The Alan Turing Institute

MLJ Workshop

Resources: **github.com/ablaom/MLJTutorial**

Anthony Blaom with Quinn Asena


The
Alan Turing
Institute


THE UNIVERSITY OF
AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND


IQVIA


NeSI
New Zealand eScience
Infrastructure


WARWICK
THE UNIVERSITY OF WARWICK


Julia
computing

The Team

Core design: A. B., Franz Kiraly, Sebastian Vollmer

Lead development: A. B., Thibaut Lienart

Other contributors: Diego Arenas, Samuel Okon, Yiannis Simillides, Julian Samaroo, Ayush Shridar, Geoffroy Dolphin, Mosè Giordano...

Julia language consultants: Avik Sengupta

Some machine learning libraries in Julia

```
22-element Vector{String}:
 "OutlierDetectionNeighbors"
 "OutlierDetectionPython"
 "OutlierDetectionNetworks"
 "ScikitLearn"
 "DecisionTree"
 "MultivariateStats"
 "MLJModels"
 "BetaML"
 "MLJLinearModels"
 "LIBSVM"
 "EvoTrees"
 "NaiveBayes"
 "MLJFlux"
 "Clustering"
 "ParallelKMeans"
 "NearestNeighborModels"
 "PartialLeastSquaresRegressor"
 "LightGBM"
 "GLM"
 "TSVD"
 "MLJText"
 "XGBoost"
```

Some multi-paradigm machine learning toolboxes in Julia

- **ScikitLearn.jl** - thin wrapper for Python's scikit-learn
- **AutoMLPipeline.jl** - IBM
- **MLJ.jl**
- FastAI.jl - specific to neural networks (interface to **Flux.jl**)

What?

What's a machine learning toolbox?

What?

What's a machine learning toolbox?

- A toolbox provides a **uniform interface** for **fitting**, **evaluating** and **tuning** machine learning models.
- Provides common **preprocessing tasks** (such as data cleaning and type coercion)
- Allows for model **composition** (e.g., pipelining)

Why?

Why learn the MLJ toolbox?

– Written in Julia (but does wrap non-native models):

Why?

Why learn the MLJ toolbox?

- Written in Julia (but does wrap non-native models):
  - Easy access to core algorithms
  - Easier to customize
  - Easier to add new performant models
  - Greater transparency and reproducibility
  - Better composability with other libraries

Why?

Why learn the MLJ toolbox?

- Written in Julia (but does wrap non-native models):
  - Easy access to core algorithms
  - Easier to customize
  - Easier to add new performant models
  - Greater transparency and reproducibility
  - Better composability with other libraries
- Start-of-the-art **model composition** that plays well with everything else.
- Meta-algorithms (e.g., tuning) are model wrappers

How to entertain your kids at home - the case for MLJ

Playmobil ©

Lego ©

3 models dinosaurs + 15 user contributed models

We support you in building your snake powered train!

Workshop Overview

- Recap of supervised and unsupervised learning
- Preview of model composition (stacking)
- Part 1 - **Data Representation** + exercises
- Part 2 - **Selecting, Training and Evaluating Models** + exercises
- Break
- Part 3 - **Transformers and Pipelines** + exercises
- Part 4 - **Tuning hyper-parameters** + exercises
- Part 5 - **Advanced model composition** (time permitting)

Supervised Learning

Learning to **predict** some target variable **y** from a knowledge of some other variables **X** (the *input features*).

Supervised Learning

Learning to **predict** some target variable **y** from a knowledge of some other variables **X** (the *input features*).

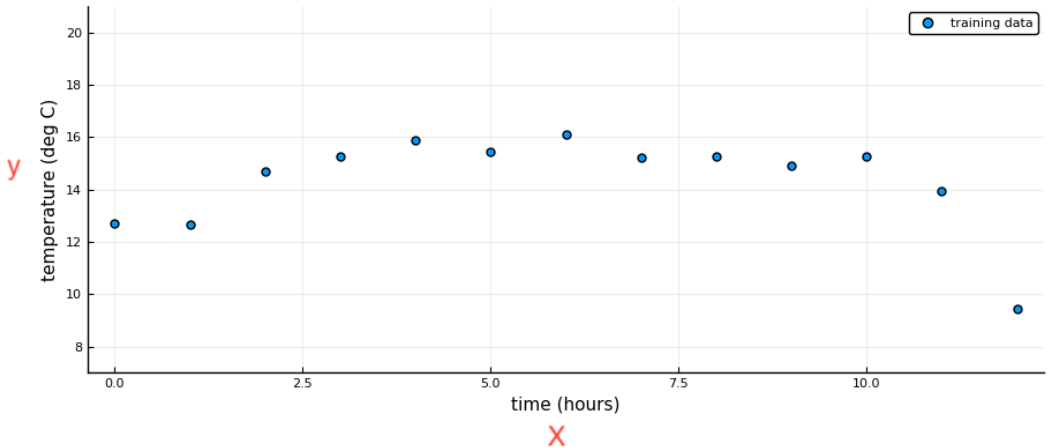
X

time	room
Int64	CategoricalValue{String,UInt32}
5	kitchen
5	bathroom
5	bedroom_1
5	living_room
6	kitchen
6	bathroom
6	bedroom_1
6	living_room
7	kitchen
7	bathroom
7	bedroom_1
7	living_room

