

# ODSC Boston 2016

<https://www.odsc.com/boston>

**20-22 May 2016**

[adam.karwan@gmail.com](mailto:adam.karwan@gmail.com)

# JULIA programming language



- core implementation MIT license
- packages licenses GPL, LGPL, BSD

- BASH
- MATLAB
- PYTHON
- C

[<http://julialang.org>]

	Fortran	Julia	Python	R	Matlab	Octave	Mathe- matica	JavaScript	Go	LuaJIT	Java
	gcc 5.1.1	0.4.0	3.4.3	3.2.2	R2015b	4.0.0	10.2.0	V8 3.28.71.19	go1.5	gsl-shell 2.3.1	1.8.0_45
fib	0.70	2.11	77.76	533.52	26.89	9324.35	118.53	3.36	1.86	1.71	1.21
parse_int	5.05	1.45	17.02	45.73	802.52	9581.44	15.02	6.06	1.20	5.77	3.35
quicksort	1.31	1.15	32.89	264.54	4.92	1866.01	43.23	2.70	1.29	2.03	2.60
mandel	0.81	0.79	15.32	53.16	7.58	451.81	5.13	0.66	1.11	0.67	1.35
pi_sum	1.00	1.00	21.99	9.56	1.00	299.31	1.69	1.01	1.00	1.00	1.00
rand_mat_stat	1.45	1.66	17.93	14.56	14.52	30.93	5.95	2.30	2.96	3.27	3.92
rand_mat_mul	3.48	1.02	1.14	1.57	1.12	1.12	1.30	15.07	1.42	1.16	2.36

**Figure:** benchmark times relative to C (smaller is better, C performance = 1.0).

# datarobot.com

## Overview

Total Equity Funding

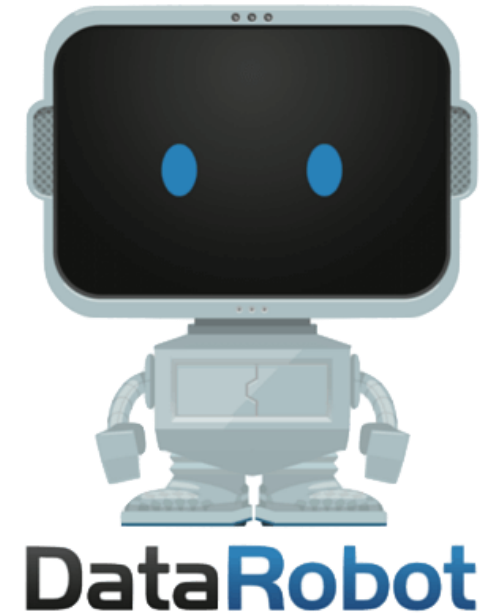
**\$57.42M** in **4 Rounds** from **10 Investors**

Most Recent Funding

**\$33M Series B** on February 11, 2016

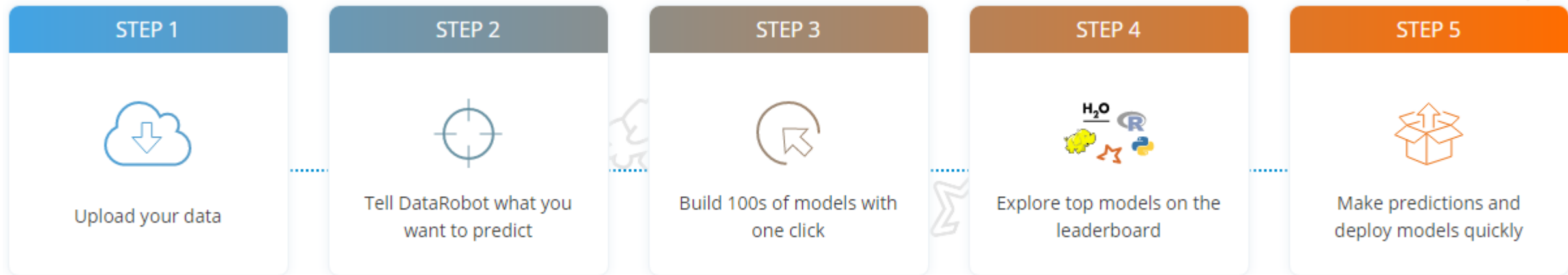
Headquarters:

