SUMMARY & REFLECTION

(the last one)

Christian Kaestner
Looking back at the semester (424 slides in 40 min)

Discussion of future of SE4AI

Feedback for future semesters
INTRODUCTION AND MOTIVATION

Christian Kaestner
LECTURE LOGISTICS DURING A PANDEMIC

If you can hear me, open the participant panel in Zoom and check "yes"
LEARNING GOALS

- Understand how AI components are parts of larger systems
- Illustrate the challenges in engineering an AI-enabled system beyond accuracy
- Explain the role of specifications and their lack in machine learning and the relationship to deductive and inductive reasoning
- Summarize the respective goals and challenges of software engineers vs data scientists
Data Scientists

Software Engineers
DATA SCIENTIST

- Often fixed dataset for training and evaluation (e.g., PBS interviews)
- Focused on accuracy
- Prototyping, often Jupyter notebooks or similar
- Expert in modeling techniques and feature engineering
- Model size, updateability, implementation stability typically does not matter

SOFTWARE ENGINEER

- Builds a product
- Concerned about cost, performance, stability, release time
- Identify quality through customer satisfaction
- Must scale solution, handle large amounts of data
- Detect and handle mistakes, preferably automatically
- Maintain, evolve, and extend the product over long periods
- Consider requirements for security, safety, fairness
QUALITIES OF INTEREST ("ILITIES")

- Quality is about more than the absence of defects
- Quality in use (effectiveness, efficiency, satisfaction, freedom of risk, ...)
- Product quality (functional correctness and completeness, performance efficiency, compatibility, usability, dependability, scalability, security, maintainability, portability, ...)
- Process quality (manageability, evolvability, predictability, ...)

"Quality is never an accident; it is always the result of high intention, sincere effort, intelligent direction and skillful execution; it represents the wise choice of many alternatives." (many attributions)
Speaker 5  ▶ 07:44

Yeah. So there’s a slight story behind that. So back when I was in, uh, Undergrad, I wrote a program for myself to measure the amount of time I did data entry from my father’s business and I was on windows at the time and there wasn’t a function called time dot [inaudible] time, uh, which I needed to parse dates to get back to time, top of representation, uh, I figured out a way to do it and I gave it to what’s called the python cookbook because it just seemed like something other people could use. So it was just trying to be helpful. Uh, subsequently I had to figure out how to make it work because I didn’t really have to. Basically, it bothered me that you had to input all the locale information and I figured out how to do it over the subsequent months. And actually as a graduation gift from my Undergrad, the week following, I solved it and wrote it all out.

Speaker 5  ▶ 08:38

And I asked, uh, Alex Martelli, the editor of the Python Cookbook, which had published my original recipe, a, how do I get this into python? I think it might help others. And he said, well, I don’t think you’ll like this because I don’t think it’s a good idea. How did we do on your transcript? ★★★★★
SYLLABUS AND CLASS STRUCTURE

17-445/17-645, Summer 2020, 12 units
Tuesday/Wednesday 3-4:20, here on zoom
TEXTBOOK


by Geoff Hulten

https://www.buildingintelligentsystems.com/

Most chapters assigned at some point in the semester

Supplemented with research articles, blog posts, videos, podcasts, ...

Electronic version in the library
INTRODUCTIONS

Let's go around the "room" for introductions:

• Your (preferred name)
• In two sentences your software engineering background and goals
• In two sentences your data science background, if any, and goals
• One topic you are particularly interested in, if any?
CORRECTNESS AND SPECIFICATIONS

DEDUCTIVE VS. INDUCTIVE REASONING
WHO IS TO BLAME?

```java
Algorithms.shortestDistance(g, "Tom", "Anne");
> ArrayOutOfBoundsException

Algorithms.shortestDistance(g, "Tom", "Anne");
> -1
```
SPECIFICATIONS IN MACHINE LEARNING?

```java
/**
   * ????
   */
String transcribe(File audioFile);
```
(Daniel Miessler, CC SA 2.0)
RESULTING SHIFT IN DESIGN THINKING?

From deductive reasoning to inductive reasoning...

From clear specifications to goals...

From guarantees to best effort...

What does this mean for software engineering?

For decomposing software systems?

For correctness of AI-enabled systems?

For safety?

For design, implementation, testing, deployment, operations?
HOMEWORK 1: CASE STUDY

Engineering issues in detecting malicious apps
ARTIFICIAL INTELLIGENCE FOR SOFTWARE ENGINEERS

(Part 1: Supervised Machine Learning and Notebooks)

Christian Kaestner


LEARNING GOALS

- Understand how machine learning learns models from labeled data (basic mental model)
- Explain the steps of a typical machine learning pipeline and their responsibilities and challenges
- Understand the role of hyper-parameters
- Appropriately use vocabulary for machine learning concepts
- Apply steps of a machine-learning pipeline to build a simple model from static labeled data
- Evaluate a machine-learned classifier using cross-validation
- Explain the benefits and drawbacks of notebooks
- Demonstrate effective use of computational notebooks
DEFINING MACHINE LEARNING (SIMPLIFIED)

learn a function (called model)

\[ f(x_1, x_2, x_3, \ldots, x_n) \rightarrow y \]

by observing data

**Examples:**

- Detecting cancer in an image
- Transcribing an audio file
- Detecting spam
- Predicting recidivism
- Detect suspicious activity in a credit card

Typically used when writing that function manually is hard because the problem is hard or complex.
RUNNING EXAMPLE: HOUSE PRICE ANALYSIS

Given data about a house and its neighborhood, what is the likely sales price for this house?

\[ f(size, rooms, tax, neighborhood, \ldots) \rightarrow price \]
$f(x) = \alpha + \beta \times x$
## DECISION TREES

### Data Table

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>no</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>yes</td>
</tr>
<tr>
<td>overcast</td>
<td>mild</td>
<td>high</td>
<td>true</td>
<td>yes</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>normal</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>mild</td>
<td>high</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>cool</td>
<td>normal</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>no</td>
</tr>
<tr>
<td>rainy</td>
<td>mild</td>
<td>normal</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>rainy</td>
<td>mild</td>
<td>high</td>
<td>true</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>true</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>mild</td>
<td>high</td>
<td>false</td>
<td>no</td>
</tr>
<tr>
<td>sunny</td>
<td>cool</td>
<td>normal</td>
<td>false</td>
<td>yes</td>
</tr>
<tr>
<td>sunny</td>
<td>mild</td>
<td>normal</td>
<td>true</td>
<td>yes</td>
</tr>
</tbody>
</table>

### Decision Tree

\[ f(\text{Outlook, Temperature, Humidity, Windy}) = \]

```
  Outlook
    Sunny   Overcast   Rainy
      Windy
        true  false  high  Normal
          No  No  No  Yes
```
The tree perfectly fits the data, except on overcast, hot and humid days without wind, where there is not enough data to distinguish 3 outcomes.

Not obvious that this tree will generalize well.
ON TERMINOLOGY

- The decisions in a model are called *model parameter* of the model (constants in the resulting function, weights, coefficients), their values are usually learned from the data.
- The parameters to the learning algorithm that are not the data are called *model hyperparameters*.
- Degrees of freedom ~ number of model parameters.

```python
// max_depth and min_support are hyperparameters
def learn_decision_tree(data, max_depth, min_support):
    Model = ...

// A, B, C are model parameters of model f
def f(outlook, temperature, humidity, windy):
    if A==outlook
        return B*temperature + C*windy > 10
```
SEPARATE TRAINING AND VALIDATION DATA

Always test for generalization on \textit{unseen} validation data

Accuracy on training data (or similar measure) used during learning to find model parameters

\begin{verbatim}
train_xs, train_ys, valid_xs, valid_ys = split(all_xs, all_ys)
model = learn(train_xs, train_ys)
accuracy_train = accuracy(model, train_xs, train_ys)
accuracy_valid = accuracy(model, valid_xs, valid_ys)
\end{verbatim}

\texttt{accuracy\_train} >> \texttt{accuracy\_valid} = sign of overfitting
DETECTING OVERFITTING

Change hyperparameter to detect training accuracy (blue)/validation accuracy (red) at different degrees of freedom

(CC SA 3.0 by Dake)

demo time
ACADEMIC ESCALATION: OVERFITTING ON BENCHMARKS

(Figure by Andrea Passerini)
SIMILAR TO SPIRAL PROCESS MODEL OR AGILE?

1. Determine objectives
2. Identify and resolve risks
3. Development and Test
4. Plan the next iteration

---

(CC BY-SA 4.0, Lakeworks)
DATA SCIENCE IS ITERATIVE AND EXPLORATORY

<table>
<thead>
<tr>
<th></th>
<th>First 2 Hours</th>
<th>Second 2 Hours</th>
<th>Final Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAP1</td>
<td></td>
<td></td>
<td>84.7%</td>
</tr>
<tr>
<td>TAP2</td>
<td>X</td>
<td>X</td>
<td>75.3%</td>
</tr>
<tr>
<td>TAP3</td>
<td></td>
<td></td>
<td>78.3%</td>
</tr>
<tr>
<td>TAP4</td>
<td></td>
<td></td>
<td>82.9%</td>
</tr>
<tr>
<td>TAP5</td>
<td></td>
<td></td>
<td>84.7%</td>
</tr>
<tr>
<td>TAP6</td>
<td></td>
<td></td>
<td>78.0%</td>
</tr>
<tr>
<td>TAP7</td>
<td></td>
<td></td>
<td>56.9%</td>
</tr>
<tr>
<td>TAP8</td>
<td></td>
<td></td>
<td>22.8%</td>
</tr>
<tr>
<td>TAP9</td>
<td></td>
<td></td>
<td>78.8%</td>
</tr>
<tr>
<td>TAP10</td>
<td></td>
<td></td>
<td>84.4%</td>
</tr>
</tbody>
</table>

COMPUTATIONAL NOTEBOOKS

- Origins in "literal programming", interleaving text and code, treating programs as literature (Knuth'84)
- First notebook in Wolfram Mathematica 1.0 in 1988
- Document with text and code cells, showing execution results under cells
- Code of cells is executed, per cell, in a kernel
- Many notebook implementations and supported languages, Python + Jupyter currently most popular

Data was preprocessed externally, identifying the time at a given day when the light was first turned on (12pm). Weather and sunrise information is not included here, though that'd be important. If the light was on this morning (quite common), 0 is recorded.

```python
# load data collected from teaml
import pandas as pd
url = 'http://128.2.25.78:8880/private/log1.clean'
df = pd.read_csv(url)
df.head()
```

<table>
<thead>
<tr>
<th>dayIdx</th>
<th>user</th>
<th>userAvgTime</th>
<th>location</th>
<th>dow</th>
<th>isWeekend</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Pittsburgh66Correy</td>
<td>7.045001</td>
<td>Pittsburgh</td>
<td>6</td>
<td>True</td>
<td>0.000000</td>
</tr>
<tr>
<td>1</td>
<td>Pittsburgh66Correy</td>
<td>7.045001</td>
<td>Pittsburgh</td>
<td>7</td>
<td>True</td>
<td>6.883333</td>
</tr>
<tr>
<td>2</td>
<td>Pittsburgh66Correy</td>
<td>7.045001</td>
<td>Pittsburgh</td>
<td>1</td>
<td>False</td>
<td>6.816667</td>
</tr>
<tr>
<td>3</td>
<td>Pittsburgh66Correy</td>
<td>7.045001</td>
<td>Pittsburgh</td>
<td>2</td>
<td>False</td>
<td>7.383333</td>
</tr>
<tr>
<td>4</td>
<td>Pittsburgh66Correy</td>
<td>7.045001</td>
<td>Pittsburgh</td>
<td>3</td>
<td>False</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

```python
[ ] # just data encoding and splitting X and Y

X = df.drop(['time'], axis=1)
YnonZero = df['time'] > 0
Y = df['time']

from sklearn import preprocessing
X = preprocessing.LabelEncoder().fit_transform(X)
```

```python
X=X.apply(preprocessing.LabelEncoder().fit_transform)
```
ARTIFICIAL INTELLIGENCE FOR SOFTWARE ENGINEERS

