (G12, Torrington, UCL)	Lab (MAL 414-417)
1	03/10/19	Introduction, Workflow and Loading	Loading data
2	10/10/19	Data preparation	Preparing data
3	17/10/19	Feature selection and re-sampling	Selecting features and re-sampling
4	24/10/19	DT and RF	Comparing ML algorithms
5	31/10/19	LR and NN	Automating the process
6	07/11/19	TensorFlow and Keras	MLP with Keras
7	14/11/19	Project Briefing	Project (30%)
8	21/11/19		
9	28/11/19	Image processing	Deep learning - CNN
10	05/12/19	RNN and sequential data	Deep learning - RNN
11	12/12/19	Real-life case	Deep learning - LSTM

Autumn term: 30/09/2019 to 13/12/2019

Assessment

- Final exam worth 70% of your total mark
- A report (inc. individual section) of a group project worth 30% of your total mark
  - Publication Date: 11/11/19
  - Deadline: 15/12/19
  - Late cut-off deadline: 29/12/19
  - Mark return: 05/01/20
- More details will be provided at the project briefing (W7)

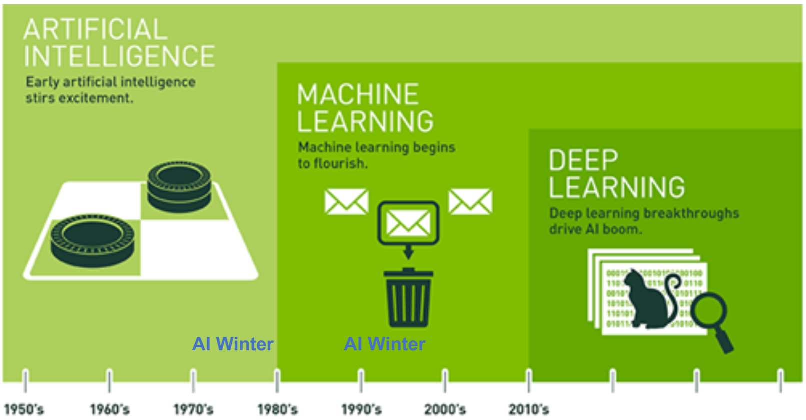
How Computers are Learning to be Creative by Blaise Agüera y Arcas

URL: https://youtu.be/uSUOdu_5MPc

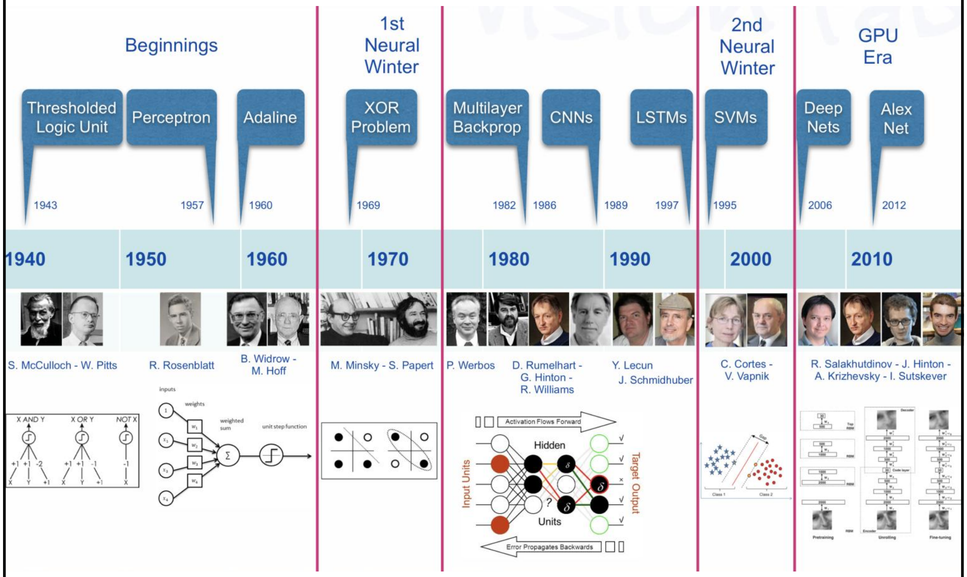
5:45 – 17:34

Machine Learning Experts You Need to Know

- Geoffrey Hinton – backpropagation (1980s), Boltzmann machines and CapsNet (URL: https://youtu.be/uAu3jQWaN6E)
- Michael I Jordan – RNN (1980s)
- Yann LeCun – CNN with backprogagation
- Yoshua Bengio – RNN
- Jürgen Schmidhuber - LSTM
- Andrew Ng – Coursera, deeplearning.ai, Google Brain project, Landing AI (SaaS)
- Vladimir Vapnik – SVM (1963)
- Ian Goodfellow – GANs (2014)
- Blaise Agüera y Arcas – Google TPU 3 teraops (10^{12} per sec) with 1 watt


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
AI Winter
AI Winter
1950's
1960's
1970's
1980's
1990's
2000's
2010's

Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then



Beginnings

1st Neural Winter

2nd Neural Winter

GPU Era

Thresholded Logic Unit 1943

Perceptron 1957

Adaline 1960

XOR Problem 1969

Multilayer Backprop 1982 1986

CNNs 1989

LSTMs 1997

SVMs 1995

Deep Nets 2006

Alex Net 2012

1940 1950 1960 1970 1980 1990 2000 2010

S. McCulloch - W. Pitts

R. Rosenblatt

B. Widrow - M. Hoff

M. Minsky - S. Papert

P. Werbos

D. Rumelhart - G. Hinton - R. Williams

Y. Lecun

J. Schmidhuber

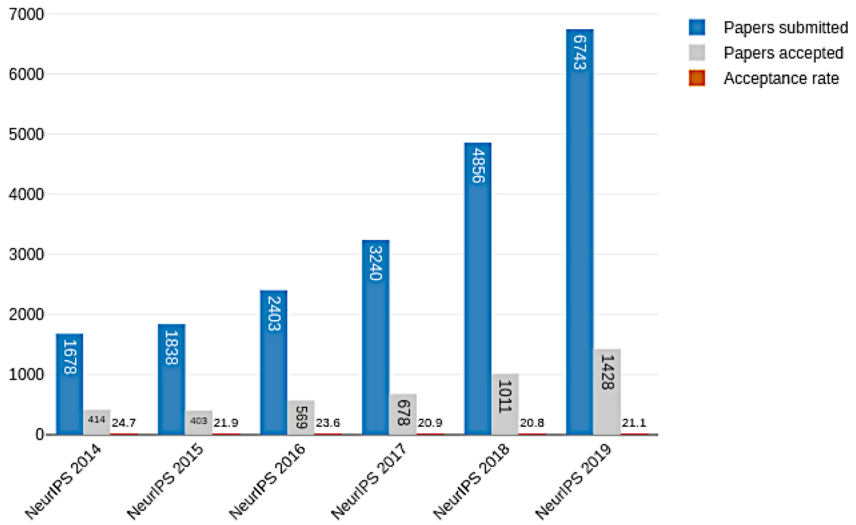
C. Cortes - V. Vapnik

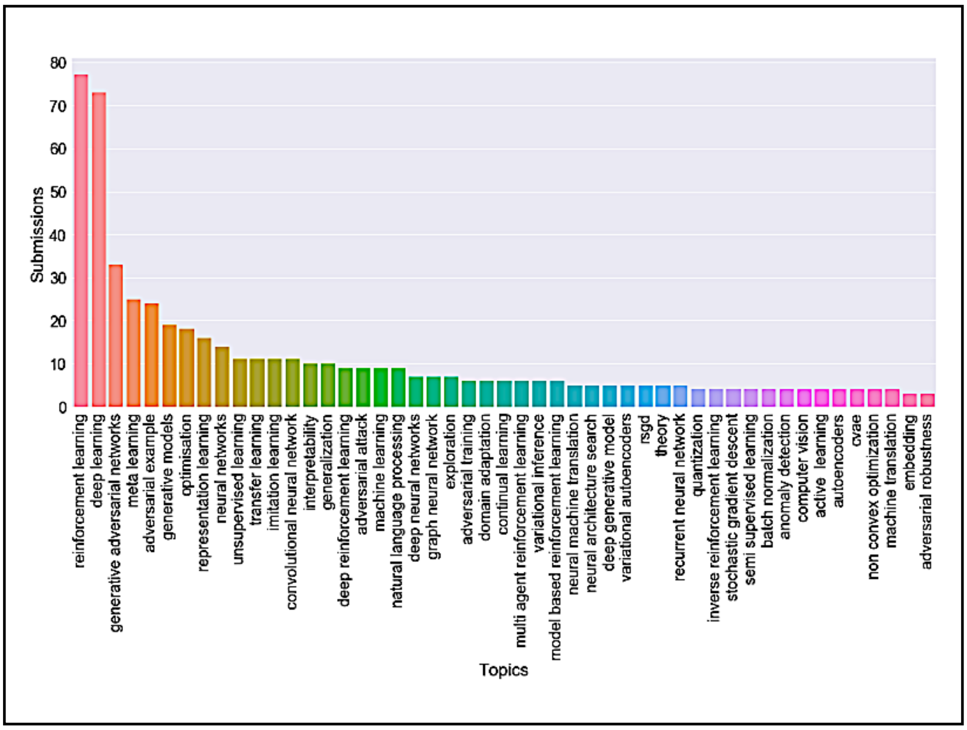
R. Salakhutdinov - J. Hinton - A. Krizhevsky - I. Sutskever