**Boston, Massachusetts**



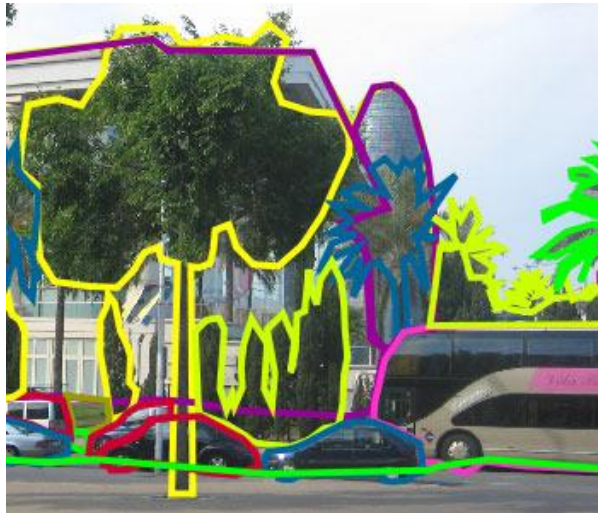
[<https://angel.co/datarobot>]

[<https://www.crunchbase.com/organization/datarobot#/entity>]



# CrowdFlower

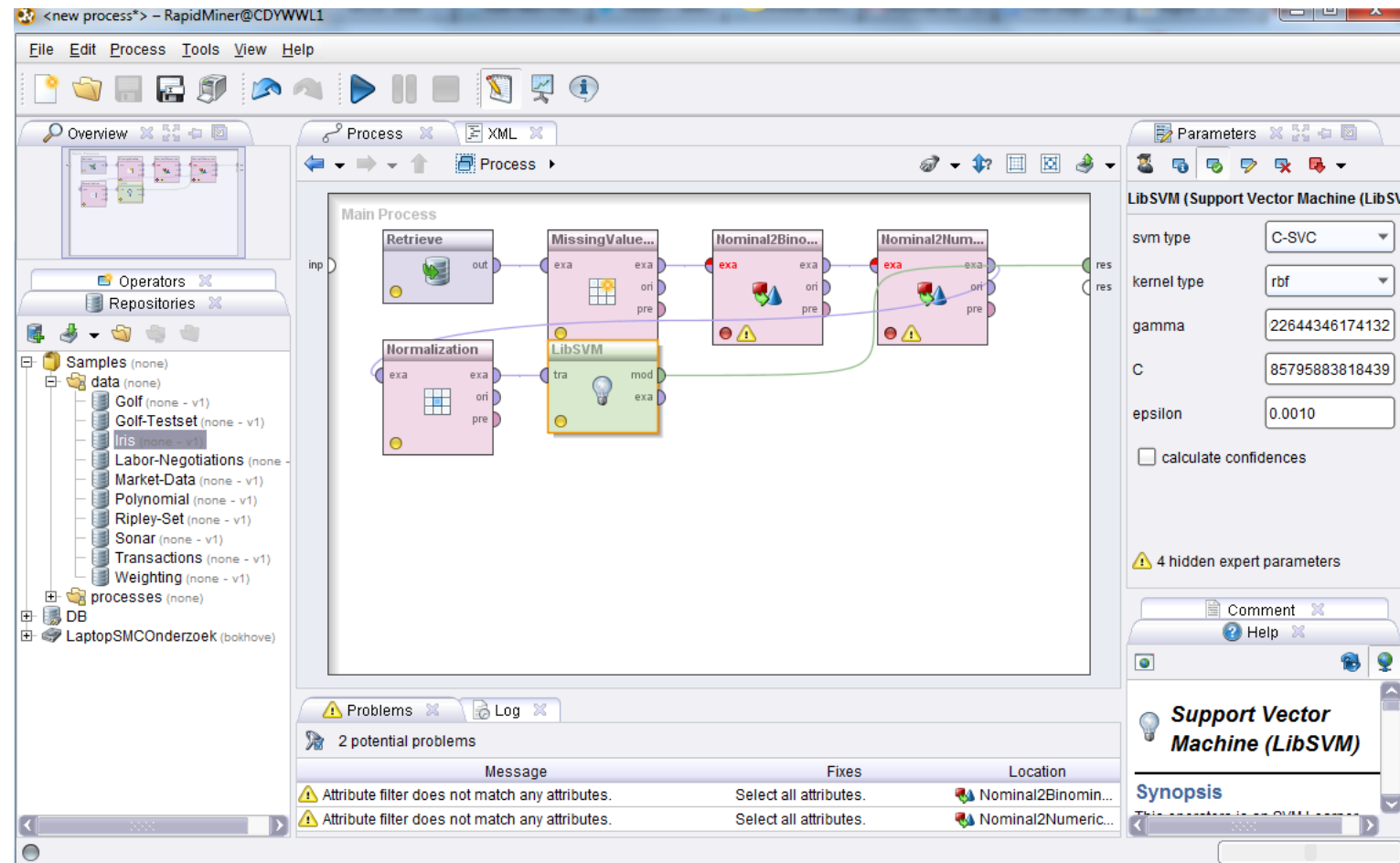
- SaaS (Software as a Service)
- Access to online workforce to of milions of people to clean, label, enrich data
- Similiar to Amazon Mechanical Turku



dataset LabelMe from MIT  
[<http://labelme.csail.mit.edu/Release3.0>]

