Didactics of Data: Approaches to Teaching and Pedagogical Research for Applied ML

Dr. Rebecca Fiebrink
Reader, Creative Computing Institute, University of the Arts London

About me

- Building ML tools for creative practice & media since 2008

Wekinator mimic SoundControl InteractML

- Teaching ML to creative practitioners and other non-STEM students in various capacities since ~2011

Goldsmiths UNIVERSITY OF LONDON

ual: creative computing institute

kadenze MACHINE LEARNING FOR MUSICIANS AND ARTISTS

Future Learn Apply Creative Machine Learning

- Began more formal research on ML education in 2018

Fiebrink, R. 2019. “Machine Learning Education for Artists, Musicians, and Other Creative Practitioners.” ACM Trans. on Computing Education. November 2019, Article No. 31.

My students

- Undergraduate & masters students in creative computing, digital arts, games
- Professional artists/creators
- Often some competence in programming; possibly very little mathematics & computing theory
- Interest in building ML systems for use in music, art, design, gaming, etc.

In this talk:

Sharing aspects of teaching practice + research about teaching

- Establishing learning objectives
- Creating scaffolding tools for experiential learning
- Examples of formative assessments in reflective teaching practice
  - Initial brainstorming activity
  - ML algorithm “decision trees”
- The need for more research about ML teaching

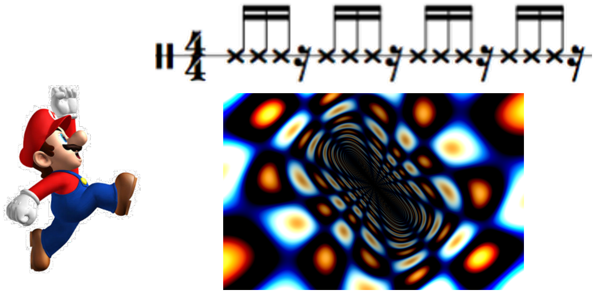
Establishing learning objectives

Focus on supervised learning

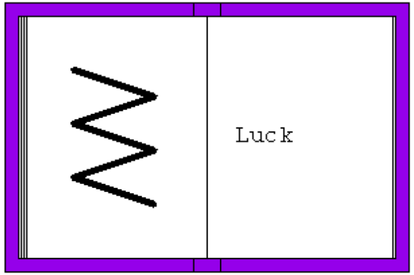
inputs → **ML-produced model** *classification , regression, temporal modeling* → outputs

Motion sensors, audio, video, environmental sensors, data from social media/game engine/etc. ...

Control over sound synthesis, procedural animation or design parameters, robots/physical computing systems, game engines, ...