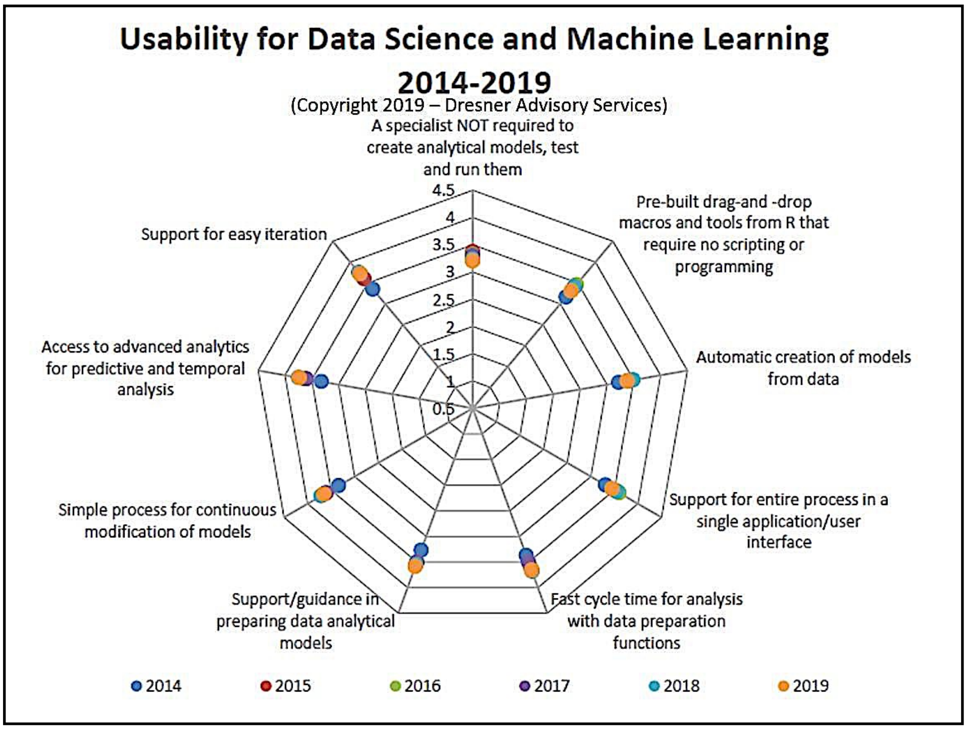
The 33rd Neural Information Processing Systems Conference, Vancouver, Canada (2019)

URL: https://neurips.cc

Statistics of acceptance rate NeurIPS



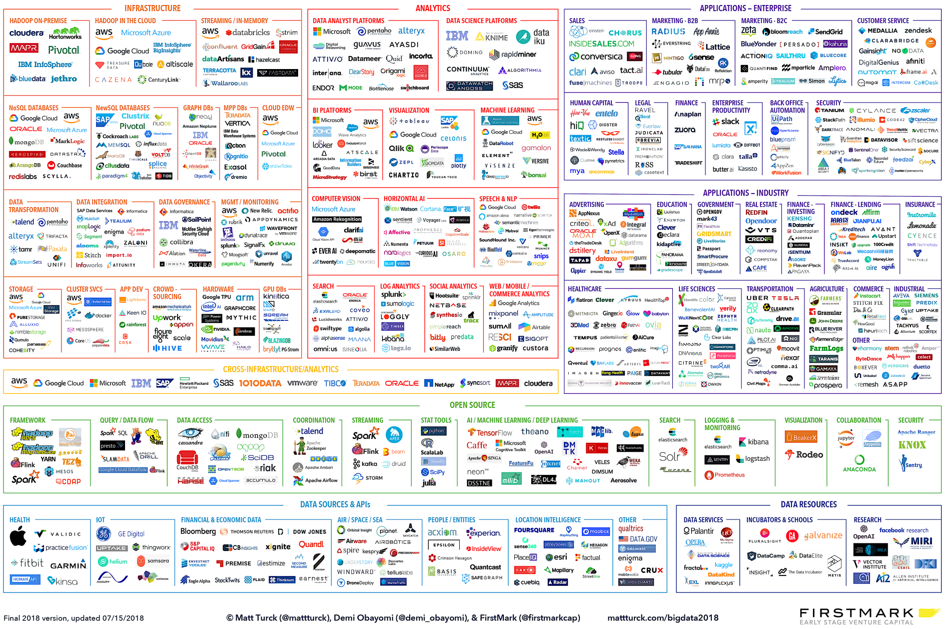




Usability for Data Science and Machine Learning 2014-2019

(Copyright 2019 – Dresner Advisory Services)

BIG DATA & AI LANDSCAPE 2018



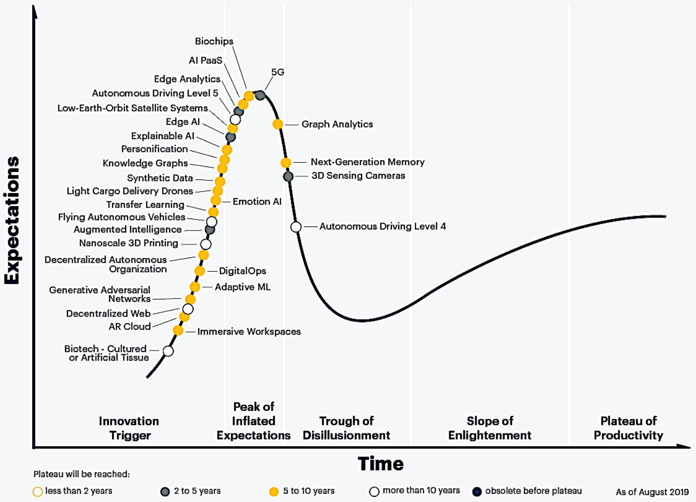
Final 2018 version, updated 07/15/2018

© Matt Turck (@mattturck), Demi Obayomi (@demi_obayomi), & FirstMark (@firstmarkcap)