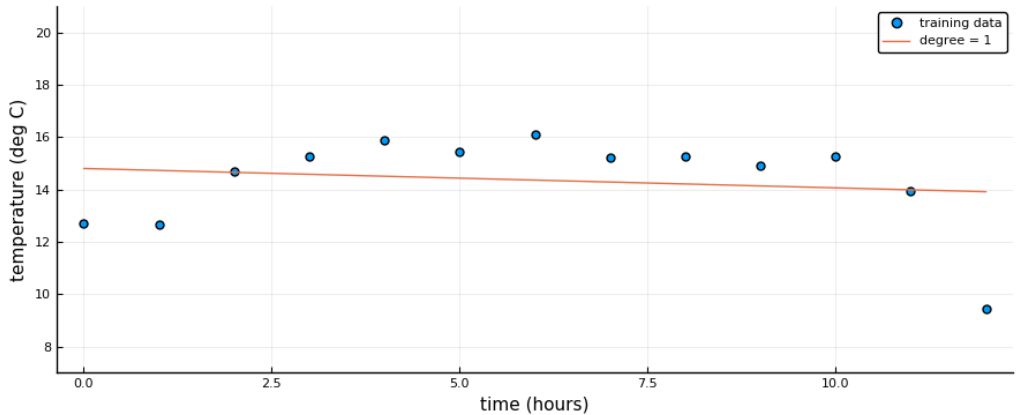
y

temperature
Float64
18.5
18.3
18.3
17.4
16.6
20.7
18.9
20.2
20.4
19.9
16.2
17.3

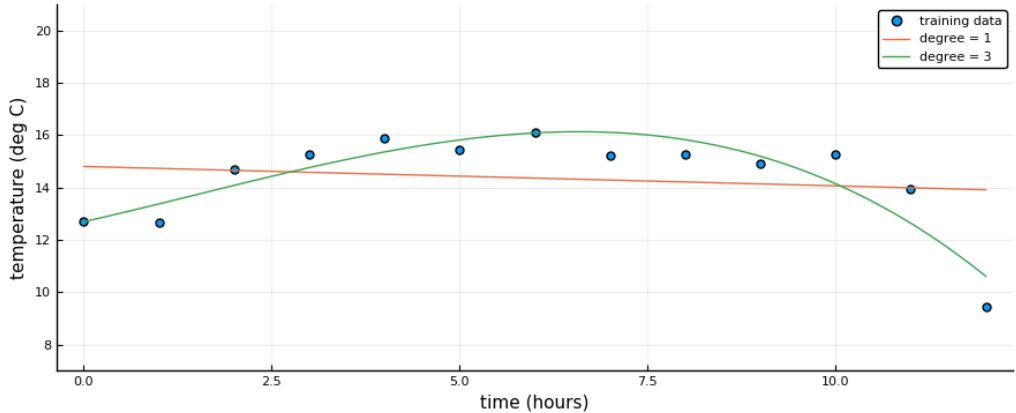
Supervised Learning



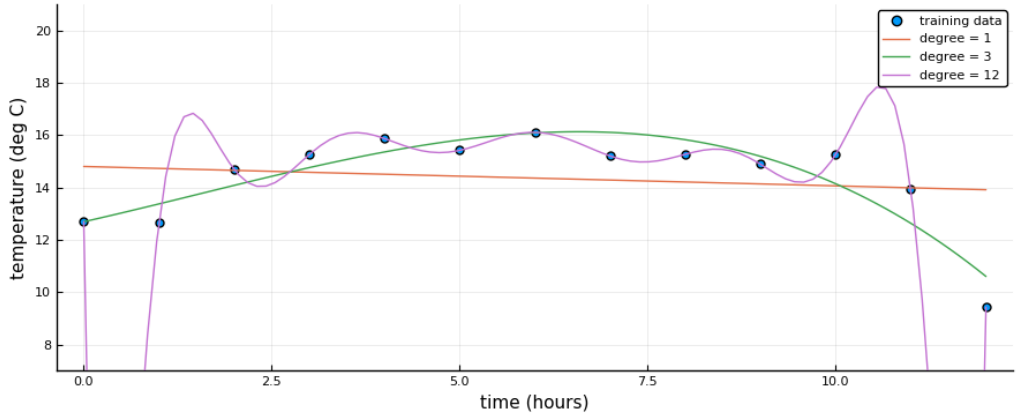
Supervised Learning



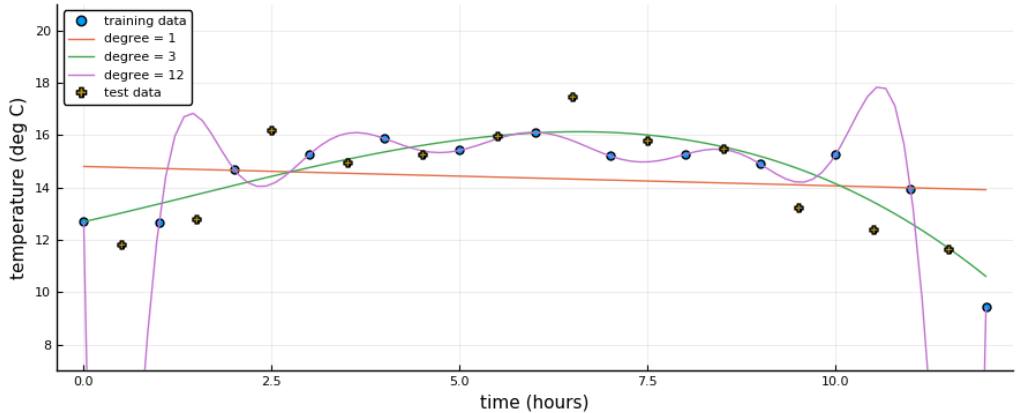
Supervised Learning



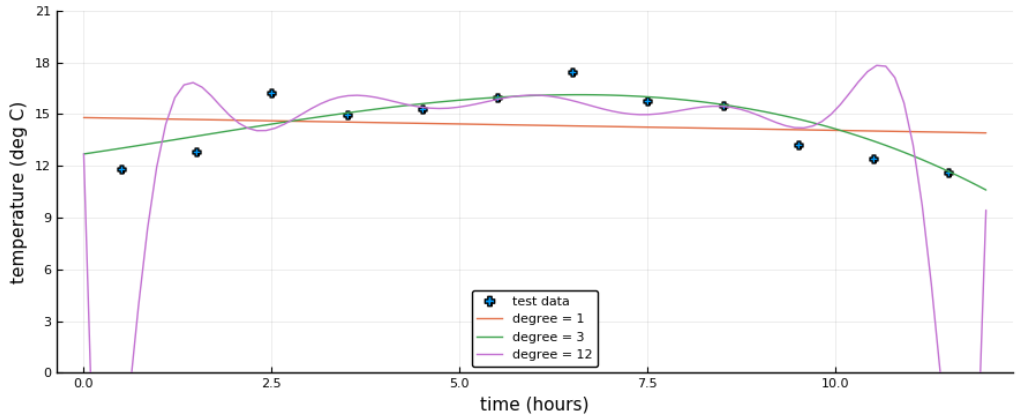
Supervised Learning



Supervised Learning

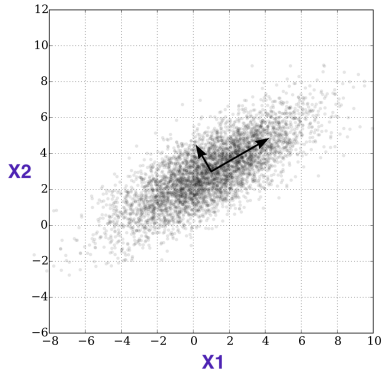


Supervised Learning