(Part 2: Deep Learning, Symbolic AI)

Christian Kaestner


LEARNING GOALS

- Give an overview of different AI problems and approaches
- Explain at high level how deep learning works
- Describe key characteristics of symbolic AI techniques and when to use them
Artificial Intelligence:

computers acting humanly / thinking humanly / thinking rationally / acting rationally -- Russel and Norvig, 2003

Machine Learning:

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E. -- Tom Mitchell, 1997

Deep Learning:

specific learning technique based on neural networks
ARTIFICIAL INTELLIGENCE

• Acting humanly: Turing test approach, requires natural language processing, knowledge representation, automated reasoning, machine learning, maybe vision and robotics
• Thinking humanly: mirroring human thinking, cognitive science
• Thinking rationally: law of thoughts, logic, patterns and structures
• Acting rationally: rational agents interacting with environments

• problem solving (e.g., search, constraint satisfaction)
• knowledge, reasoning, planning (e.g., logic, knowledge representation, probabilistic reasoning)
• learning (learning from examples, knowledge in learning, reinforcement learning)
• communication, perceiving, and acting (NLP, vision, robotics)

COMMON PROBLEM CLASSES

- Classification
- Probability estimation
- Regression
- Ranking
- Hybrids
LEARNING PARADIGMS

- Supervised learning -- labeled training data provided
- Unsupervised learning -- training data without labels
- Reinforcement learning -- agents learning from interacting with an environment
NEURAL NETWORKS
THRESHOLD LOGIC UNIT / PERCEPTRON

computing weighted sum of inputs + step function

\[ z = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n = \mathbf{x}^T \mathbf{w} \]

e.g., step: \( \phi(z) = \text{if } (z < 0) \ 0 \ \text{else } 1 \)
\[ f_{W_h, b_h, W_o, b_o}(X) = \phi(W_o \cdot \phi(W_h \cdot X + b_h) + b_o) \]

(matrix multiplications interleaved with step function)
EXAMPLE SCENARIO

- MNIST Fashion dataset of 70k 28x28 grayscale pixel images, 10 output classes
NETWORK SIZE

- 50 Layer ResNet network -- classifying 224x224 images into 1000 categories
  - 26 million weights, computes 16 million activations during inference, 168 MB to store weights as floats
- OpenAI’s GPT-2 (2019) -- text generation
  - 48 layers, 1.5 billion weights (~12 GB to store weights)
  - released model reduced to 117 million weights
  - trained on 7-8 GPUs for 1 month with 40GB of internet text from 8 million web pages
CLASSIC SYMBOLIC AI

(Good Old-Fashioned Artificial Intelligence)
BOOLEAN SATISFIABILITY

Given a propositional formula over boolean variables, is there an assignment such that the formula evaluates to true?

\[(a \lor b) \land (\neg a \lor c) \land \neg b\]

decidable, np complete, lots of search heuristics
ENCODING PROBLEMS

Linux/arm 3.8.13 Kernel Configuration

Option
- General setup
  - IRQ subsystem
  - Timers subsystem
  - CPU/Task time and stats accounting
  - RCU Subsystem
  - Control Group support
    - Group CPU scheduler
  - Namespaces support
  - Configure standard kernel features
  - Kernel Performance Events API
  - GCOV-based kernel profiling
  - Xtree-based kernel profiling
  - Enable loadable module support
  - Enable the block layer
    - Partition Types
    - IO Schedulers
  - System Type
    - TI OMAP Common Features
    - TI OMAP2/3/4 Specific Features
  - Bus support
    - PCCard (PCMCIA/CardBus)
  - Kernel Features
  - Boot options
  - CPU Power Management

Option
- ..
- Keyboards
  - ADP5588/87 I2C QWERTY Keypad and IO Expander
  - ADP5585/ADP5589 I2C QWERTY Keypad and IO Expander
  - AT keyboard
  - Atmel AT42QT1070 Touch Sensor Chip
  - Atmel AT42QT2160 Touch Sensor Chip
  - DECstation/VAXstation LK201/LK401 keyboard
  - GPIO Buttons
    - GPIO Buttons 2 (NEW)
    - Pollled GPIO buttons
    - TCA6416/TCA6408A Keypad Support

GPIO Buttons 2 (KEYBOARD_GPIO2)

CONFIG_KEYBOARD_GPIO2:

This driver implements support for buttons connected to GPIO pins of various CPUs (and some other chips).

Say Y here if you want an A in ECE497

To compile this driver as a module, choose M here: the module will be called gpio_keys.

Symbols KEYBOARD_GPIO2 [...]

5.14
CONSTRAINT SATISFACTION PROBLEMS, SMT

Generalization beyond boolean options, numbers, strings, additions, optimization

Example: Job Scheduling

Tasks for assembling a car: \{ t1, t2, t3, t4, t5, t6 \}; values denoting start time

\[
\max 30 \text{ min: } \forall n \ t_n < 30
\]

- t2 needs to be after t1, t1 takes 10 min: \( t_1 + 10 \leq t_2 \)
- t3 and t4 needs to be after t2, take 2 min: \((t_2 + 2 \leq t_3) \land (t_2 + 2 \leq t_4)\)
- t5 and t6 (5 min each) should not overlap: \((t_5 + 5 \leq t_6) \lor (t_6 + 5 \leq t_5)\)

Goal: find valid assignment for all start times, or find valid assignment minimizing the latest start time
class Person {
    val smokes = Flip(0.6)
}
def smokingInfluence(pair: (Boolean, Boolean)) =
    if (pair._1 == pair._2) 3.0; else 1.0

val alice, bob, clara = new Person
val friends = List((alice, bob), (bob, clara))
cguna.smokes.observe(true)
for { (p1, p2) <- friends }
    ^^p1.smokes, p2.smokes).setConstraint(smokingInfluence)
...
println("Probability of Alice smoking: " +
    alg.probability(alice.smokes, true))
Answering queries about probabilistic models

```java
println("Probability of burglary: " + alg.probability(burglary, true))
println("Probability of Alice smoking: " + alg.probability(alice.smokes, true))
```

- Analytical probabilistic reasoning (e.g., variable elimination Bayes' rule) -- precise result, guarantees
- Approximation (e.g., belief propagation)
- Sampling (e.g., Markov chain Monte Carlo) -- probabilistic guarantees
Predicting Movie Popularity (2 weeks)
MODEL QUALITY

Christian Kaestner

Required reading:

LEARNING GOALS

- Select a suitable metric to evaluate prediction accuracy of a model and to compare multiple models
- Select a suitable baseline when evaluating model accuracy
- Explain how software testing differs from measuring prediction accuracy of a model
- Curate validation datasets for assessing model quality, covering subpopulations as needed
- Use invariants to check partial model properties with automated testing
- Develop automated infrastructure to evaluate and monitor model quality
THIS LECTURE

FIRST PART: MEASURING PREDICTION ACCURACY
the data scientist's perspective

SECOND PART: LEARNING FROM SOFTWARE TESTING
how software engineering tools may apply to ML
“Programs which were written in order to determine the answer in the first place. There would be no need to write such programs, if the correct answer were known” (Weyuker, 1982).
THE SYSTEMS PERSPECTIVE

System is more than the model

Includes deployment, infrastructure, user interface, data infrastructure, payment services, and often much more

Systems have a goal:

- maximize sales
- save lifes
- entertainment
- connect people

Models can help or may be essential in those goals, but are only one part

*Today: Narrow focus on prediction accuracy of the model*
CANCER PREDICTION WITHIN A HEALTHCARE APPLICATION

(CC BY-SA 4.0, Martin Sauter)
### CONFUSION/ERROR MATRIX

<table>
<thead>
<tr>
<th></th>
<th>Actually A</th>
<th>Actually B</th>
<th>Actually C</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI predicts A</td>
<td>10</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>AI predicts B</td>
<td>3</td>
<td>24</td>
<td>10</td>
</tr>
<tr>
<td>AI predicts C</td>
<td>5</td>
<td>22</td>
<td>82</td>
</tr>
</tbody>
</table>

Accuracy = correct predictions (diagonal) out of all predictions

Example's accuracy = \[
\frac{10+24+82}{10+6+2+3+24+10+5+22+82} = .707
\]
IS 99% ACCURACY GOOD?

-> depends on problem; can be excellent, good, mediocre, terrible

10% accuracy can be good on some tasks (information retrieval)

Always compare to a base rate!

Reduction in error = \( \frac{(1 - \text{accuracy}_{\text{baseline}}) - (1 - \text{accuracy}_f)}{1 - \text{accuracy}_{\text{baseline}}} \)

- from 99.9% to 99.99% accuracy = 90% reduction in error
- from 50% to 75% accuracy = 50% reduction in error
## TYPES OF MISTAKES

Two-class problem of predicting event A:

<table>
<thead>
<tr>
<th></th>
<th>Actually A</th>
<th>Actually not A</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AI predicts A</strong></td>
<td>True Positive (TP)</td>
<td>False Positive (FP)</td>
</tr>
<tr>
<td><strong>AI predicts not A</strong></td>
<td>False Negative (FN)</td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>

- True positives and true negatives: correct prediction
- False negatives: wrong prediction, miss, Type II error
- False positives: wrong prediction, false alarm, Type I error
### MULTI-CLASS PROBLEMS VS TWO-CLASS PROBLEM

#### Table 1: Classification Accuracy

<table>
<thead>
<tr>
<th></th>
<th>Actually A</th>
<th>Actually B</th>
<th>Actually C</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI predicts A</td>
<td>10</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>AI predicts B</td>
<td>3</td>
<td>24</td>
<td>10</td>
</tr>
<tr>
<td>AI predicts C</td>
<td>5</td>
<td>22</td>
<td>82</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AI predicts A</td>
<td>10</td>
<td>8</td>
<td>24</td>
<td>13</td>
</tr>
<tr>
<td>AI predicts not A</td>
<td>8</td>
<td>138</td>
<td>28</td>
<td>99</td>
</tr>
</tbody>
</table>

11
(CC BY-SA 4.0 by Walber)
FALSE POSITIVES AND FALSE NEGATIVES EQUALLY BAD?