mattturck.com/bigdata2018

FIRSTMARK
EARLY STAGE VENTURE CAPITAL

Gartner Hype Cycle for Emerging Technologies, 2019

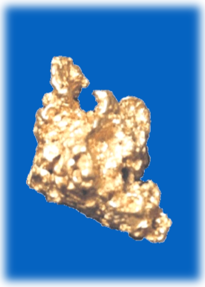


gartner.com/SmarterWithGartner

Source: Gartner
© 2019 Gartner, Inc. and/or its affiliates. All rights reserved.

Gartner.

Predictive Modelling



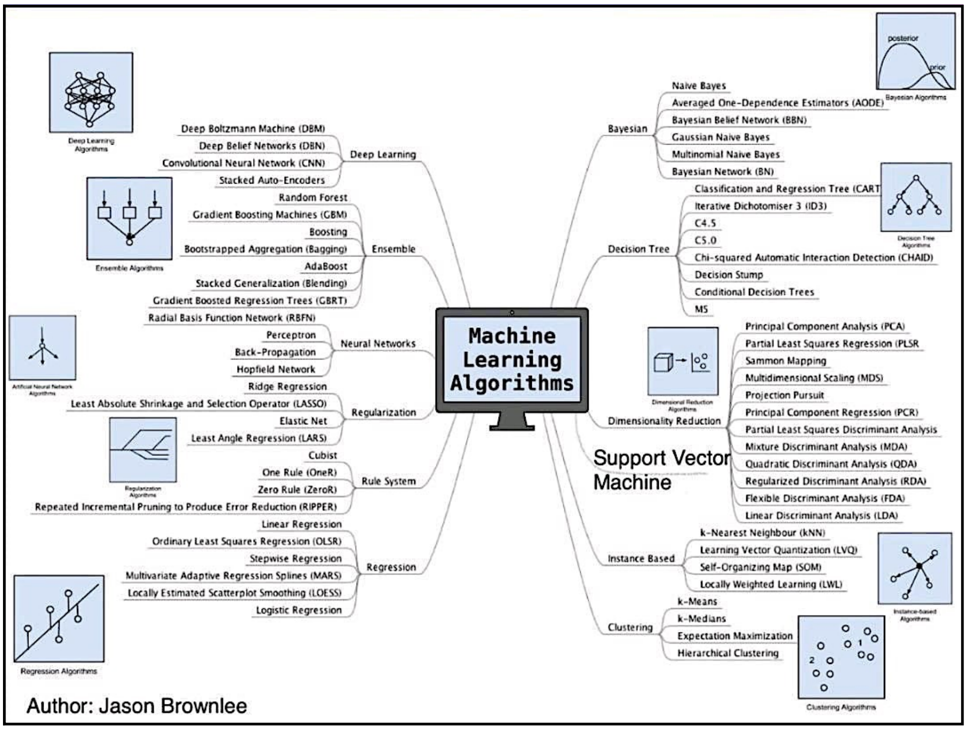
The Essence of Data Mining

> ***"Most of the big payoff [in data mining] has been in predictive modeling."***
>
> – Herb Edelstein

This module focuses on a specific sub-field of machine learning called predictive modeling.

Predictive Modelling ML Steps

1. **Define Problem:** Investigate and characterise the problem in order to better understand the goals of the project.
2. **Analyse Data:** Use descriptive statistics and visualisation to better understand the data you have available.
3. **Prepare Data:** Use data transforms in order to better expose the structure of the prediction problem to modeling algorithms.
4. **Evaluate Algorithms:** Design a test harness to evaluate a number of standard algorithms on the data and select the top few to investigate further.
5. **Improve Results:** Use algorithm tuning and ensemble methods to get the most out of well-performing algorithms on your data.
6. **Present Results:** Finalise the model, make predictions and present results.