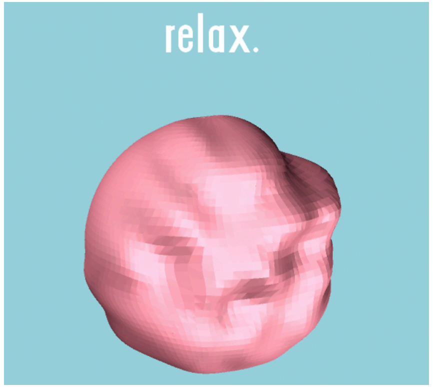


Example student projects

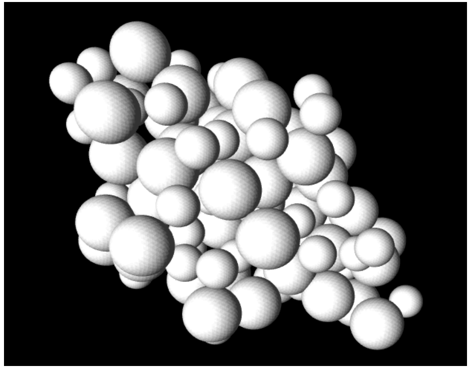


Cast spells in a game by drawing

Automatic voice harmoniser

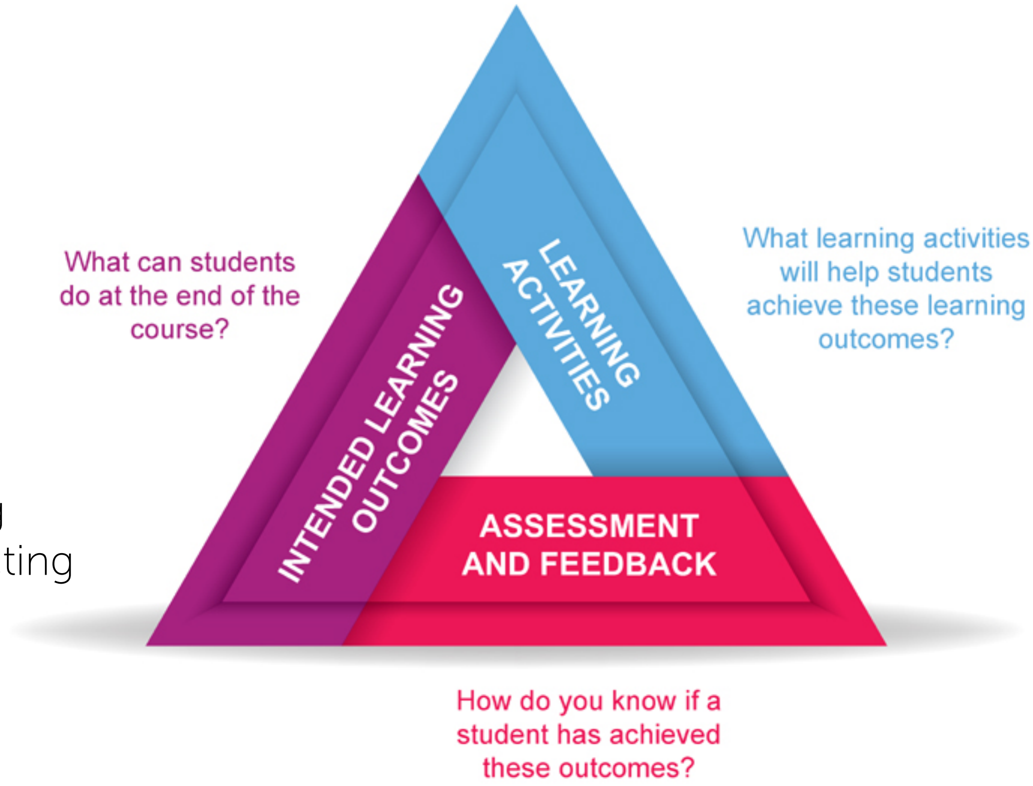


Digital “stress ball” responds to your face



Gesturally-controlled 3D graphic generation

Constructive Alignment



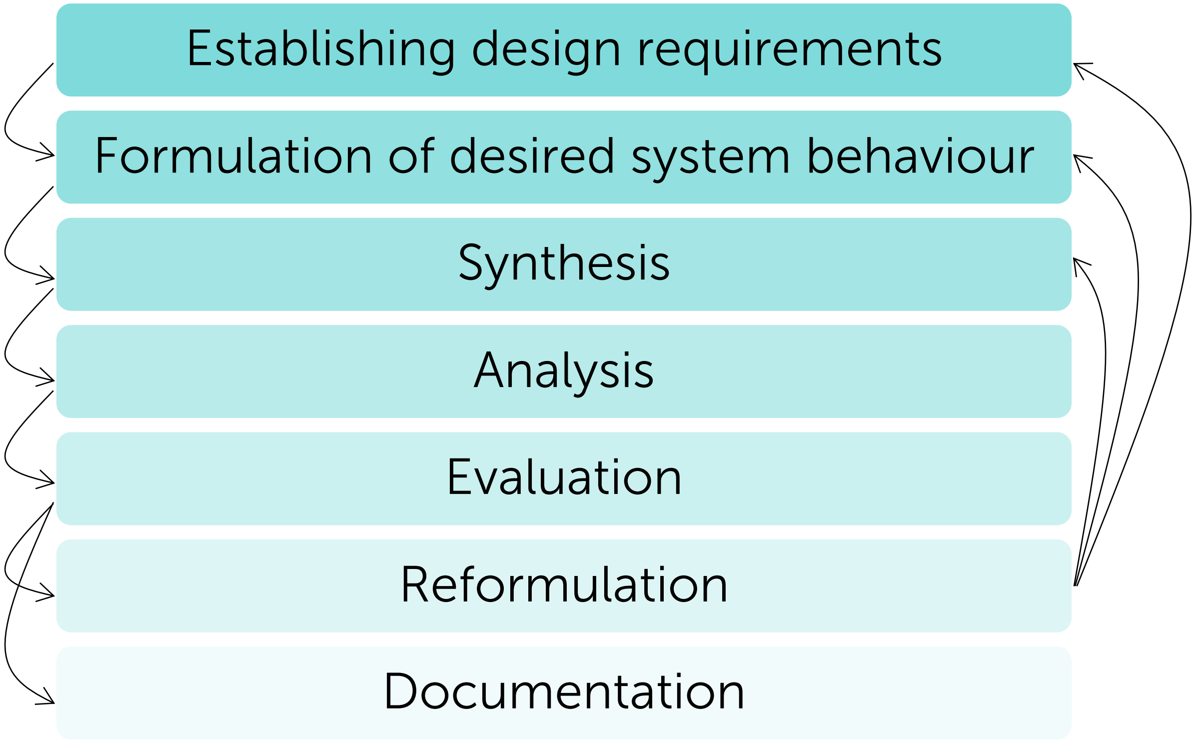
See Biggs, J. "Aligning teaching for constructing learning." The Higher Education Academy.