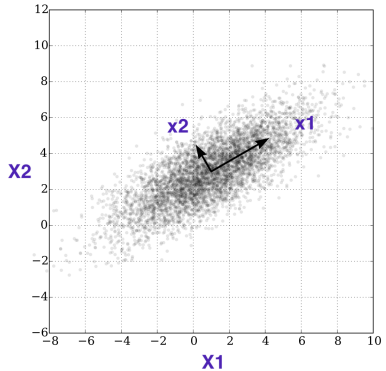
Unsupervised Learning

Learning data **transformations**, e.g., dimension reduction

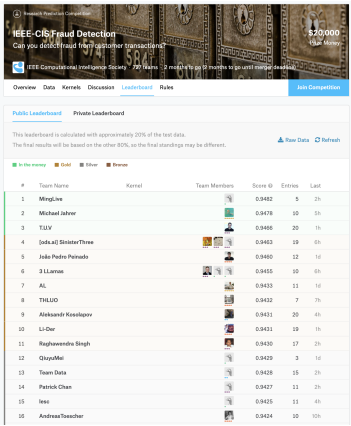


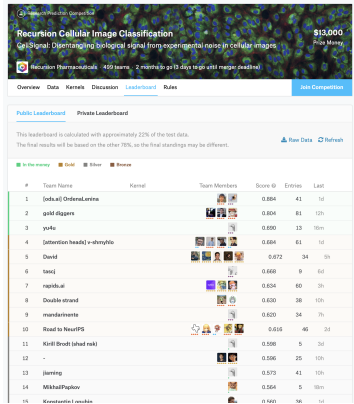
Unsupervised Learning

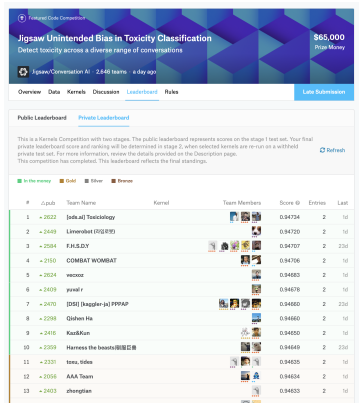
Learning data **transformations**, e.g., dimension reduction



Data science competitions (kaggle)







Installing the tutorials

github.com/ablaom/MLJTutorial

turing.ac.uk
@turinginst