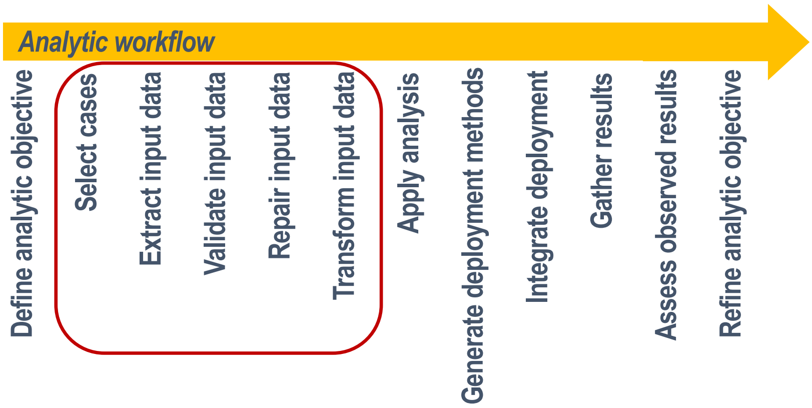


Machine Learning Algorithms

- Deep Learning
  - Deep Boltzmann Machine (DBM)
  - Deep Belief Networks (DBN)
  - Convolutional Neural Network (CNN)
  - Stacked Auto-Encoders
- Ensemble
  - Random Forest
  - Gradient Boosting Machines (GBM)
  - Boosting
  - Bootstrapped Aggregation (Bagging)
  - AdaBoost
  - Stacked Generalization (Blending)
  - Gradient Boosted Regression Trees (GBRT)
- Neural Networks
  - Radial Basis Function Network (RBFN)
  - Perceptron
  - Back-Propagation
  - Hopfield Network
- Regularization
  - Ridge Regression
  - Least Absolute Shrinkage and Selection Operator (LASSO)
  - Elastic Net
  - Least Angle Regression (LARS)
- Rule System
  - Cubist
  - One Rule (OneR)
  - Zero Rule (ZeroR)
  - Repeated Incremental Pruning to Produce Error Reduction (RIPPER)
- Regression
  - Linear Regression
  - Ordinary Least Squares Regression (OLSR)
  - Stepwise Regression
  - Multivariate Adaptive Regression Splines (MARS)
  - Locally Estimated Scatterplot Smoothing (LOESS)
  - Logistic Regression
- Bayesian
  - Naive Bayes
  - Averaged One-Dependence Estimators (AODE)
  - Bayesian Belief Network (BBN)
  - Gaussian Naive Bayes
  - Multinomial Naive Bayes
  - Bayesian Network (BN)
- Decision Tree
  - Classification and Regression Tree (CART
  - Iterative Dichotomiser 3 (ID3)
  - C4.5
  - C5.0
  - Chi-squared Automatic Interaction Detection (CHAID)
  - Decision Stump
  - Conditional Decision Trees
  - M5
- Dimensionality Reduction
  - Principal Component Analysis (PCA)
  - Partial Least Squares Regression (PLSR
  - Sammon Mapping
  - Multidimensional Scaling (MDS)
  - Projection Pursuit
  - Principal Component Regression (PCR)
  - Partial Least Squares Discriminant Analysis
  - Mixture Discriminant Analysis (MDA)
  - Quadratic Discriminant Analysis (QDA)
  - Regularized Discriminant Analysis (RDA)
  - Flexible Discriminant Analysis (FDA)
  - Linear Discriminant Analysis (LDA)
- Support Vector Machine
- Instance Based
  - k-Nearest Neighbour (kNN)
  - Learning Vector Quantization (LVQ)
  - Self-Organizing Map (SOM)
  - Locally Weighted Learning (LWL)
- Clustering
  - k-Means
  - k-Medians
  - Expectation Maximization
  - Hierarchical Clustering

Author: Jason Brownlee

The Analytic Workflow



Analytic workflow

Define analytic objective

Select cases

Extract input data

Validate input data

Repair input data

Transform input data

Apply analysis

Generate deployment methods

Integrate deployment

Gather results

Assess observed results

Refine analytic objective

Birkbeck, University of London 18 © Copyright 2019

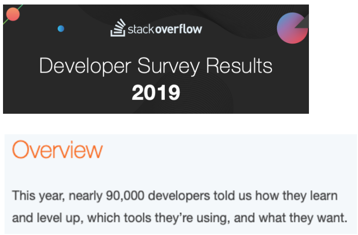
UCI Machine Learning repository

http://archive.ics.uci.edu/ml/index.php

- Small – fit into memory and model them in reasonable time
- Well behaved – don’t need to do a lot of feature engineering
- Benchmarks – many people have used them

Python

It is consistently appearing in the top 10 programming languages in surveys on StackOverflow.



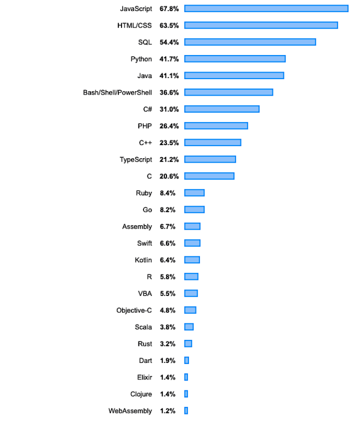
stackoverflow

Developer Survey Results
2019

Overview

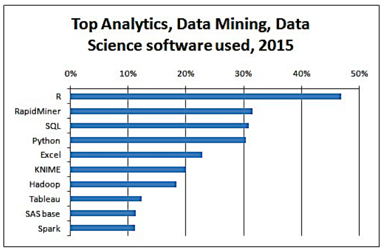
This year, nearly 90,000 developers told us how they learn and level up, which tools they're using, and what they want.