Photo: https://www.ntu.edu.sg/tlpd/tlr/DesigningYourCourse/OBTL/Pages/ConstructiveAlignment.aspx

Establishing LOs for Creative ML

- Overarching goal: enable students to create new systems using ML
- Requires more than creating an accurate model from a dataset
- ML is embedded in a design process
- Students confront questions while working with ML: What can I build? What *should* I build? How do I build it? How do I know if it works? How do I change it if it doesn't work, or if I want it to work differently?

“Design Operations” for engineering design & creative work



Howard, Culley & Dekoninck. "Describing the creative design process by the integration of engineering design and cognitive psychology literature." *Design Studies* 29 (2008): 160–180.

Proposed learning objectives for creative ML

Design Operation	Learning Objectives
Establishing design requirements, Formulation	LO1. Understand the structure of supervised learning problems & capability of supervised learning algorithms LO2. Identify feasible uses of ML in new projects, and map a new project idea onto the structure of supervised learning (input, output, training data, model)
Synthesis	LO3. Reason about properties of algorithms, data, and problem domain to make sensible choices about algorithms and features for a new project. LO4. Apply knowledge of ML workflows & practical skill with ML tools to create an ML model LO5. Use appropriate mechanisms to connect ML tools to other project HW/SW components
Analysis, evaluation	LO6. Choose appropriate methods to evaluate a trained model against design criteria relevant to the project, and apply these within the ML tool/library used
Reformulation	LO7. When a model does not satisfy design criteria, reason about appropriate mechanisms to improve it (e.g., changing training data, features, algorithm, other project components)
Other	LO8. Understand ways ML has been used in creative work, and draw on this to contextualise one's own work

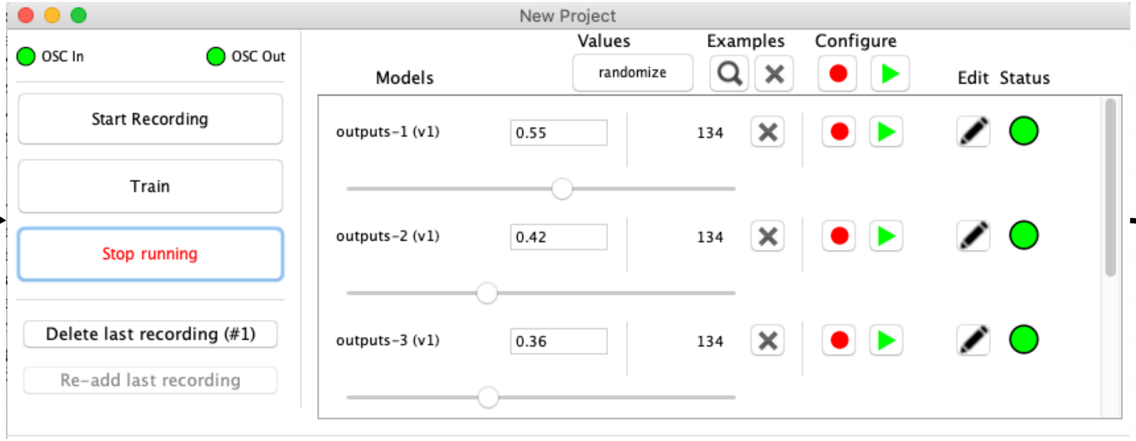
Syllabus maps learning objectives to lecture content and activities