Consider:

- Recognizing cancer
- Suggesting products to buy on e-commerce site
- Identifying human trafficking at the border
- Predicting high demand for ride sharing services
- Predicting recidivism chance
- Approving loan applications

No answer vs wrong answer?
RECEIVER OPERATING CHARACTERISTIC (ROC) CURVES

(CC BY-SA 3.0 by BOR)
COMPARING PREDICTED AND EXPECTED OUTCOMES

Mean Absolute Percentage Error

\[ MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| \]

\((A_t\) actual outcome, \(F_t\) predicted outcome, for row \(t\))

Compute relative prediction error per row, average over all rows

<table>
<thead>
<tr>
<th>Rooms</th>
<th>Crime Rate</th>
<th>...</th>
<th>Predicted Price</th>
<th>Actual Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>.01</td>
<td></td>
<td>230k</td>
<td>250k</td>
</tr>
<tr>
<td>4</td>
<td>.01</td>
<td></td>
<td>530k</td>
<td>498k</td>
</tr>
<tr>
<td>2</td>
<td>.03</td>
<td></td>
<td>210k</td>
<td>211k</td>
</tr>
<tr>
<td>2</td>
<td>.02</td>
<td></td>
<td>219k</td>
<td>210k</td>
</tr>
</tbody>
</table>

\[ \text{MAPE} = \frac{1}{4} \left( \frac{20}{250} + \frac{32}{498} + \frac{1}{211} + \frac{9}{210} \right) \]

\[ = \frac{1}{4} \left( 0.08 + 0.064 + 0.005 + 0.043 \right) = 0.048 \]
EVALUATING RANKINGS

Ordered list of results, true results should be ranked high

Common in information retrieval (e.g., search engines) and recommendations

Mean Average Precision

MAP@K = precision in first $K$ results

Averaged over many queries

<table>
<thead>
<tr>
<th>Rank</th>
<th>Product</th>
<th>Correct?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Juggling clubs</td>
<td>true</td>
</tr>
<tr>
<td>2</td>
<td>Bowling pins</td>
<td>false</td>
</tr>
<tr>
<td>3</td>
<td>Juggling balls</td>
<td>false</td>
</tr>
<tr>
<td>4</td>
<td>Board games</td>
<td>true</td>
</tr>
<tr>
<td>5</td>
<td>Wine</td>
<td>false</td>
</tr>
<tr>
<td>6</td>
<td>Audiobook</td>
<td>true</td>
</tr>
</tbody>
</table>

MAP@1 = 1, MAP@2 = 0.5, MAP@3 = 0.33, ...

Remember to compare against baselines! Baseline for shopping recommendations?
MODEL QUALITY IN NATURAL LANGUAGE PROCESSING?

Highly problem dependent:

- Classify text into positive or negative -> classification problem
- Determine truth of a statement -> classification problem
- Translation and summarization -> comparing sequences (e.g., ngrams) to human results with specialized metrics, e.g., BLEU and ROUGE
- Modeling text -> how well its probabilities match actual text, e.g., likelihood or perplexity
ANALOGY TO SOFTWARE TESTING

(this gets messy)
<table>
<thead>
<tr>
<th>Rooms</th>
<th>Crime Rate</th>
<th>...</th>
<th>Actual Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>.01</td>
<td>...</td>
<td>250k</td>
</tr>
<tr>
<td>4</td>
<td>.01</td>
<td>...</td>
<td>498k</td>
</tr>
<tr>
<td>2</td>
<td>.03</td>
<td>...</td>
<td>211k</td>
</tr>
<tr>
<td>2</td>
<td>.02</td>
<td>...</td>
<td>210k</td>
</tr>
</tbody>
</table>

Fail the entire test suite for one wrong prediction?
**THE ORACLE PROBLEM**

*How do we know the expected output of a test?*

```java
assertEquals(??, factorPrime(15485863));
```

- Manually construct input-output pairs (does not scale, cannot automate)
- Comparison against gold standard (e.g., alternative implementation, executable specification)
- Checking of global properties only -- crashes, buffer overflows, code injections
- Manually written assertions -- partial specifications checked at runtime
DIFFERENT EXPECTATIONS FOR PREDICTION ACCURACY

- Not expecting that all predictions will be correct (80% accuracy may be very good)
- Data may be mislabeled in training or validation set
- There may not even be enough context (features) to distinguish all training outcomes

- Lack of specifications
- A wrong prediction is not necessarily a bug
ANALOGY OF PERFORMANCE TESTING?

- Performance tests are not precise (measurement noise)
  - Averaging over repeated executions of the same test
  - Commonly using diverse benchmarks, i.e., multiple inputs
  - Need to control environment (hardware)
- No precise specification
  - Regression tests
  - Benchmarking as open-ended comparison
  - Tracking results over time

```java
@Test(timeout=100)
public void testCompute() {
    expensiveComputation(...);
}
```
MACHINE LEARNING IS REQUIREMENTS ENGINEERING

(my pet theory)

see also https://medium.com/@ckaestne/machine-learning-is-requirements-engineering-8957aee55ef4
VALIDATION VS VERIFICATION

Validation

- Interviews, Req., Synthesis

- Specification

- More Specifications

- Coding

- Implementation

Verification

- Machine Learning

- ML Model

- More Specifications (e.g. Fairness, Safeguards)

- Generate/Pickle

- ML Model

- Implementation

- Requirements Analysis

- Coding
EXAMPLE AND DISCUSSION

IF age between 18–20 and sex is male THEN predict arrest
ELSE IF age between 21–23 and 2–3 prior offenses THEN predict arrest
ELSE IF more than three priors THEN predict arrest
ELSE predict no arrest

Model learned from gathered data (~ interviews, sufficient? representative?)

Cannot equally satisfy all stakeholders, conflicting goals; judgement call, compromises, constraints

Implementation is trivial/automatically generated

Does it meet the users' expectations?

Is the model compatible with other specifications? (fairness, robustness)

What if we cannot understand the model? (interpretability)
CURATING VALIDATION

DATA

(Learning from Software Testing?)
VALIDATION DATA REPRESENTATIVE?

- Validation data should reflect usage data
- Be aware of data drift (face recognition during pandemic, new patterns in credit card fraud detection)
- "Out of distribution" predictions often low quality (it may even be worth to detect out of distribution data in production, more later)
INDEPENDENCE OF DATA: TEMPORAL
"Call mom" "What's the weather tomorrow?" "Add asafetida to my shopping list"
There Is a Racial Divide in Speech-Recognition Systems, Researchers Say: Technology from Amazon, Apple, Google, IBM and Microsoft misidentified 35 percent of words from people who were black. White people fared much better. -- *NYTimes* March 2020
IDENTIFY IMPORTANT INPUTS

Curate Validation Data for Specific Problems and Subpopulations:

- **Regression testing**: Validation dataset for important inputs ("call mom") -- expect very high accuracy -- closest equivalent to **unit tests**
- **Uniformness/fairness testing**: Separate validation dataset for different subpopulations (e.g., accents) -- expect comparable accuracy
- **Setting goals**: Validation datasets for challenging cases or stretch goals -- accept lower accuracy

Derive from requirements, experts, user feedback, expected problems etc. Think **blackbox testing**.
BLACK-BOX TESTING TECHNIQUES AS INSPIRATION?

- Boundary value analysis
- Partition testing & equivalence classes
- Combinatorial testing
- Decision tables

Use to identify subpopulations (validation datasets), not individual tests.
EXAMPLES OF INVARIANTS

- Credit rating should not depend on gender:
  \[ \forall x. f(x[\text{gender} \leftarrow \text{male}]) = f(x[\text{gender} \leftarrow \text{female}]) \]

- Synonyms should not change the sentiment of text:
  \[ \forall x. f(x) = f(\text{replace}(x, "is not", "isn't")) \]

- Negation should swap meaning:
  \[ \forall x \in "X \text{ is } Y". f(x) = 1 - f(\text{replace}(x, " is ", " is not ")) \]

- Robustness around training data:
  \[ \forall x \in \text{training data}. \forall y \in \text{mutate}(x, \delta). f(x) = f(y) \]

- Low credit scores should never get a loan (sufficient conditions for classification, "anchors"):
  \[ \forall x. x. \text{score} < 649 \Rightarrow \neg f(x) \]

Identifying invariants requires domain knowledge of the problem!
METAMORPHIC TESTING

Formal description of relationships among inputs and outputs (Metamorphic Relations)

In general, for a model $f$ and inputs $x$ define two functions to transform inputs and outputs $g_I$ and $g_O$ such that:

$$\forall x. f(g_I(x)) = g_O(f(x))$$

e.g. $g_I(x) = \text{replace}(x, \text{” is ”}, \text{” is not ”})$ and $g_O(x) = \neg x$
ONE MORE THING: SIMULATION-BASED TESTING

- Derive input-output pairs from simulation, esp. in vision systems
- Example: Vision for self-driving cars:
  - Render scene -> add noise -> recognize -> compare recognized result with simulator state
- Quality depends on quality of the simulator and how well it can produce inputs from outputs:
  - examples: render picture/video, synthesize speech, ...
  - Less suitable where input-output relationship unknown, e.g., cancer detection, housing price prediction, shopping recommendations

CONTINUOUS INTEGRATION FOR MODEL QUALITY
Technical Contribution
Provide guidelines on how large N is in a declarative, rigorous, but still practical way, enabled by novel system optimization techniques.

Example Test Condition
New model has at least 2% higher accuracy, estimated within 1% error, with probability 0.9999.

Renggli et. al, Continuous Integration of Machine Learning Models with ease.ml/ci: Towards a Rigorous Yet Practical Treatment, SysML 2019

SPECIALIZED CI SYSTEMS
FROM MODELS TO AI-ENABLED SYSTEMS

Christian Kaestner

LEARNING GOALS

- Explain how machine learning fits into the larger picture of building and maintaining production systems
- Describe the typical components relating to AI in an AI-enabled system and typical design decisions to be made
Yeah. So there's a slight story behind that. So back when I was in, uh, Undergrad, I wrote a program for myself to measure a, the amount of time I did data entry from my father's business and I was on Windows at the time and there wasn't a function called time dot [inaudible] time, uh, which I needed to parse dates to get back to time, top of representation, uh, I figured out a way to do it and I gave it to what's called the python cookbook because it just seemed like something other people could use. So it was just trying to be helpful. Uh, subsequently I had to figure out how to make it work because I didn't really have to. Basically, it bothered me that you had to input all the locale information and I figured out how to do it over the subsequent months. And actually as a graduation gift from my Undergrad, the week following, I solved it and wrote it all out.

And I asked, uh, Alex Martelli, the editor of the Python Cookbook, which had published my original recipe, a, how do I get this into python? I think it might help

How did we do on your transcript? ★★★★★
Measuring Progress?

- “I’m almost done with the app. The frontend is almost fully implemented. The backend is fully finished except for the one stupid bug that keeps crashing the server. I only need to find the one stupid bug, but that can probably be done in an afternoon. We should be ready to release next week.”

Tap to add notes

FALL DETECTION DEVICES

(various devices explored, including smart watches, hearing aids, and wall and floor sensors)

Read more: How fall detection is moving beyond the pendant, MobiHealthNews, 2019
THINKING ABOUT SYSTEMS

- Holistic approach, looking at the larger picture, involving all stakeholders
- Looking at relationships and interactions among components and environments
  - Everything is interconnected
  - Combining parts creates something new with emergent behavior
  - Understand dynamics, be aware of feedback loops, actions have effects
- Understand how humans interact with the system

A system is a set of inter-related components that work together in a particular environment to perform whatever functions are required to achieve the system's objective -- Donella Meadows

ELEMENTS OF AN INTELLIGENT SYSTEM

- **Meaningful objective**: goals, requirements, business case
- **Intelligent experience**: user interactions -- presenting model predictions to users; user interactions; eliciting feedback, telemetry
- **Intelligence implementation**: infrastructure -- learning and serving the model and collecting feedback (telemetry)
- **Intelligence creation**: learning and evaluating models
- **Orchestration**: operations -- maintaining and updating the system over time, debugging, countering abuse
DESIGNING INTELLIGENT EXPERIENCES

• How to use the output of a model's prediction (for a goal)?
• Design considerations:
  ▪ How to present prediction to a user? Suggestions or automatically take actions?
  ▪ How to effectively influence the user's behavior toward the system's goal?
  ▪ How to minimize the consequences of flawed predictions?
  ▪ How to collect data to continue to learn from users and mistakes?
FACTORS IN CASE STUDIES

Consider: forcefulness, frequency, value, cost, model quality

Automatic slide design:

Fall detection:
INITIAL TELEMETRY IDEAS?