# RapidMiner

- Cross – Platform
  - machine-learning
  - data mining
  - text mining
  - predictive analytics
  - business analytics
- Project at Dortmund Uni
- Ingo Miersva



The screenshot displays the RapidMiner interface with a workflow titled "Main Process". The workflow consists of the following steps:

- Retrieve**: Connects to a data source.
- Normalization**: Normalizes the data.
- MissingValue...**: Handles missing values.
- LibSVM**: The core machine learning process.
- Nominal2Bino...**: Converts nominal attributes to binary.
- Nominal2Num...**: Converts nominal attributes to numeric.

The right-hand pane shows the **Parameters** for the **LibSVM (Support Vector Machine (LibSVM))** process:

- svm type: C-SVC
- kernel type: rbf
- gamma: 22644346174132
- C: 85795883818439
- epsilon: 0.0010
- ☐ calculate confidences

Below the parameters, there is a section for **Problems** and **Log**. The **Problems** section indicates "2 potential problems":

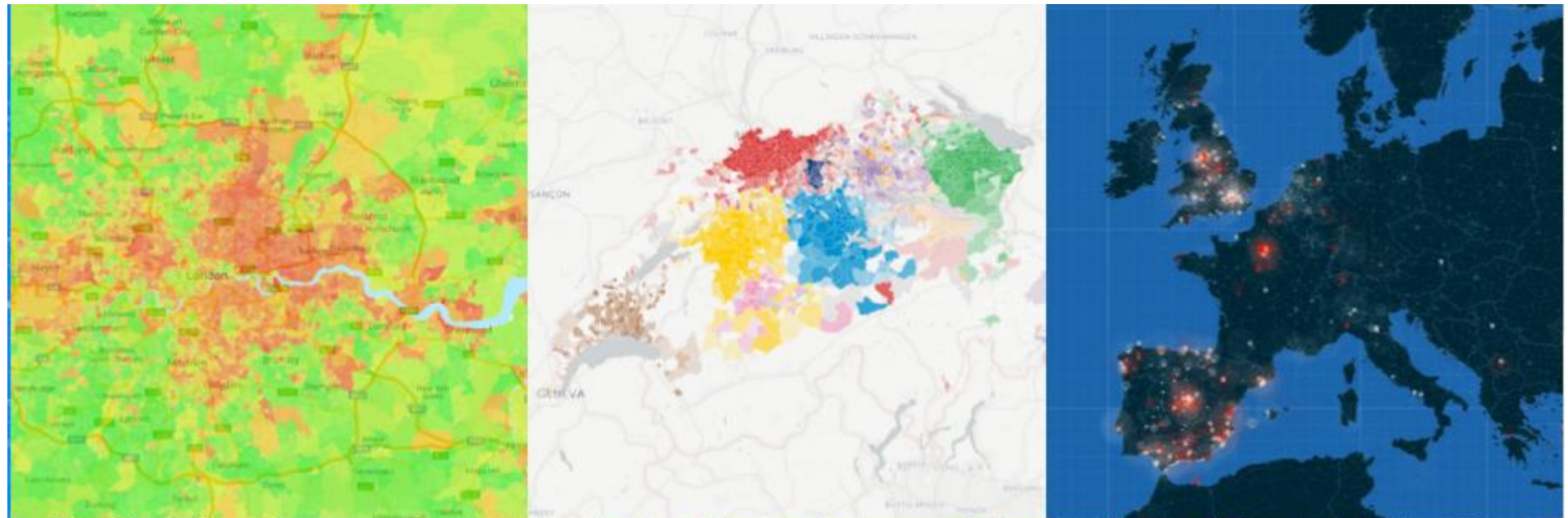
Message	Fixes	Location
Attribute filter does not match any attributes.	Select all attributes.	Nominal2Binomin...
Attribute filter does not match any attributes.	Select all attributes.	Nominal2Numeric...

The bottom right corner features a **Support Vector Machine (LibSVM)** synopsis.