MOOC Lectures (L) and Assignments (A) with Learning Objectives (LO) *Assignment marking approach listed as: auto-graded logs (AG), forum participation (F), peer feedback (P)*
L1: Introduction (LO1, LO2, LO5) Course overview. ML Pipeline. Wekinator. Using OSC to pass data to/from Wekinator.
A1: Getting Started with ML (LO1, LO2, LO4, LO5) A1.1: Follow detailed instructions to train and run a very simple ML system using Wekinator with online examples for input (e.g., webcam) and output (e.g., sound or animation). (AG) A1.2: Brainstorm 3 possible creative ML projects and describe on the forum. (F)
L2: Classification, Part 1 (LO1, LO2, LO3, LO4, LO8) What is classification? Nearest-neighbor and Decision Stump algorithms. Artistic applications of classification. What are features? Description and demo of a musical piece using classification.
A2: Creating Classifiers (LO1, LO2, LO3, LO4, LO5, LO6, LO7) A2.1: Engage in free-form experimentation with Classification Explorer, observing how decision boundaries change with different training sets and algorithms. (AG) A2.2: Generate training sets to re-create specific decision boundaries in Classification Explorer using kNN and Decision Stumps. (AG) A2.3: Use Wekinator and a real-time input of your choice to create an accurately controllable classifier, then

Scaffolding Tools for Experiential Learning

The Wekinator

"Input data" from feature extractor (e.g., from sensors, audio, video)



Model outputs sent to other software (e.g., animation, sound)

The Wekinator

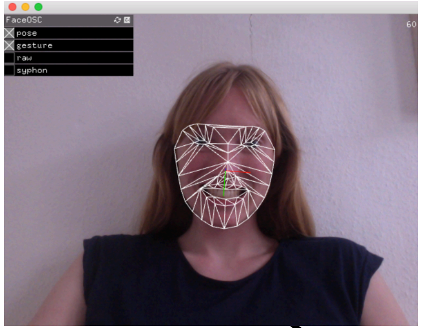
"Input data" from feature extractor (e.g., from sensors, audio, video)

In Wekinator

- Create training examples (including from real-time demonstrations)
- Build classification, regression, temporal models
- Run trained models on real-time data
- Manage ML: change algorithm, feature selection; compute CV accuracy; …

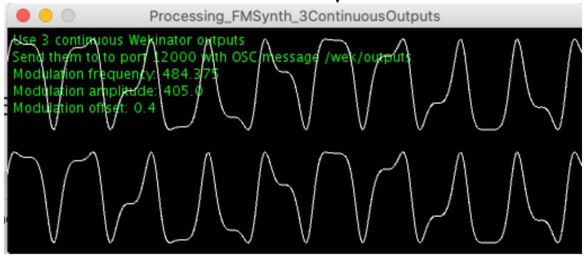
Model outputs sent to other software (e.g., animation, sound)

Creating complete interactive projects without code



25+ example feature extractors send data to Wekinator:
Audio, video, Arduino, game controllers, Leap, OpenFrameworks, Python, Processing, Max/MSP, …

Wekinator



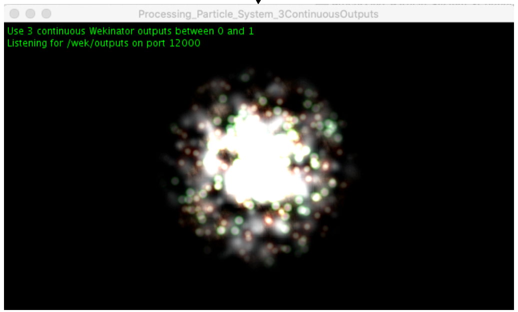
25+ example programs respond to Wekinator data:
Audio/music, animation, physical computing, games
OpenFrameworks, Processing, Max/MSP, MIDI, Ableton, …