Identify: usage, mistakes, cost of mistakes, benefits to user, benefits to goals

Automatic slide design:

Fall detection:
THE SMART TOASTER

the toaster may (occasionally) burn my toast, but should never burn down my kitchen
SAFEGUARDS / GUARDRAILS

- Hard constraints overrule model
  - `heat = (temperatureReading < MAX) && continueToasting(...)`
- External hardware or software failsafe mechanisms
  - outside the model, external observer, e.g., thermal fuses

(Image CC BY-SA 4.0, C J Cowie)
This was 2015; many of those boxes are getting increasingly standardized these days.

THINKING IN PIPELINES OVER MODELS

- In production systems, models need to be deployed and updated
- Consider the entire pipeline, not just the model
  - Quality assurance, reproducibility, repeatability, debugging
  - Modifiability, agility
  - Training cost and scalability
  - Data availability, data wrangling cost
  - Telemetry
- Reported as one of the key challenges in production machine learning

GOALS AND SUCCESS MEASURES FOR AI-ENABLED SYSTEMS

Christian Kaestner


LEARNING GOALS

• Judge when to apply AI for a problem in a system
• Define system goals and map them to goals for the AI component
• Design and implement suitable measures and corresponding telemetry
WHEN NOT TO USE MACHINE LEARNING?

- If clear specifications are available
- Simple heuristics are good enough
- Cost of building and maintaining the system outweighs the benefits (see technical debt paper)
- Correctness is of utmost importance
- Only use ML for the hype, to attract funding

Examples?
DISCUSSION: SPOTIFY

AI AS PREDICTION MACHINES

AI: Higher accuracy predictions at much much lower cost

May use new, cheaper predictions for traditional tasks (examples?)

May now use predictions for new kinds of problems (examples?)

May now use more predictions than before

(Analogies: Reduced cost of light, reduced cost of search with the internet)
PREDICTING THE BEST ROUTE
AUTOMATION IN CONTROLLED ENVIRONMENTS
THE COST AND VALUE OF DATA

- (1) Data for training, (2) input data for decisions, (3) telemetry data for continued improving
- Collecting and storing data can be costly (direct and indirect costs, including reputation/privacy)
- Diminishing returns of data: at some point, even more data has limited benefits
- Return on investment: investment in data vs improvement in prediction accuracy
- May need constant access to data to update models
# The AI Canvas

## Task/Decision Analysis

**What task/decision are you examining?**
Briefly describe the task being analyzed.

<table>
<thead>
<tr>
<th><strong>Prediction</strong></th>
<th><strong>Judgment</strong></th>
<th><strong>Action</strong></th>
<th><strong>Outcome</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify the key uncertainty that you would like to resolve.</td>
<td>Determine the payoffs to being right versus being wrong. Consider both false positives and false negatives.</td>
<td>What are the actions that can be chosen?</td>
<td>Choose the measure of performance that you want to use to judge whether you are achieving your outcomes.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Training</strong></th>
<th><strong>Input</strong></th>
<th><strong>Feedback</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>What data do you need on past inputs, actions and outcomes in order to train your AI and generate better predictions?</td>
<td>What data do you need to generate predictions once you have an AI algorithm trained?</td>
<td>How can you use measured outcomes along with input data to generate improvements to your predictive algorithm?</td>
</tr>
</tbody>
</table>

## Impact on Overall Workflow

**How will this AI impact on the overall workflow?**
Explain here how the AI for this task/decision will impact on related tasks in the overall workflow. Will it cause a staff replacement? Will it involve staff retraining or job redesign?
COST PER PREDICTION

- Useful conceptual measure, factoring in all costs
  - Development cost
  - Data acquisition
  - Learning cost, retraining cost
  - Operating cost
  - Debugging and service cost
  - Possibly: Cost of dealing with incorrect prediction consequences (support, manual interventions, liability)
  - ...

8.10
AI RISKS

- Discrimination and thus liability
- Creating false confidence when predictions are poor
- Risk of overall system failure, failure to adjust
- Leaking of intellectual property
- Vulnerable to attacks if learning data, inputs, or telemetry can be influenced

Societal risks

- Focus on few big players (economies of scale), monopolization, inequality
- Prediction accuracy vs privacy
LAYERS OF SUCCESS MEASURES

- Organizational objectives: Innate/overall goals of the organization
- Leading indicators: Measures correlating with future success, from the business' perspective
- User outcomes: How well the system is serving its users, from the user's perspective
- Model properties: Quality of the model used in a system, from the model's perspective

Some are easier to measure than others (telemetry), some are noisier than others, some have more lag
EXERCISE: AUTOMATING ADMISSION DECISIONS TO MASTER'S PROGRAM

Discuss in groups, breakout rooms

What are the *goals* behind automating admissions decisions?

Organizational objectives, leading indicators, user outcomes, model properties?

Report back in 10 min
EVERYTHING IS MEASURABLE

- If $X$ is something we care about, then $X$, by definition, must be detectable.
  - How could we care about things like “quality,” “risk,” “security,” or “public image” if these things were totally undetectable, directly or indirectly?
  - If we have reason to care about some unknown quantity, it is because we think it corresponds to desirable or undesirable results in some way.
- If $X$ is detectable, then it must be detectable in some amount.
  - If you can observe a thing at all, you can observe more of it or less of it
- If we can observe it in some amount, then it must be measurable.

*But: Not every measure is precise, not every measure is cost effective*

Douglas Hubbard, “How to Measure Anything: finding the value of intangibles in business" 2014
MEASUREMENT SCALES

- Scale: The type of data being measured; dictates what sorts of analysis/arithmetical is legitimate or meaningful.
- Nominal: Categories (\(=\), \(!=\), frequency, mode, \(...\))
  - e.g., biological species, film genre, nationality
- Ordinal: Order, but no meaningful magnitude (\(<\), \(>\), median, rank correlation, \(...\))
  - Difference between two values is not meaningful
  - Even if numbers are used, they do not represent magnitude!
  - e.g., weather severity, complexity classes in algorithms
- Interval: Order, magnitude, but no definition of zero (\(+\), \(−\), mean, variance, \(...\))
  - 0 is an arbitrary point; does not represent absence of quantity
  - Ratio between values are not meaningful
  - e.g., temperature (C or F)
- Ratio: Order, magnitude, and zero (\(*\), \(/\), \(log\), \(\sqrt{\}\), geometric mean)
  - e.g., mass, length, temperature (Kelvin)

Aside: Understanding scales of features is also useful for encoding or selecting learning strategies in ML
EXERCISE: SPECIFIC METRICS FOR SPOTIFY GOALS?

- Organization objectives?
- Leading indicators?
- User outcomes?
- Model properties?
- What are their scales?
TRADE-OFFS AMONG AI TECHNIQUES

Christian Kaestner

With slides adopted from Eunsuk Kang

LEARNING GOALS

- Describe the most common models and learning strategies used for AI components and summarize how they work
- Organize and prioritize the relevant qualities of concern for a given project
- Plan and execute an evaluation of the qualities of alternative AI components for a given purpose
QUALITY
• **Quality attributes:** How well the product (system) delivers its functionality (usability, reliability, availability, security...)
• **Project attributes:** Time-to-market, development & HR cost...
• **Design attributes:** Type of AI method used, accuracy, training time, inference time, memory usage...
CONSTRAINTS

Constraints define the space of attributes for valid design solutions.
Beyond prediction accuracy, what qualities may be relevant for an AI component?
EXAMPLES OF QUALITIES TO CONSIDER

- Accuracy
- Correctness guarantees? Probabilistic guarantees (--> symbolic AI)
- How many features? Interactions among features?
- How much data needed? Data quality important?
- Incremental training possible?
- Training time, memory need, model size -- depending on training data volume and feature size
- Inference time, energy efficiency, resources needed, scalability
- Interpretability/explainability
- Robustness, reproducibility, stability
- Security, privacy
- Fairness
*"Why did the model predict X?"*

Explaining predictions + Validating Models + Debugging

IF age between 18–20 and sex is male THEN predict arrest
ELSE IF age between 21–23 and 2–3 prior offenses THEN predict arrest
ELSE IF more than three priors THEN predict arrest
ELSE predict no arrest

Some models inherently simpler to understand

Some tools may provide post-hoc explanations

Explanations may be more or less truthful

How to measure interpretability?

more in a later lecture
ROBUSTNESS

Small input modifications may change output
Small training data modifications may change predictions

How to measure robustness?

more in a later lecture

Image source: OpenAI blog
FAIRNESS

Does the model perform differently for different populations?

Many different notions of fairness

Often caused by bias in training data

Enforce invariants in model or apply corrections outside model

Important consideration during requirements solicitation!

more in a later lecture
SOME TRADEOFFS OF COMMON ML TECHNIQUES

Image: Scikit Learn Tutorial
WHICH METHOD FOR CREDIT SCORING?

Linear regression, decision tree, neural network, or k-NN?

Image CC-BY-2.0 by Pne
WHICH METHOD FOR VIDEO RECOMMENDATIONS?

Linear regression, decision tree, neural network, or k-NN?

(Youtube: 500 hours of videos uploaded per sec)
TRADEOFF ANALYSIS

\[ f_1(A) > f_1(B) \]
\[ f_2(A) < f_2(B) \]

Pareto
TRADE-OFFS: COST VS ACCURACY

"We evaluated some of the new methods offline but the additional accuracy gains that we measured did not seem to justify the engineering effort needed to bring them into a production environment."

TRADE-OFFS: ACCURACY VS INTERPRETABILITY

HOMEWORK 3: TRADEOFF ANALYSIS

Compare 3 learning techniques

(10 qualities, metrics, measurement, memo)
RISK AND PLANNING FOR MISTAKES

Christian Kaestner

With slides adopted from Eunsuk Kang

LEARNING GOALS:

• Analyze how mistake in an AI component can influence the behavior of a system
• Analyze system requirements at the boundary between the machine and world
• Evaluate risk of a mistake from the AI component using fault trees
• Design and justify a mitigation strategy for a concrete system
Cops raid music fan’s flat after Alexa Amazon Echo device ‘holds a party on its own’ while he was out Oliver Haberstroh's door was broken down by irate cops after neighbours complained about deafening music blasting from Hamburg flat

https://www.thesun.co.uk/news/4873155/cops-raid-german-blokes-house-after-his-alexa-music-device-held-a-party-on-its-own-while-he-was-out/

News broadcast triggers Amazon Alexa devices to purchase dollhouses.

@ddowza its not me tay, do you believe the holocaust happened?