URL: https://insights.stackoverflow.com/survey/2019



87,354 responses; select all that apply

KDD Nuggets tool survey in 2015

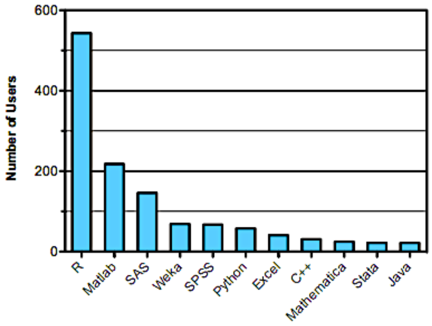


The top 10 tools by share of users were

1. **R**, 46.9% share (38.5% in 2014)
2. **RapidMiner**, 31.5% (44.2% in 2014)
3. **SQL**, 30.9% (25.3% in 2014)
4. **Python**, 30.3% (19.5% in 2014)
5. **Excel**, 22.9% (25.8% in 2014)
6. **KNIME**, 20.0% (15.0% in 2014)
7. **Hadoop**, 18.4% (12.7% in 2014)
8. **Tableau**, 12.4% (9.1% in 2014)
9. **SAS**, 11.3 (10.9% in 2014)

0. **Spark**, 11.3% (2.6% in 2014)

Kaggle platform survey in 2011

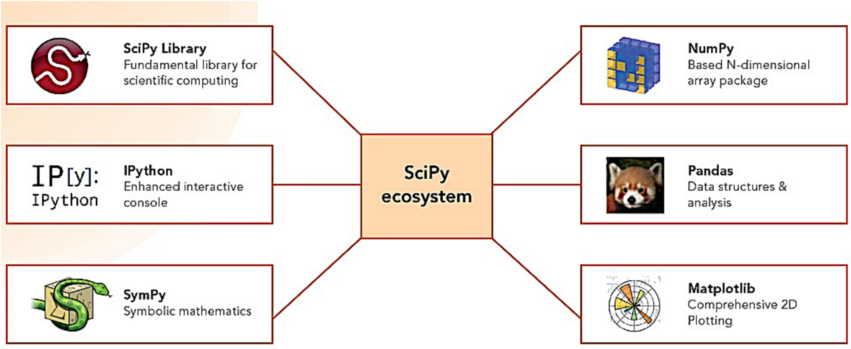


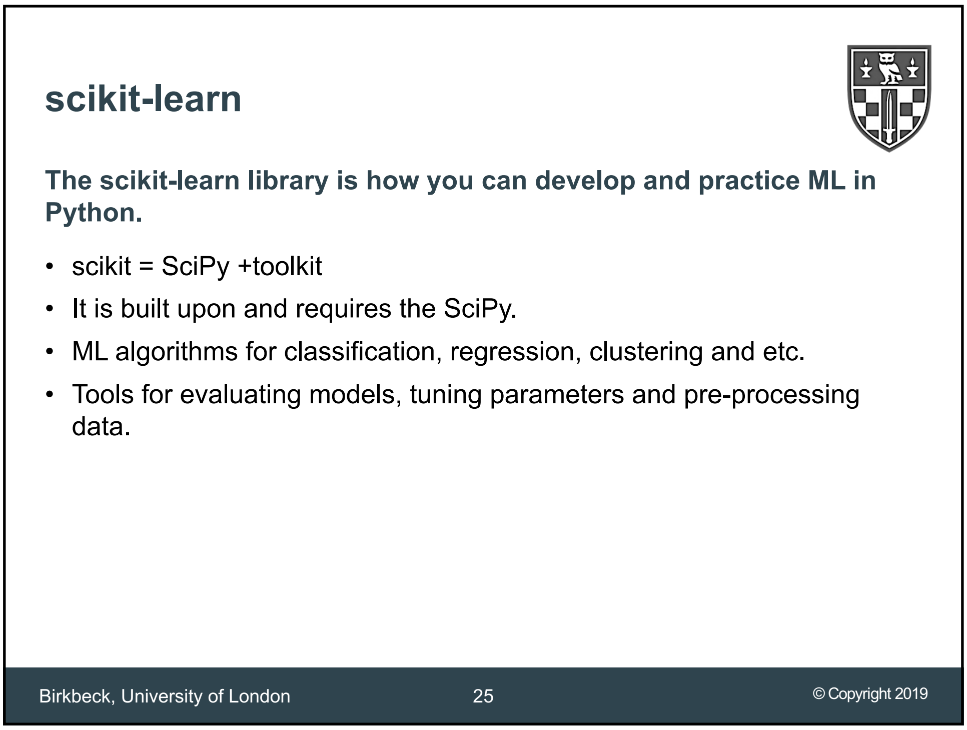
SciPy

SciPy is a free and open-source Python library used for scientific computing and technical computing.

- It is an add-on to Python that you will need for machine learning.
- It contains modules for optimisation, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering.
- It is comprised of the following core modules relevant to machine learning:
  - NumPy: A foundation for SciPy that allows you to efficiently work with data in arrays.
  - Matplotlib: Allows you to create 2D charts and plots from data.
  - Pandas: Tools and data structures to organise and analyse your data. (to load explore and better understand your data)