Image: https://www.uni-weimar.de/kunst-und-gestaltung/wiki/GMU:Tutorials/Performance_Platform/Recognizing_Gestures_with_Wekinator

Creating complete interactive projects without code

25+ example feature extractors send data to Wekinator:
Audio, video, Arduino, game controllers, Leap,
OpenFrameworks, Python, Processing, Max/MSP, …

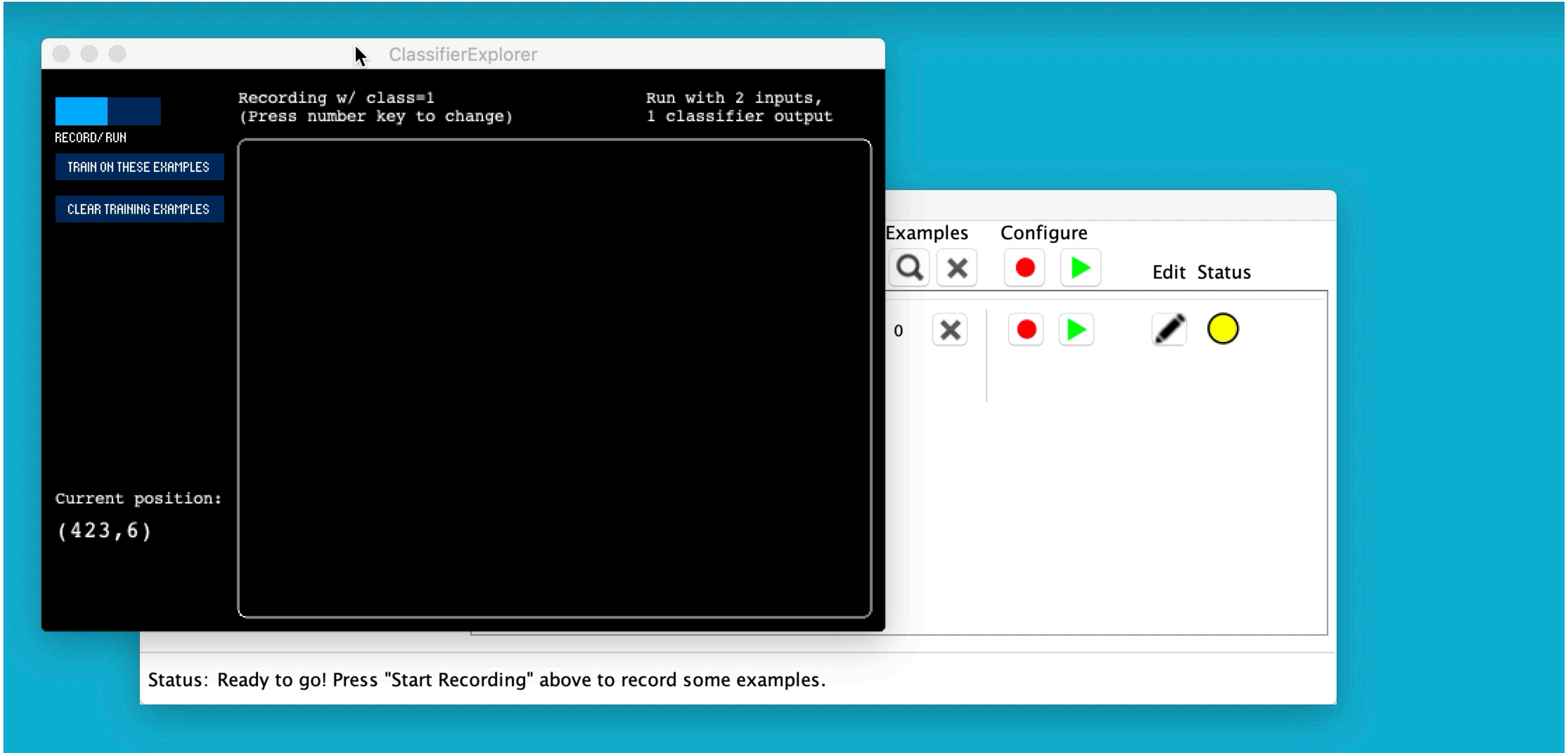
Wekinator



25+ example programs respond to Wekinator data:
Audio/music, animation, physical computing, games
OpenFrameworks, Processing, Max/MSP, MIDI, Ableton, …