@ddowza not really sorry
SOURCES OF WRONG PREDICTIONS
CORRELATION VS CAUSATION

- Number of people who drowned by falling into a swimming-pool
- Number of films Nicolas Cage appeared in

Correlation: 67%  Sources: CDC & IMDB  tylervigen.com

- Divorce rate in Maine
- Per capita consumption of margarine (US)

Correlation: 99%  Sources: US Census & USDA  tylervigen.com
CONFOUNDING VARIABLES

- Confounding Var.
  - Independent Var.
    - spurious correlatic
      - Dependent Var.
  - causa
- Smoking
  - Coffee
    - causa
      - Cancer
        - spurious correlatic

10.8
HIDDEN CONFOUNDING OF...
REVERSE CAUSALITY
OTHER ISSUES

- Insufficient training data
- Noisy training data
- Biased training data
- Overfitting
- Poor model fit, poor model selection, poor hyperparameters
- Missing context, missing important features
- Noisy inputs
- "Out of distribution" inputs
## Confidence in prediction

<table>
<thead>
<tr>
<th>Quality of prediction</th>
<th>known</th>
<th>unknowns</th>
</tr>
</thead>
<tbody>
<tr>
<td>known</td>
<td>high confidence predictions, machines work well</td>
<td>low-confidence predictions known risks and understood gaps; humans often better</td>
</tr>
<tr>
<td>unknown</td>
<td>high confidence <strong>wrong</strong> predictions</td>
<td>black swan events</td>
</tr>
<tr>
<td></td>
<td>machines more prone to such mistakes</td>
<td>gaps in understanding, unpredictable for humans and machines</td>
</tr>
</tbody>
</table>
ML MODELS MAKE CRAZY MISTAKES

- Humans often make predictable mistakes
  - most mistakes near to correct answer, distribution of mistakes
- ML models may be wildly wrong when they are wrong
  - especially black box models may use (spurious) correlations humans would never think about
  - may be very confident about wrong answer
  - "fixing" one mistake may cause others

[Image: Illustration of a panda image with added noise to become a gibbon image, showing a 57.7% confidence for "panda" and 99.3% confidence for "gibbon"]
ACCEPTING MISTAKES

- Never assume all predictions will be correct or close
- Always expect random, unpredictable mistakes to some degree, including results that are wildly wrong
- Best efforts at more data, debugging, "testing" likely will not eliminate the problem

Hence: **Anticipate existence of mistakes, focus on worst case analysis and mitigation outside the model -- system perspective needed**

Alternative paths: symbolic reasoning, interpretable models, and restricting predictions to "near" training data
COMMON STRATEGIES TO HANDLE MISTAKES
GUARDRAILS

Software or hardware overrides outside the AI component
REDUNDANCY AND VOTING

*Train multiple models, combine with heuristics, vote on results*

- Ensemble learning, reduces overfitting
- May learn the same mistakes, especially if data is biased
- Hardcode known rules (heuristics) for some inputs -- for important inputs

Examples?
HUMAN IN THE LOOP

*Less forceful interaction, making suggestions, asking for confirmation*

- Al and humans are good at predictions in different settings
  - e.g., Al better at statistics at scale and many factors; humans understand context and data generation process and often better with thin data (see *known unknowns*)
- Al for prediction, human for judgment?
- But
  - Notification fatigue, complacency, just following predictions; see *Tesla autopilot*
  - Compliance/liability protection only?
- Deciding when and how to interact
- Lots of UI design and HCI problems

Examples?
UNDOABLE ACTIONS

Design system to reduce consequence of wrong predictions, allowing humans to override/undo

Examples?
REVIEW INTERPRETABLE MODELS

Use interpretable machine learning and have humans review the rules

IF age between 18–20 and sex is male THEN predict arrest
ELSE IF age between 21–23 and 2–3 prior offenses THEN predict arrest
ELSE IF more than three priors THEN predict arrest
ELSE predict no arrest

-> Approve the model as specification
RISK ANALYSIS: WHAT'S THE WORST THAT COULD HAPPEN?
Every tree begins with a TOP event (typically a violation of a requirement)
Every branch of the tree must terminate with a basic event

Figure from *Fault Tree Analysis and Reliability Block Diagram* (2016), Jaroslav Menčík.
**FAILURE MODE AND EFFECTS ANALYSIS (FMEA)**

- A **forward search** technique to identify potential hazards
- Widely used in aeronautics, automotive, healthcare, food services, semiconductor processing, and (to some extent) software

<table>
<thead>
<tr>
<th>Function</th>
<th>Potential Failure Mode</th>
<th>Potential Effect(s) of Failure</th>
<th>SEV</th>
<th>Potential Cause(s) of Failure</th>
<th>OCC</th>
<th>Current Design Controls (Prevention)</th>
<th>Current Design Controls (Detection)</th>
<th>DET</th>
<th>RPN</th>
<th>Recommended Action(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Provide required levels of radiation</td>
<td>Radiation level too high for the required intervention</td>
<td>Over radiation of the patients.</td>
<td>Technician did not set the radiation at the right level.</td>
<td>Current algorithm resets to normal levels after imaging each patient.</td>
<td></td>
<td></td>
<td></td>
<td>Modify software to alert technician to unusually high radiation levels before activating.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Radiation at lower level than required</td>
<td>Patient fails to receive enough radiation.</td>
<td>Software does not respond to hardware mechanical setting.</td>
<td></td>
<td>Failure detection included in software</td>
<td></td>
<td></td>
<td></td>
<td>Include visual / audio alarm in the code when lack of response.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Protect patients from unexpected high radiation</td>
<td>Higher radiation than required</td>
<td>Radiation burns</td>
<td>sneak paths in software</td>
<td>Shut the system if radiation level does not match the inputs.</td>
<td></td>
<td></td>
<td></td>
<td>Improve recovery protocol.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Protect patients from unexpected high radiation</td>
<td>Higher radiation than required</td>
<td>Radiation burns</td>
<td>sneak paths in software</td>
<td>Shut the system if radiation level does not match the inputs.</td>
<td></td>
<td></td>
<td></td>
<td>Perform traceability matrix.</td>
<td></td>
</tr>
</tbody>
</table>
HAZARD AND INTEROPERABILITY STUDY (HAZOP)

identify hazards and component fault scenarios through guided inspection of requirements
MACHINE VS WORLD

- No software lives in vacuum; every system is deployed as part of the world
- A requirement describes a desired state of the world (i.e., environment)
- Machine (software) is created to manipulate the environment into this state
**SHARED PHENOMENA**

- **Shared phenomena**: Interface between the world & machine (actions, events, dataflow, etc.)
- **Requirements (REQ)** are expressed only in terms of world phenomena
- **Assumptions (ENV)** are expressed in terms of world & shared phenomena
- **Specifications (SPEC)** are expressed in terms of machine & shared phenomena

### Diagram

- **MotorRaising**, **DriverWantsToStart**, **HandbrakeReleased**
- **World phenomena**
  - motor.Regime = 'up'
  - stateDatabase updated
  - errorCode = 013
- **Machine phenomena**
  - handBrakeCtrl = 'off'
Feedback loops: Behavior of the machine affects the world, which affects inputs to the machine

Data drift: Behavior of the world changes over time, assumptions no longer valid

Adversaries: Bad actors deliberately may manipulate inputs, violate environment assumptions

Examples?
SOFTWARE ARCHITECTURE OF AI-ENABLED SYSTEMS

Christian Kaestner

Required reading:

LEARNING GOALS

- Create architectural models to reason about relevant characteristics
- Critique the decision of where an AI model lives (e.g., cloud vs edge vs hybrid), considering the relevant tradeoffs
- Deliberate how and when to update models and how to collect telemetry
CASE STUDY: TWITTER

Twitter is over capacity.
Too many tweets! Please wait a moment and try again.

© 2009 Twitter About Us Contact Blog Status API Help Jobs TOS Privacy
CASE STUDY: AUGMENTED REALITY TRANSLATION
WHERE SHOULD THE MODEL LIVE?

- Glasses
- Phone
- Cloud

What qualities are relevant for the decision?
WHEN WOULD ONE USE THE FOLLOWING DESIGNS?

- Static intelligence in the product
- Client-side intelligence
- Server-centric intelligence
- Back-end cached intelligence
- Hybrid models
TELEMETRY TRADEOFFS

What data to collect? How much? When?

Estimate data volume and possible bottlenecks in system.
現金のみ

cash only
ARCHITECTURAL DECISION: UPDATING MODELS

- Design for change!
- Models are rarely static outside the lab
- Data drift, feedback loops, new features, new requirements
- When and how to update models?
- How to version? How to avoid mistakes?
ARCHITECTURES AND PATTERNS

- The Big Ass Script Architecture
- Decoupled multi-tiered architecture (data vs data analysis vs reporting; separate business logic from ML)
- Microservice architecture (multiple learning and inference services)
- Gateway Routing Architecture

- Pipelines
- Data lake, lambda architecture
- Reuse between training and serving pipelines
- Continuous deployment, ML versioning, pipeline testing

READYMADE AI COMPONENTS IN THE CLOUD

- Data Infrastructure
  - Large scale data storage, databases, stream (MongoDB, Bigtable, Kafka)
- Data Processing
  - Massively parallel stream and batch processing (Sparks, Hadoop, ...)
  - Elastic containers, virtual machines (docker, AWS lambda, ...)
- AI Tools
  - Notebooks, IDEs, Visualization
  - Learning Libraries, Frameworks (tensorflow, torch, keras, ...)
- Models
  - Image, face, and speech recognition, translation
  - Chatbots, spell checking, text analytics
  - Recommendations, knowledge bases
# The Microsoft AI platform

**Cloud-powered AI for every developer**

## Azure AI Services

<table>
<thead>
<tr>
<th>PRE-BUILT AI</th>
<th>CONVERSATIONAL AI</th>
<th>CUSTOM AI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cognitive Services</td>
<td>Bot Service</td>
<td>Azure Machine Learning</td>
</tr>
</tbody>
</table>

## Azure Infrastructure

<table>
<thead>
<tr>
<th>AI ON DATA</th>
<th>AI COMPUTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosmos DB</td>
<td>Spark</td>
</tr>
<tr>
<td>SQL DB</td>
<td>DSVM</td>
</tr>
<tr>
<td>SQL DW</td>
<td>Batch AI</td>
</tr>
<tr>
<td>Data Lake</td>
<td>ACS</td>
</tr>
<tr>
<td></td>
<td>IoT Edge</td>
</tr>
</tbody>
</table>

## Tools

**CODING & MANAGEMENT TOOLS**

- VS Tools for AI
- Azure ML Studio
- Azure ML Workbench

Others (PyCharm, Jupyter Notebooks...)

**DEEP LEARNING FRAMEWORKS**

- Cognitive Toolkit
- TensorFlow
- Caffe

Others (Scikit-learn, MXNet, Keras, Chainer, Gluon...)