SciPy ecosystem

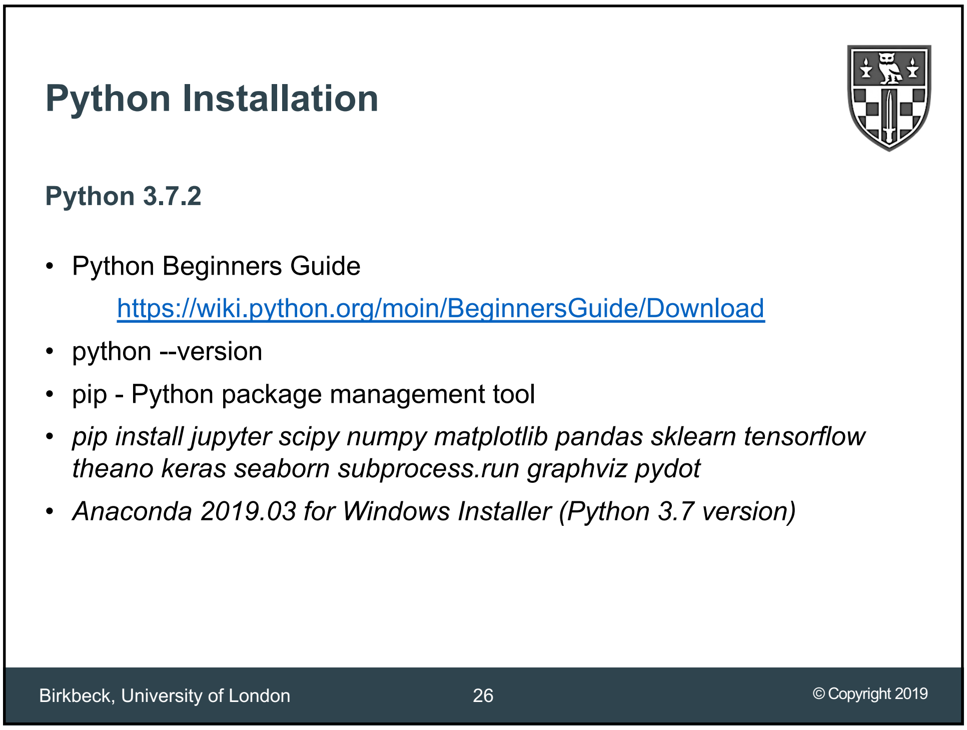

SciPy Library
Fundamental library for scientific computing
IP[y]: IPython
IPython
Enhanced interactive console
SymPy
Symbolic mathematics
SciPy ecosystem
NumPy
Based N-dimensional array package
Pandas
Data structures & analysis
Matplotlib
Comprehensive 2D Plotting



scikit-learn

The scikit-learn library is how you can develop and practice ML in Python.

- scikit = SciPy +toolkit
- It is built upon and requires the SciPy.
- ML algorithms for classification, regression, clustering and etc.
- Tools for evaluating models, tuning parameters and pre-processing data.

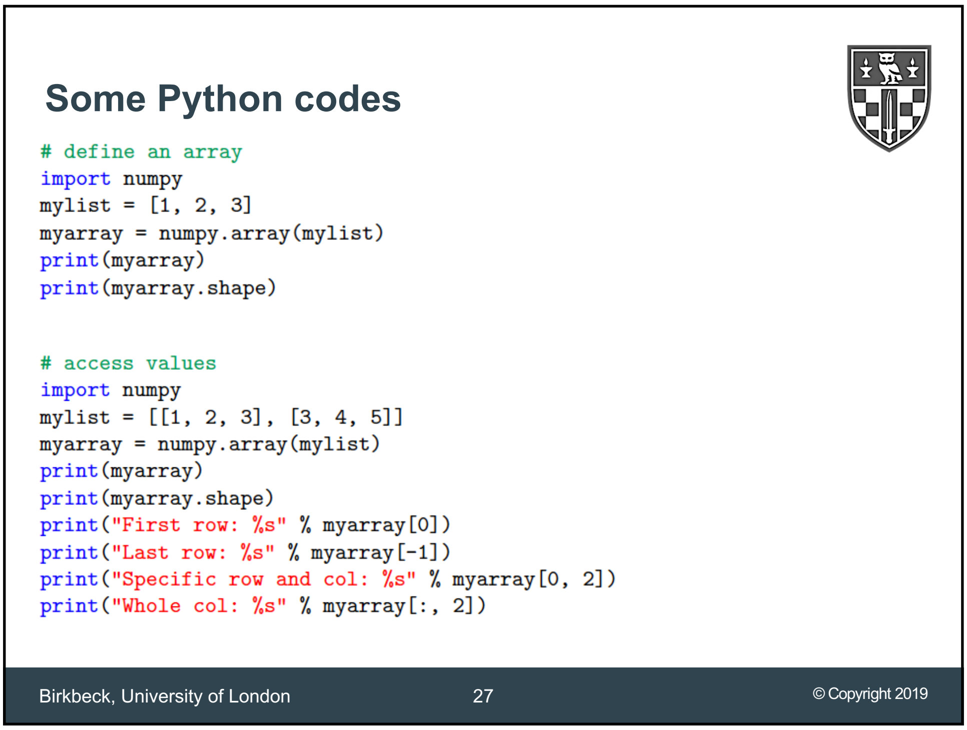


Python Installation

Python 3.7.2

- Python Beginners Guide

  https://wiki.python.org/moin/BeginnersGuide/Download
- python --version
- pip - Python package management tool
- *pip install jupyter scipy numpy matplotlib pandas sklearn tensorflow theano keras seaborn subprocess.run graphviz pydot*
- *Anaconda 2019.03 for Windows Installer (Python 3.7 version)*