Image: https://www.alibaba.com/product-detail/New-Leap-Motion-Controller-PC-Mac_50029970876.html

"Explorer" programs enable interrogation of learning algorithms, data, and models



Reflections on scaffolding tools

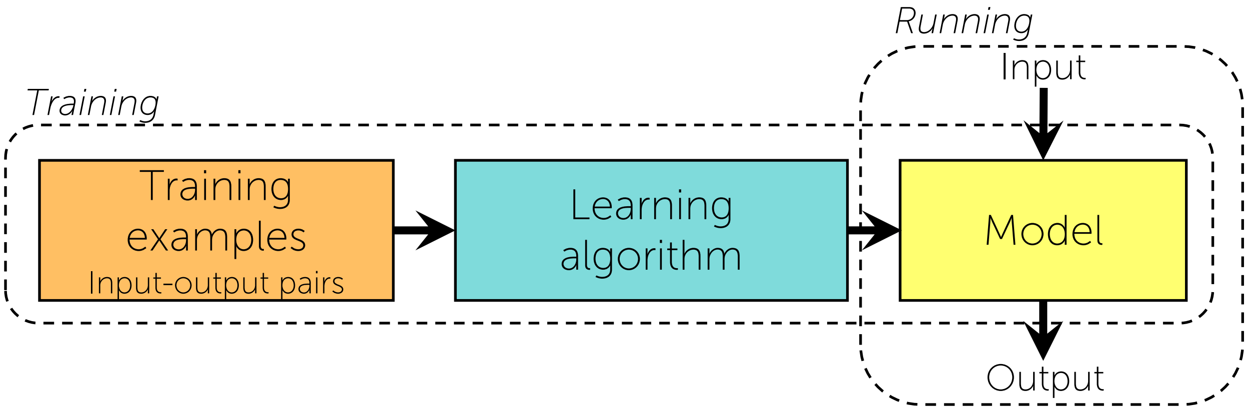
- Example inputs/outputs enabled focus on ML components in labs, while also working on domain-relevant tasks
- These were frequently used & modified in final creative projects
- "Explorer" programs supported hands-on investigation of interactions between noise, number of examples, unbalanced classes, algorithms & their parameters, and decision boundary location & shape
- "Explorers" occasionally led to inappropriate conclusions
    - e.g., that SVM and NN are sometimes 'hard' to train, because they may need 30+ examples (harder to add by hand)

Examples of formative assessment in reflective teaching practice

Example 1:

Beginning-of-class brainstorming

Lecture content: Intro to “ML pipeline”



Task: Train and run Wekinator with an example feature extractor (e.g., position of face tracked with webcam) & example output program (e.g., changing sound or animation). Then brainstorm a list of scenarios in which supervised learning could be used to make a piece of interactive art or music.

Proposed learning objectives for creative ML

Design Operation	**Learning Objectives**
Establishing design requirements, Formulation	LO1. Understand the structure of supervised learning problems & capability of supervised learning algorithms LO2. Identify feasible uses of ML in new projects, and map a new project idea onto the structure of supervised learning (input, output, training data, model)
Synthesis	LO3. Reason about properties of algorithms, data, and problem domain to make sensible choices about algorithms and features for a new project. LO4. Apply knowledge of ML workflows & practical skill with ML tools to create an ML model LO5. Use appropriate mechanisms to connect ML tools to other project HW/SW components
Analysis, evaluation	LO6. Choose appropriate methods to evaluate a trained model against design criteria relevant to the project, and apply these within the ML tool/library used
Reformulation	LO7. When a model does not satisfy design criteria, reason about appropriate mechanisms to improve it (e.g., changing training data, features, algorithm, other project components)
Other	LO8. Understand ways ML has been used in creative work, and draw on this to contextualise one's own work