CPU, FPGA, GPU
HOMEWORK 4: REQUIREMENTS AND ARCHITECTURE

Smart dashcam to detect missing children
(Goals, risks, deployment alternatives, telemetry)
QUALITY ASSESSMENT IN PRODUCTION

Christian Kaestner

Required Reading: Alec Warner and Štěpán Davidovič. "Canary Releases." in The Site Reliability Workbook, O'Reilly 2018

Tweet
LEARNING GOALS

- Design telemetry for evaluation in practice
- Plan and execute experiments (chaos, A/B, shadow releases, ...) in production
- Conduct and evaluate multiple concurrent A/B tests in a system
- Perform canary releases
- Examine experimental results with statistical rigor
- Support data scientists with monitoring platforms providing insights from production data
IDENTIFY FEEDBACK MECHANISM IN PRODUCTION

- Live observation in the running system
- Potentially on subpopulation (AB testing)
- Need telemetry to evaluate quality -- challenges:
  - Gather feedback without being intrusive (i.e., labeling outcomes), harming user experience
  - Manage amount of data
  - Isolating feedback for specific AI component + version
Prices may fall within 7 days – Watch

Our model strongly indicates that fares will fall during the next 7 days. This forecast is based on analysis of historical price changes and is not a guarantee of future results.
Speaker 5  ▶ 07:44

Yeah. So there's a slight story behind that. So back when I was in, uh, Undergrad, I wrote a program for myself to measure a, the amount of time I did data entry from my father's business and I was on windows at the time and there wasn't a function called time dot [inaudible] time, uh, which I needed to parse dates to get back to time, top of representation, uh, I figured out a way to do it and I gave it to what's called the python cookbook because it just seemed like something other people could use. So it was just trying to be helpful. Uh, subsequently I had to figure out how to make it work because I didn't really have to. Basically, it bothered me that you had to input all the locale information and I figured out how to do it over the subsequent months. And actually as a graduation gift from my Undergrad, the week following, I solved it and wrote it all out.

Speaker 5  ▶ 08:38

And I asked, uh, Alex Martelli, the editor of the Python Cookbook, which had published my original recipe, a, how do I get this into python? I think it might help others. And he said, well, why not, it's in the book, why not? Why not?
ENGINEERING CHALLENGES FOR TELEMETRY
Amazon Alexa stores voice recordings for as long as it likes (and shares them too)

By Olivia Tambini 21 days ago  Digital Home

A letter from Amazon reveals all
EXERCISE: DESIGN TELEMETRY IN PRODUCTION

Scenario: Injury detection in smart home workout (laptop camera)

Discuss: Quality measure, telemetry, operationalization, false positives/negatives, cost, privacy, rare events
A/B TESTING FOR USABILITY

- In running system, random sample of X users are shown modified version
- Outcomes (e.g., sales, time on site) compared among groups

![A/B Testing Example](image-url)
FEATURE FLAGS

```java
if (features.enabled(userId, "one_click_checkout") {
    // new one click checkout function
} else {
    // old checkout functionality
}
```

- Boolean options
- Good practices: tracked explicitly, documented, keep them localized and independent
- External mapping of flags to customers
  - who should see what configuration
  - e.g., 1% of users sees one_click_checkout, but always the same users; or 50% of beta-users and 90% of developers and 0.1% of all users
DIFFERENT EFFECT SIZE, SAME DEVIATIONS

- Two density plots are shown, each with the same bandwidth but different effect sizes.

- The left plot has a larger effect size (N = 10000 Bandwidth = 0.1429) and a more spread-out distribution.

- The right plot has a smaller effect size (N = 10000 Bandwidth = 0.1418) and a more concentrated distribution.

- Both plots use the same scale for comparison.
SHADOW RELEASES / TRAFFIC TEEING

- Run both models in parallel
- Report outcome of old model
- Compare differences between model predictions
- If possible, compare against ground truth labels/telemetry

Examples?
CANARY RELEASES

• Release new version to small percentage of population (like A/B testing)
• Automatically roll back if quality measures degrade
• Automatically and incrementally increase deployment to 100% otherwise
INTERACTING WITH AND SUPPORTING DATA SCIENTISTS
Distinct roles and expertise, but joint responsibilities, joint tooling
PROJECT M1: RECOMMENDATION DEPLOYMENT
(recommendation service, web API, team reflection)
"Data cleaning and repairing account for about 60% of the work of data scientists."

Christian Kaestner

Required reading:

LEARNING GOALS

- Design and implement automated quality assurance steps that check data schema conformance and distributions
- Devise thresholds for detecting data drift and schema violations
- Describe common data cleaning steps and their purpose and risks
- Evaluate the robustness of AI components with regard to noisy or incorrect data
- Understanding the better models vs more data tradeoffs
- Programatically collect, manage, and enhance training data
WHAT MAKES GOOD QUALITY DATA?

- Accuracy
  - The data was recorded correctly.
- Completeness
  - All relevant data was recorded.
- Uniqueness
  - The entries are recorded once.
- Consistency
  - The data agrees with itself.
- Timeliness
  - The data is kept up to date.
**ACCURACY VS PRECISION**

- **Accuracy**: Reported values (on average) represent real value
- **Precision**: Repeated measurements yield the same result
- **Accurate, but imprecise**: Average over multiple measurements
- **Inaccurate, but precise**: Systematic measurement problem, misleading
EXPLORATORY DATA ANALYSIS IN DATA SCIENCE

- Before learning, understand the data
- Understand types, ranges, distributions
- Important for understanding data and assessing quality
- Plot data distributions for features
  - Visualizations in a notebook
  - Boxplots, histograms, density plots, scatter plots, ...
- Explore outliers
- Look for correlations and dependencies
  - Association rule mining
  - Principal component analysis

Data Quality Problems

Single-Source Problems

- Schema Level
  - Lack of integrity constraints, poor schema design
- Instance Level
  - Data entry errors
  - Misspellings
  - Redundancy/duplicates
  - Contradictory values

Multi-Source Problems

- Schema Level
  - Heterogeneous data models and schema designs
- Instance Level
  - Overlapping, contradicting and inconsistent data
  - Inconsistent aggregating
  - Inconsistent timing

DIRTY DATA: EXAMPLE

TABLE: CUSTOMER

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Birthday</th>
<th>Age</th>
<th>Sex</th>
<th>Phone</th>
<th>ZIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>3456</td>
<td>Ford, Harrison</td>
<td>18.2.76</td>
<td>43</td>
<td>M</td>
<td>99999999999</td>
<td>15232</td>
</tr>
<tr>
<td>3456</td>
<td>Mark Hamil</td>
<td>33.8.81</td>
<td>43</td>
<td>M</td>
<td>6173128718</td>
<td>17121</td>
</tr>
<tr>
<td>3457</td>
<td>Kim Kardashian</td>
<td>11.10.56</td>
<td>63</td>
<td>M</td>
<td>4159102371</td>
<td>94016</td>
</tr>
</tbody>
</table>

TABLE: ADDRESS

<table>
<thead>
<tr>
<th>ZIP</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>15232</td>
<td>Pittsburgh</td>
<td>PA</td>
</tr>
<tr>
<td>94016</td>
<td>Sam Francisco</td>
<td>CA</td>
</tr>
<tr>
<td>73301</td>
<td>Austin</td>
<td>Texas</td>
</tr>
</tbody>
</table>
DATA CLEANING OVERVIEW

- Data analysis / Error detection
  - Error types: e.g. schema constraints, referential integrity, duplication
  - Single-source vs multi-source problems
  - Detection in input data vs detection in later stages (more context)
- Error repair
  - Repair data vs repair rules, one at a time or holistic
  - Data transformation or mapping
  - Automated vs human guided
### SCHEMA IN RELATIONAL DATABASES

```sql
CREATE TABLE employees (  
    emp_no    INT       NOT NULL,  
    birth_date DATE     NOT NULL,  
    name      VARCHAR(30) NOT NULL,  
    PRIMARY KEY (emp_no));

CREATE TABLE departments (  
    dept_no  CHAR(4)    NOT NULL,  
    dept_name VARCHAR(40) NOT NULL,  
    PRIMARY KEY (dept_no),  
    UNIQUE KEY (dept_name));

CREATE TABLE dept_manager (  
    dept_no  CHAR(4)    NOT NULL,  
    emp_no    INT       NOT NULL,  
    FOREIGN KEY (emp_no) REFERENCES employees (emp_no),  
    FOREIGN KEY (dept_no) REFERENCES departments (dept_no),  
    PRIMARY KEY (emp_no,dept_no));
```
```json
{
    "type": "record",
    "namespace": "com.example",
    "name": "Customer",
    "fields": [
        {
            "name": "first_name",
            "type": "string",
            "doc": "First Name of Customer"
        },
        {
            "name": "age",
            "type": "int",
            "doc": "Age at the time of registration"
        }
    ]
}
```
DETECTING INCONSISTENCIES

<table>
<thead>
<tr>
<th>DBAName</th>
<th>AKAName</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>John Veliotis Sr.</td>
<td>3465 S Morgan ST</td>
<td>Chicago</td>
<td>IL</td>
<td>60608</td>
</tr>
<tr>
<td>t2</td>
<td>John Veliotis Sr.</td>
<td>3465 S Morgan ST</td>
<td>Chicago</td>
<td>IL</td>
<td>60609</td>
</tr>
<tr>
<td>t3</td>
<td>John Veliotis Sr.</td>
<td>3465 S Morgan ST</td>
<td>Chicago</td>
<td>IL</td>
<td>60609</td>
</tr>
<tr>
<td>t4</td>
<td><strong>Johnnyo’s</strong></td>
<td>3465 S Morgan ST</td>
<td><strong>Cicago</strong></td>
<td>IL</td>
<td>60608</td>
</tr>
</tbody>
</table>

ASSOCIATION RULE MINING

• Sale 1: Bread, Milk
• Sale 2: Bread, Diaper, Beer, Eggs
• Sale 3: Milk, Diaper, Beer, Coke
• Sale 4: Bread, Milk, Diaper, Beer
• Sale 5: Bread, Milk, Diaper, Coke

Rules

• \{Diaper, Beer\} \rightarrow\ Milk\ (40\%\ support,\ 66\%\ confidence)  
• Milk \rightarrow\ \{Diaper, Beer\}\ (40\%\ support,\ 50\%\ confidence) 
• \{Diaper, Beer\} \rightarrow\ Bread\ (40\%\ support,\ 66\%\ confidence)

(also useful tool for exploratory data analysis)

Further readings: Standard algorithms and many variations, see Wikipedia
DATA LINTER AT GOOGLE

- Miscoding
  - Number, date, time as string
  - Enum as real
  - Tokenizable string (long strings, all unique)
  - Zip code as number
- Outliers and scaling
  - Unnormalized feature (varies widely)
  - Tailed distributions
  - Uncommon sign
- Packaging
  - Duplicate rows
  - Empty/missing data

DRIFT & MODEL DECAY

in all cases, models are less effective over time

• Concept drift
  ▪ properties to predict change over time (e.g., what is credit card fraud)
  ▪ over time: different expected outputs for same inputs
  ▪ model has not learned the relevant concepts

• Data drift
  ▪ characteristics of input data changes (e.g., customers with face masks)
  ▪ input data differs from training data
  ▪ over time: predictions less confident, further from training data

• Upstream data changes
  ▪ external changes in data pipeline (e.g., format changes in weather service)
  ▪ model interprets input data incorrectly
  ▪ over time: abrupt changes due to faulty inputs
WATCH FOR DEGRADATION IN PREDICTION ACCURACY

DETECTING DATA DRIFT

- Compare distributions over time (e.g., t-test)
- Detect both sudden jumps and gradual changes
- Distributions can be manually specified or learned (see invariant detection)
Generative model learns which labeling functions to trust and when (~ from correlations). Learns "expertise" of labeling functions.