# CartoDB.com



- SaaS cloud tool that provides GIS and web mapping
- Very effective visualizations with data filtering API



# H<sub>2</sub>O.ai

- OpenSource on R, Python and SPARK
- cool interface, speed & scalability

The logo for H2O.ai, featuring the text "H<sub>2</sub>O.ai" in a bold, black, sans-serif font. The "2" is a subscript. The background is a solid yellow square.

## Scientific Advisory Council

Stephen Boyd

Professor of EE Engineering, Stanford

Rob Tibshirani

Professor of Health Research and Policy, and  
Statistics, Stanford

Trevor Hastie

Professor of Statistics, Stanford

Springer Texts in Statistics

Gareth James  
Daniela Witten  
Trevor Hastie  
Robert Tibshirani

## An Introduction to Statistical Learning

with Applications in R

 Springer

# Others

- Pfizer
- Bokeh – NY taxi visualization
- Opendata (Chile)
- DTW (dynamic time warping) algorithm
- Medical data (open) <https://health.data.ny.gov/>
- Kdnuggets
- Kaggle
- Twitter data access – many tools

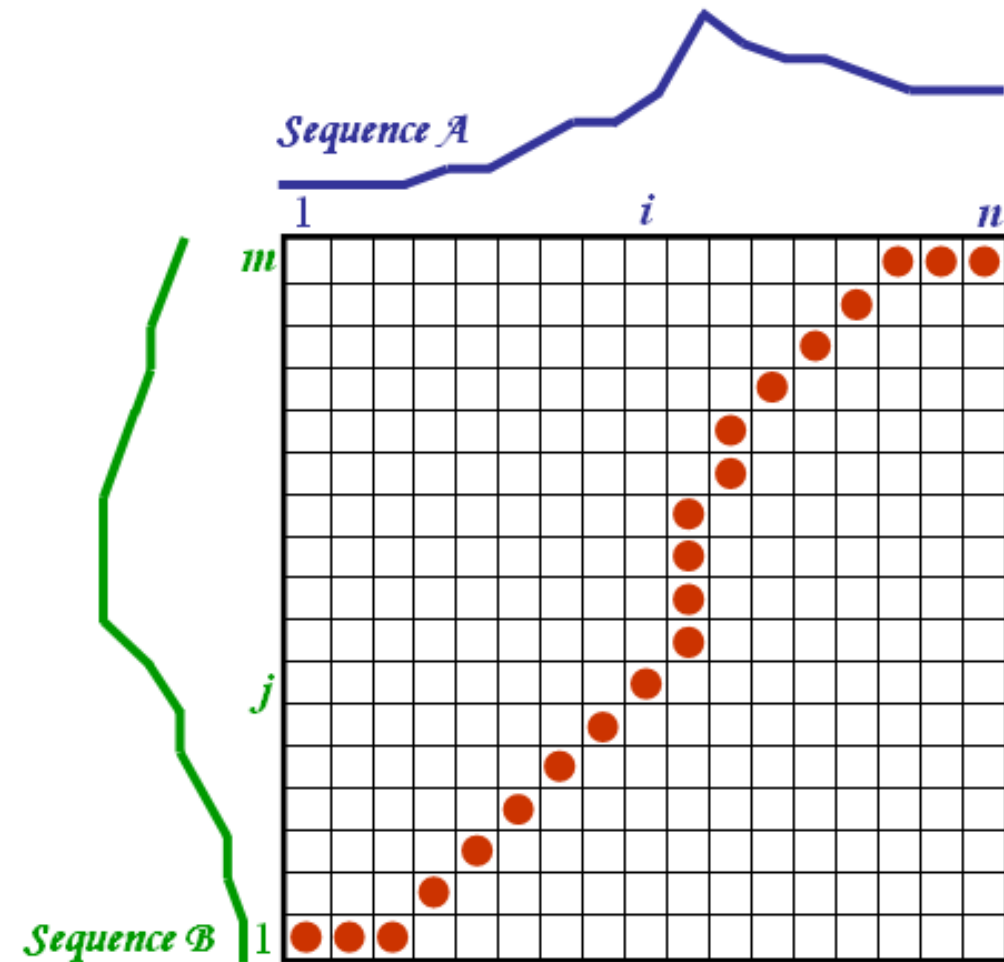
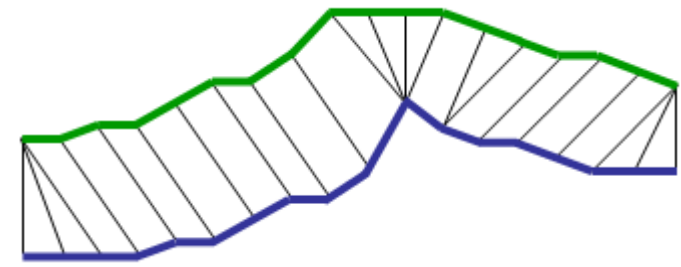