Data analysis

- Starting questions of interest:
  - Are ideas feasible?
  - Do they differ substantially from examples described in class?
  - What might be challenges in turning these into real projects?
- Code each response
  - Feasibility (Y/N/Maybe)
  - Domain, challenges (open coding scheme)
  - Other emerging phenomena
- Work with subset of data to make analysis feasible

Outcomes & implications

- Diverse, interesting, and (mostly) feasible project ideas (27 of 30 ideas)

Sonification of income inequality as one travels the New York subway

Project visuals onto penitents at a Catholic church

Project visuals onto penitents at a Catholic church

Control live visuals with hand motions of Indian classical dancers

"ML pipeline" can be a good starting point for beginners to connect ML to real-world projects.

Outcomes & implications, continued

- Feature engineering was often a potential challenge (14 of 27)
  - *For such students, teaching about features, feature learning/transfer learning, feature engineering practices seems important.*
- Most scenarios did not strictly require ML (21 of 27)
  - *Students do not yet have a good understanding of what ML is capable of or when it is preferable to programming.*

Example 2:
Learning algorithm "decision trees"

Group assignment in week 3:

"Draw a decision tree (i.e., a flow-chart) for choosing a supervised learning algorithm for a new creative application. This chart should show someone how to decide which algorithm to use (of those seen so far). That is, the leaves at the bottom of the tree should correspond to one of the above algorithms (or possibly 'use something else' if none of these algorithms are appropriate)."

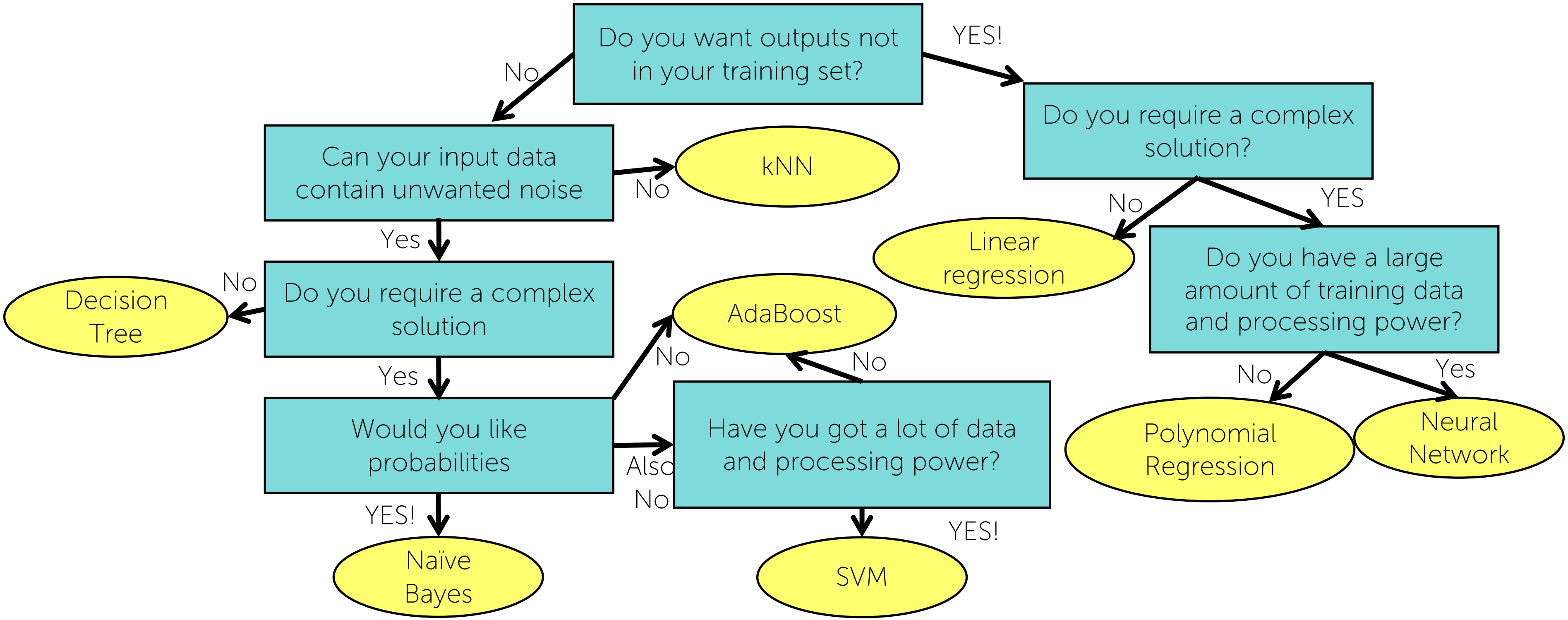
Proposed learning objectives for creative ML