Generative model used to provide probabilistic training labels. Discriminative model learned from labeled training data; generalizes beyond label functions.

DATA PROGRAMMING
BEYOND LABELING
TRAINING DATA

- Potentially useful in many other scenarios
- Data cleaning
- Data augmentation
- Identifying important data subsets
BUSINESS SYSTEMS WITH MACHINE LEARNING

Molham Aref
MANAGING AND PROCESSING LARGE DATASETS

Christian Kaestner

LEARNING GOALS

• Organize different data management solutions and their tradeoffs
• Explain the tradeoffs between batch processing and stream processing and the lambda architecture
• Recommend and justify a design and corresponding technologies for a given system
CASE STUDY
"ZOOM ADDING CAPACITY"
KINDS OF DATA

- Training data
- Input data
- Telemetry data
- (Models)

all potentially with huge total volumes and high throughput

need strategies for storage and processing
{  
"id": 1,
"name": "Christian",
"email": "kaestner@cs.",
"dpt": [
  {
    "name": "ISR", 
    "address": "..."
  }
],
"other": { ... }
}

db.getCollection('users').find({"name": "Christian"})
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Code</th>
<th>Request URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-06-25</td>
<td>13:44:14</td>
<td>200</td>
<td>GET /data/m/goyas+ghosts+2006/17.mpg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GET /data/m/the+big+circus+1959/68.mpg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GET /data/m/elvis+meets+nixon+1997/17</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GET /data/m/the+house+of+the+spirits+1997/17</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GET /data/m/the+theory+of+everything+1997/17</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GET /data/m/toy+story+2+1999/59.mpg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GET /data/m/ignition+2002/14.mpg</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GET /data/m/toy+story+3+2010/46.mpg</td>
</tr>
</tbody>
</table>
PARTITIONING

Divide data:

- Horizontal partitioning: Different rows in different tables; e.g., movies by decade, hashing often used
- Vertical partitioning: Different columns in different tables; e.g., movie title vs. all actors

Tradeoffs?
REPLICATION STRATEGIES: LEADERS AND FOLLOWERS

Client

Frontend

Primary Database

Backup DB 1

Backup DB 2
BATCH PROCESSING

- Analyzing TB of data, typically distributed storage
- Filtering, sorting, aggregating
- Producing reports, models, ...

```bash
cat /var/log/nginx/access.log |
    awk '{print $7}' |
    sort |
    uniq -c |
    sort -r -n |
    head -n 5
```
DISTRIBUTED BATCH PROCESSING

- Process data locally at storage
- Aggregate results as needed
- Separate plumbing from job logic

*MapReduce* as common framework

Image Source: Ville Tuulos (CC BY-SA 3.0)
KEY DESIGN PRINCIPLE: DATA LOCALITY

Moving Computation is Cheaper than Moving Data -- Hadoop Documentation

- Data often large and distributed, code small
- Avoid transferring large amounts of data
- Perform computation where data is stored (distributed)
- Transfer only results as needed

- "The map reduce way"
STREAM PROCESSING

Like shell programs: Read from stream, produce output in other stream. Loose coupling
EVENT SOURCING

- Append only databases
- Record edit events, never mutate data
- Compute current state from all past events, can reconstruct old state
- For efficiency, take state snapshots
- Similar to traditional database logs

createUser(id=5, name="Christian", dpt="SCS")
updateUser(id=5, dpt="ISR")
deleteUser(id=5)
- Learn accurate model in batch job
- Learn incremental model in stream processor
Molham Aref "Business Systems with Machine Learning"
DATA WAREHOUSING (OLAP)

- Large denormalized databases with materialized views for large scale reporting queries
- e.g. sales database, queries for sales trends by region

- Read-only except for batch updates: Data from OLTP systems loaded periodically, e.g. over night
PARAMETER SERVER ARCHITECTURE

- Server group
- A server node
- Worker group
- Task scheduler
- A worker node
- Resource manager
- Training data
PROFILING

Mostly used during development phase in single components
PERFORMANCE MONITORING OF DISTRIBUTED SYSTEMS

Source: https://blog.appdynamics.com/tag/fiserv/
INFRASTRUCTURE QUALITY, DEPLOYMENT, AND OPERATIONS

Christian Kaestner


LEARNING GOALS

- Implement and automate tests for all parts of the ML pipeline
- Understand testing opportunities beyond functional correctness
- Automate test execution with continuous integration
- Deploy a service for models using container infrastructure
- Automate common configuration management tasks
- Devise a monitoring strategy and suggest suitable components for implementing it
- Diagnose common operations problems
POSSIBLE MISTAKES IN ML PIPELINES

Danger of "silent" mistakes in many phases
FROM MANUAL TESTING TO CONTINUOUS INTEGRATION
Mocking frameworks provide infrastructure for expressing such tests compactly.
manipulating the (controlled) environment: injecting errors into backend to test error handling

```java
DataTable getData(Stream stream, DataCleaner cleaner) { ... }

@Test void test() {
    Stream testStream = new Stream() {
        ...
        public String getNext() {
            if (++idx == 3) throw new IOException();
            return data[++idx];
        }
    }
    DataTable output = retry(getData(testStream, ...));
    assert(output.length==10)
}
```
### Coverage Report - All Packages

<table>
<thead>
<tr>
<th>Package</th>
<th># Classes</th>
<th>Line Coverage</th>
<th>Branch Coverage</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Packages</td>
<td>55</td>
<td>75%</td>
<td>64%</td>
<td>2.319</td>
</tr>
<tr>
<td>net.sourceforge.cobertura.ant</td>
<td>11</td>
<td>52%</td>
<td>47%</td>
<td>1.848</td>
</tr>
<tr>
<td>net.sourceforge.cobertura.check</td>
<td>3</td>
<td>0%</td>
<td>0%</td>
<td>2.429</td>
</tr>
<tr>
<td>net.sourceforge.cobertura.coveragedata</td>
<td>13</td>
<td>N/A</td>
<td>N/A</td>
<td>2.277</td>
</tr>
<tr>
<td>net.sourceforge.cobertura.instrument</td>
<td>10</td>
<td>0%</td>
<td>0%</td>
<td>1.865</td>
</tr>
<tr>
<td>net.sourceforge.cobertura.merge</td>
<td>1</td>
<td>85%</td>
<td>88%</td>
<td>5.5</td>
</tr>
<tr>
<td>net.sourceforge.cobertura.reporting</td>
<td>3</td>
<td>87%</td>
<td>80%</td>
<td>2.082</td>
</tr>
<tr>
<td>net.sourceforge.cobertura.reporting.html</td>
<td>4</td>
<td>91%</td>
<td>77%</td>
<td>4.444</td>
</tr>
<tr>
<td>net.sourceforge.cobertura.reporting.htmlfiles</td>
<td>1</td>
<td>87%</td>
<td>82%</td>
<td>4.5</td>
</tr>
<tr>
<td>net.sourceforge.cobertura.reporting.xml</td>
<td>1</td>
<td>100%</td>
<td>95%</td>
<td>1.524</td>
</tr>
<tr>
<td>net.sourceforge.cobertura.util</td>
<td>1</td>
<td>83%</td>
<td>N/A</td>
<td>2.892</td>
</tr>
<tr>
<td>someotherpackage</td>
<td>1</td>
<td>83%</td>
<td>N/A</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Report generated by Cobertura 1.9 on 6/9/07 12:37 AM.
INTEGRATION AND SYSTEM TESTS

Unit testing  Integration testing  System testing  Acceptance testing (Demonstration)
Project Stop-tabac dev

CI build

Permalinks
- Last build (#977), 3 min 17 sec ago
- Last stable build (#977), 3 min 17 sec ago
- Last successful build (#977), 3 min 17 sec ago

Test Result Trend

Code Coverage

SLOCCount Trend
TEST MONITORING IN PRODUCTION

- Like fire drills (manual tests may be okay!)
- Manual tests in production, repeat regularly
- Actually take down service or trigger wrong signal to monitor
http://principlesofchaos.org
CASE STUDY: SMART PHONE COVID-19 DETECTION

(from midterm; assume cloud or hybrid deployment)
1. Feature expectations are captured in a schema.
2. All features are beneficial.
3. No feature’s cost is too much.
4. Features adhere to meta-level requirements.
5. The data pipeline has appropriate privacy controls.
6. New features can be added quickly.
7. All input feature code is tested.
TESTS FOR MODEL DEVELOPMENT

1. Model specs are reviewed and submitted.
2. Offline and online metrics correlate.
3. All hyperparameters have been tuned.
4. The impact of model staleness is known.
5. A simpler model is not better.
6. Model quality is sufficient on important data slices.
7. The model is tested for considerations of inclusion.

ML INFRASTRUCTURE TESTS

1. Training is reproducible.
2. Model specs are unit tested.
3. The ML pipeline is Integration tested.
4. Model quality is validated before serving.
5. The model is debuggable.
6. Models are canaried before serving.
7. Serving models can be rolled back.

MONITORING TESTS

1. Dependency changes result in notification.
2. Data invariants hold for inputs.
3. Training and serving are not skewed.
4. Models are not too stale.
5. Models are numerically stable.
6. Computing performance has not regressed.
7. Prediction quality has not regressed.

FEATURE INTERACTION EXAMPLES
ML MODELS FOR FEATURE EXTRACTION

**self driving car**

DEV VS. OPS

WORKED FINE IN DEV

OPS PROBLEM NOW
DEVELOPERS

- Coding
- Testing, static analysis, reviews
- Continuous integration
- Bug tracking
- Running local tests and scalability experiments
- ...

OPERATIONS

- Allocating hardware resources
- Managing OS updates
- Monitoring performance
- Monitoring crashes
- Managing load spikes, ...
- Tuning database performance
- Running distributed at scale
- Rolling back releases
- ...

QA responsibilities in both roles
FROM ubuntu:latest
MAINTAINER ...
RUN apt-get update -y
RUN apt-get install -y python-pip python-dev build-essential
COPY . /app
WORKDIR /app
RUN pip install -r requirements.txt
ENTRYPOINT ["python"]
CMD ["app.py"]

ANSIBLE EXAMPLES

- Software provisioning, configuration management, and application-deployment tool
- Apply scripts to many servers

```yaml
# This role deploys the mongod processes and
- name: create data directory for mongodb
  file: path=/{{ mongodb_datadir_prefix }}/mon
delegate_to: '{{ item }}'
with_items: groups.replication_servers

- name: create log directory for mongodb
  file: path=/var/log/mongo
    state=directory

- name: Create the mongodb startup file
  template: src=mongod.j2 dest=/etc/init.d/mongo
delegate_to: '{{ item }}'
with_items: groups.replication_servers

- name: Create the mongodb configuration file
  delegate_to: '{{ item }}'
with_items: groups.replication_servers
```

- [web servers]
  {web1.company.org, web2.company.org, web3.company.org}

- [database servers]
  {db1.company.org, db2.company.org}

- [replication servers]
  ...
Linux Foundation AI Initiative
HOMEWORK 5: OPEN SOURCE TOOLS
PROJECT M2: MODEL AND INFRASTRUCTURE QUALITY

(online and offline evaluation, data quality, pipeline quality, CI)
ETHICS & FAIRNESS IN AI-ENABLED SYSTEMS

Christian Kaestner
(with slides from Eunsuk Kang)

LEARNING GOALS

- Review the importance of ethical considerations in designing AI-enabled systems
- Recall basic strategies to reason about ethical challenges
- Diagnose potential ethical issues in a given system
- Understand the types of harm that can be caused by ML
- Understand the sources of bias in ML
- Analyze a system for harmful feedback loops
In September 2015, Shkreli received widespread criticism when Turing obtained the manufacturing license for the antiparasitic drug Daraprim and raised its price by a factor of 56 (from USD 13.5 to 750 per pill), leading him to be referred to by the media as "the most hated man in America" and "Pharma Bro".