# DTW (dynamic time warping)

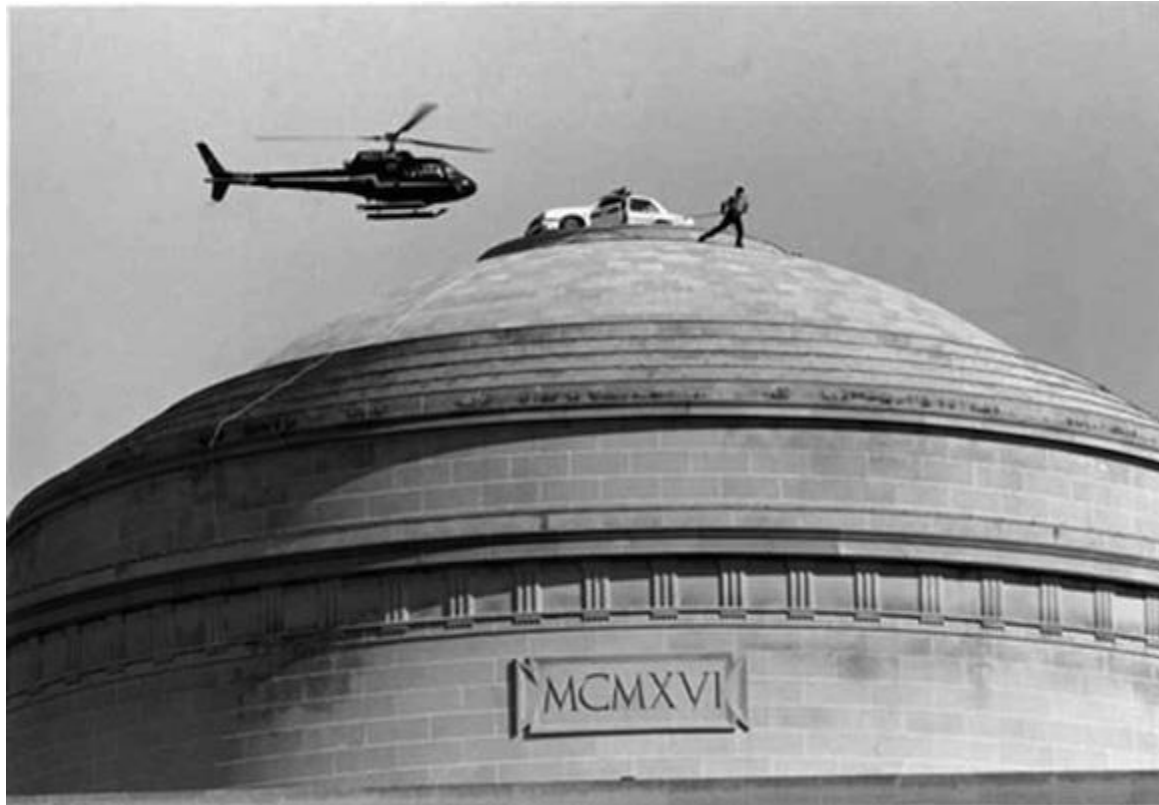
- For time series analysis problems

```
int DTWDistance(s: array [1..n], t: array [1..m]) {  
  DTW := array [0..n, 0..m]  
  
  for i := 1 to n  
    DTW[i, 0] := infinity  
  for i := 1 to m  
    DTW[0, i] := infinity  
  DTW[0, 0] := 0  
  
  for i := 1 to n  
    for j := 1 to m  
      cost := d(s[i], t[j])  
      DTW[i, j] := cost + minimum(DTW[i-1, j ],    // insertion  
                                  DTW[i , j-1],    // deletion  
                                  DTW[i-1, j-1])    // match  
  
  return DTW[n, m]  
}
```

[[http://wearables.cc.gatech.edu/paper\\_of\\_week/DTW\\_myths.pdf](http://wearables.cc.gatech.edu/paper_of_week/DTW_myths.pdf)]



# MIT hacks



# Links at the end

<http://erum.ue.poznan.pl> 12-14 October 2016, Poznań (Poland)

<http://bigdatatech.pl> 25 February 2016, Warsaw (Poland)