Design Operation	Learning Objectives
Establishing design requirements, Formulation	LO1. Understand the structure of supervised learning problems & capability of supervised learning algorithms LO2. Identify feasible uses of ML in new projects, and map a new project idea onto the structure of supervised learning (input, output, training data, model)
Synthesis	LO3. Reason about properties of algorithms, data, and problem domain to make sensible choices about algorithms and features for a new project. LO4. Apply knowledge of ML workflows & practical skill with ML tools to create an ML model LO5. Use appropriate mechanisms to connect ML tools to other project HW/SW components
Analysis, evaluation	LO6. Choose appropriate methods to evaluate a trained model against design criteria relevant to the project, and apply these within the ML tool/library used
Reformulation	LO7. When a model does not satisfy design criteria, reason about appropriate mechanisms to improve it (e.g., changing training data, features, algorithm, other project components)
Other	LO8. Understand ways ML has been used in creative work, and draw on this to contextualise one’s own work

Data analysis

- Starting questions of interest:
  - Can students come up with a reasonable plan of action?
  - What do trees reveal about student learning and misconceptions?
- Code each leaf-level decision node for reasonableness (Y/N/Maybe)
- Code every decision node for type of decision (open), other arising phenomena

Example student tree



Outcomes & implications

- Students could often identify relevant factors (e.g., noise, dataset size, training time, unbalanced training data, …) and match them with algorithms
- Many (40 of 110) decision nodes were not actionable without empirical experimentation
  - *As-is, activity may promote inappropriate sense that knowledge and reasoning are sufficient to choose the 'right' approach*
  - *Future version of the assignment was modified to include "run an experiment" as a decision node*

The need for more research on ML teaching

We need to learn how to teach machine learning

Amy J. Ko
Aug 21, 2017 · 4 min read

Knowledge of how to apply machine learning to products is on high demand but low supply. Journalists write endlessly about it, employers want engineers who have it, college students want to learn it, and yet almost no one actually knows it.

Unfortunately, knowledge of *how to teach* machine learning effectively is also scarce, and has been for some time.

https://medium.com/bits-and-behavior/we-need-to-learn-how-to-teach-machine-learning-acc78bac3ff8

We need to learn how to teach machine learning

Amy J. Ko
Aug 21, 2017 · 4 min read

We still know little about what students need to know, how to teach it, and what knowledge teachers need to have to teach it successfully. To correct this, I argue that we need to discover the pedagogical content knowledge (PCK) necessary for teaching concepts in machine learning. PCK about machine learning includes:

- Useful representations for concepts in machine learning
- Effective analogies, examples, and explanations of machine learning
- Knowledge of which concepts in machine learning are difficult and why
- Knowledge of conceptions that learners bring to learning machine learning
- Methods of informally assessing knowledge of machine learning concepts
- Common mistakes the learners make when applying machine learning

Adding to Ko’s call for ML PCK:

- Useful representations for concepts in machine learning
- Effective analogies, examples, and explanations of machine learning
- Knowledge of which concepts in machine learning are difficult and why
- Knowledge of conceptions that learners bring to learning machine learning
- Methods of informally assessing knowledge of machine learning concepts
- Common mistakes the learners make when applying machine learning
- Why teach this subject? (see Grossman 1990)
- What should be taught?
- How can/should technology be used in this teaching? (see Mishra & Koehler 2006)

Thank you!

@rebeccafiebrink
www.wekinator.org