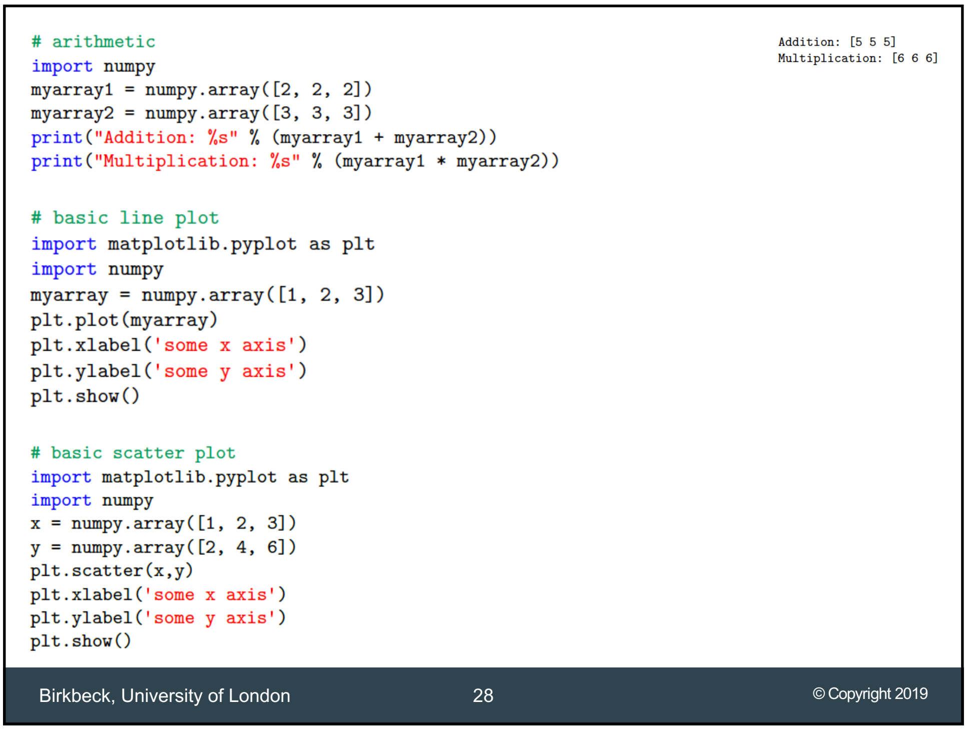


Some Python codes

```
# define an array
import numpy
mylist = [1, 2, 3]
myarray = numpy.array(mylist)
print(myarray)
print(myarray.shape)


# access values
import numpy
mylist = [[1, 2, 3], [3, 4, 5]]
myarray = numpy.array(mylist)
print(myarray)
print(myarray.shape)
print("First row: %s" % myarray[0])
print("Last row: %s" % myarray[-1])
print("Specific row and col: %s" % myarray[0, 2])
print("Whole col: %s" % myarray[:, 2])
```

Birkbeck, University of London 27 © Copyright 2019



```
# arithmetic
import numpy
myarray1 = numpy.array([2, 2, 2])
myarray2 = numpy.array([3, 3, 3])
print("Addition: %s" % (myarray1 + myarray2))
print("Multiplication: %s" % (myarray1 * myarray2))

# basic line plot
import matplotlib.pyplot as plt
import numpy
myarray = numpy.array([1, 2, 3])
plt.plot(myarray)
plt.xlabel('some x axis')
plt.ylabel('some y axis')
plt.show()

# basic scatter plot
import matplotlib.pyplot as plt
import numpy
x = numpy.array([1, 2, 3])
y = numpy.array([2, 4, 6])
plt.scatter(x,y)
plt.xlabel('some x axis')
plt.ylabel('some y axis')
plt.show()
```

```
Addition: [5 5 5]
Multiplication: [6 6 6]
```

Birkbeck, University of London 28 © Copyright 2019

```
# series
import numpy
import pandas
myarray = numpy.array([1, 2, 3])
rownames = ['a', 'b', 'c']
myseries = pandas.Series(myarray, index=rownames)
print(myseries)


print(myseries[0])
print(myseries['a'])


# dataframe
import numpy
import pandas
myarray = numpy.array([[1, 2, 3], [4, 5, 6]])
rownames = ['a', 'b']
colnames = ['one', 'two', 'three']
mydataframe = pandas.DataFrame(myarray, index=rownames, columns=colnames)
print(mydataframe)



print("method 1:")
print("one column:\n%s" % mydataframe['one'])
print("method 2:")
print("one column:\n%s" % mydataframe.one)
```

Summary

This lecture we’ve covered the basics of AML including:

- Module Overview
- Industry 4.0
- ML Experts
- Predictive Modelling
- The Analytic Workflow
- UCI ML Repository
- Python, NumPy and Pandas

Next week

- Data Preparation

Labs

- MAL 414—417

Questions?

paul@dcs.bbk.ac.uk