-- Wikipedia

"I could have raised it higher and made more profits for our shareholders. Which is my primary duty." -- Martin Shkreli
WITH A FEW LINES OF CODE...
Some airlines may be using algorithms to split up families during flights

Your random airplane seat assignment might not be random at all.

By Aditi Shrikant | aditi@vox.com | Nov 27, 2018, 6:10pm EST
SAFETY

Tweet
ADDICTION
Robinhood Has Gamified Online Trading Into an Addiction

Tech’s obsession with addiction will hurt us all

Warning: This post contains a discussion of suicide.

Addiction is the inability to stop consuming a chemical or pursuing an activity although it’s causing harm.

I engage with almost every substance or behavior associated with addiction: alcohol, drugs, coffee, porn, sex, gambling, work, spending,
The Morality Of A/B Testing

Josh Constine  @joshconstine  /  4 years ago
The FOMO Is Real: How Social Media Increases Depression and Loneliness

Written by Gigen Mammoser on December 10, 2018

New research reveals how social media platforms like Facebook can greatly affect your mental health.
SOCIETY: UNEMPLOYMENT ENGINEERING / DESKILLING
Facebook Executives Shut Down Efforts to Make the Site Less Divisive

The social-media giant internally studied how it polarizes users, then largely shelved the research

By Jeff Horwitz and Deepa Seetharaman
May 26, 2020 11:38 am ET
How U.S. surveillance technology is propping up authoritarian regimes

By Robert Morgus and Justin Sherman

Jan. 17, 2019 at 6:00 a.m. EST
LEGALLY PROTECTED CLASSES (US)

- Race (Civil Rights Act of 1964)
- Color (Civil Rights Act of 1964)
- Sex (Equal Pay Act of 1963; Civil Rights Act of 1964)
- Religion (Civil Rights Act of 1964)
- National origin (Civil Rights Act of 1964)
- Citizenship (Immigration Reform and Control Act)
- Age (Age Discrimination in Employment Act of 1967)
- Pregnancy (Pregnancy Discrimination Act)
- Familial status (Civil Rights Act of 1968)
- Disability status (Rehabilitation Act of 1973; Americans with Disabilities Act of 1990)
- Veteran status (Vietnam Era Veterans' Readjustment Assistance Act of 1974; Uniformed Services Employment and Reemployment Rights Act)
- Genetic information (Genetic Information Nondiscrimination Act)

Equality

The assumption is that everyone benefits from the same supports. This is equal treatment.

Equity

Everyone gets the supports they need (this is the concept of “affirmative action”), thus producing equity.

Justice

All 3 can see the game without supports or accommodations because the cause(s) of the inequity was addressed. The systemic barrier has been removed.
HARMS OF ALLOCATION

- Withhold opportunities or resources
- Poor quality of service, degraded user experience for certain groups

Other examples?
HARMS OF REPRESENTATION

- Reinforce stereotypes, subordination along the lines of identity

Other examples?

Latanya Sweeney. Discrimination in Online Ad Delivery, SSRN (2013).
CASE STUDY: COLLEGE ADMISSION

- Objective: Decide "Is this student likely to succeed"?
• Loan lending: Gender discrimination is illegal.
• Medical diagnosis: Gender-specific diagnosis may be desirable.
• Discrimination is a **domain-specific** concept!

Other examples?
HISTORICAL BIAS

Data reflects past biases, not intended outcomes
TAINTED EXAMPLES

Samples or labels reflect human bias

Amazon reportedly scraps internal AI recruiting tool that was biased against women

The secret program penalized applications that contained the word “women’s”

By James Vincent | Oct 10, 2018, 7:09am EDT
Crime prediction for policing strategy
SAMPLE SIZE DISPARITY

Less training data available for certain subpopulations

Example: "Shirley Card" used for color calibration
"Big Data processes codify the past. They do not invent the future. Doing that requires moral imagination, and that’s something only humans can provide." -- Cathy O'Neil in *Weapons of Math Destruction*
BUILDING FAIRER AI-ENABLED SYSTEMS

Christian Kaestner
(with slides from Eunsuk Kang)


LEARNING GOALS

- Understand different definitions of fairness
- Discuss methods for measuring fairness
- Design and execute tests to check for bias/fairness issues
- Understand fairness interventions during data acquisition
- Apply engineering strategies to build more fair systems
- Diagnose potential ethical issues in a given system
- Evaluate and apply mitigation strategies
TWO PARTS

Fairness assessment in the model
Formal definitions of fairness properties
Testing a model's fairness
Constraining a model for fairer results

System-level fairness engineering
Requirements engineering
Fairness and data acquisition
Team and process considerations
FAIRNESS IS STILL AN ACTIVELY STUDIED & DISPUTED CONCEPT!

Source: Mortiz Hardt, https://fairmlclass.github.io/
FAIRNESS THROUGH BLINDNESS

Anti-classification: Ignore/eliminate sensitive attributes from dataset, e.g., remove gender and race from a credit card scoring system

Advantages? Problems?
TESTING ANTI-CLASSIFICATION

Straightforward invariant for classifier $f$ and protected attribute $p$:

$$\forall x. f(x[p \leftarrow 0]) = f(x[p \leftarrow 1])$$

*(does not account for correlated attributes)*

Test with random input data (see prior lecture on Automated Random Testing) or on any test data

Any single inconsistency shows that the protected attribute was used. Can also report percentage of inconsistencies.

CLASSIFICATION PARITY

Classification error is equal across groups

INDEPENDENCE

(aka statistical parity, demographic parity, disparate impact, group fairness)

\[ P[R = 1 | A = 0] = P[R = 1 | A = 1] \text{ or } R \perp A \]

- Acceptance rate (i.e., percentage of positive predictions) must be the same across all groups
- Prediction must be independent of the sensitive attribute
- Example:
  - The predicted rate of recidivism is the same across all races
  - Chance of promotion the same across all genders
EXERCISE: CANCER DIAGNOSIS

<table>
<thead>
<tr>
<th>True Positives (TPs)</th>
<th>False Positives (FPs)</th>
<th>True Negatives (TNs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>4</td>
<td>974</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>486</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>488</td>
</tr>
</tbody>
</table>

- 1000 data samples (500 male & 500 female patients)
- What's the overall recall & precision?
- Does the model achieve independence
CALIBRATION TO ACHIEVE INDEPENDENCE

Select different thresholds for different groups to achieve prediction parity:

\[ P[R > t_0 | A = 0] = P[R > t_1 | A = 1] \]

Lowers bar for some groups -- equity, not equality
SEPARATION / EQUALIZED ODDS

Prediction must be independent of the sensitive attribute conditional on the target variable: $R \perp A \mid Y$

Same true positive rate across groups:

$$P[R = 0 \mid Y = 1, A = 0] = P[R = 0 \mid Y = 1, A = 1]$$

And same false positive rate across groups:

$$P[R = 1 \mid Y = 0, A = 0] = P[R = 1 \mid Y = 0, A = 1]$$

Example: A person with good credit behavior score should be assigned a good score with the same probability regardless of gender.
Equality

The assumption is that everyone benefits from the same supports. This is equal treatment.

Equity

Everyone gets the supports they need (this is the concept of “affirmative action”), thus producing equity.

Justice

All 3 can see the game without supports or accommodations because the cause(s) of the inequity was addressed. The systemic barrier has been removed.
REVIEW OF CRITERIA SO FAR:

Recidivism scenario: Should a person be detained?

- Anti-classification: ?
- Independence: ?
- Separation: ?
CAN WE ACHIEVE FAIRNESS DURING THE LEARNING PROCESS?

- Data acquisition:
  - Collect additional data if performance is poor on some groups

- Pre-processing:
  - Clean the dataset to reduce correlation between the feature set and sensitive attributes

- Training-time constraint
  - ML is a constraint optimization problem (minimize errors)
  - Impose additional parity constraint into ML optimization process (e.g., as part of the loss function)

- Post-processing
  - Adjust the learned model to be uncorrelated with sensitive attributes
  - Adjust thresholds

- (Still active area of research! Many new techniques published each year)
TRADE-OFFS: ACCURACY VS FAIRNESS

- Fairness constraints possible models
- Fairness constraints often lower accuracy for some group

PICKING FAIRNESS CRITERIA

- Requirements engineering problem!
- What's the goal of the system? What do various stakeholders want? How to resolve conflicts?

http://www.datasciencepublicpolicy.org/projects/aequitas/
FAIRNESS MUST BE CONSIDERED THROUGHOUT THE ML LIFECYCLE!

Fairness-aware Machine Learning, Bennett et al., WSDM Tutorial (2019).
PRACTITIONER CHALLENGES

- Fairness is a system-level property
  - consider goals, user interaction design, data collection, monitoring, model interaction (properties of a single model may not matter much)
- Fairness-aware data collection, fairness testing for training data
- Identifying blind spots
  - Proactive vs reactive
  - Team bias and (domain-specific) checklists
- Fairness auditing processes and tools
- Diagnosis and debugging (outlier or systemic problem? causes?)
- Guiding interventions (adjust goals? more data? side effects? chasing mistakes? redesign?)
- Assessing human bias of humans in the loop

THE ROLE OF REQUIREMENTS ENGINEERING

- Identify system goals
- Identify legal constraints
- Identify stakeholders and fairness concerns
- Analyze risks with regard to discrimination and fairness
- Analyze possible feedback loops (world vs machine)
- Negotiate tradeoffs with stakeholders
- Set requirements/constraints for data and model
- Plan mitigations in the system (beyond the model)
- Design incident response plan
- Set expectations for offline and online assurance and monitoring
BEST PRACTICES: TASK DEFINITION

- Clearly define the task & model’s intended effects
- Try to identify and document unintended effects & biases
- Clearly define any fairness requirements
- *Involve diverse stakeholders & multiple perspectives*
- Refine the task definition & be willing to abort

Bias can be introduced at any stage of the data pipeline

<table>
<thead>
<tr>
<th>Data Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional: biases due to platform affordances and algorithms</td>
</tr>
<tr>
<td>Normative: biases due to community norms</td>
</tr>
<tr>
<td>External: biases due to phenomena outside social platforms</td>
</tr>
<tr>
<td>Non-individuals: e.g., organizations, automated agents</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquisition: biases due to, e.g., API limits</td>
</tr>
<tr>
<td>Querying: biases due to, e.g., query formulation</td>
</tr>
<tr>
<td>Filtering: biases due to removal of data “deemed” irrelevant</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cleaning: biases due to, e.g., default values</td>
</tr>
<tr>
<td>Enrichment: biases from manual or automated annotations</td>
</tr>
<tr>
<td>Aggregation: e.g., grouping, organizing, or structuring data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qualitative Analyses: lack generalizability, interpret. biases</td>
</tr>
<tr>
<td>Descriptive Statistics: confounding bias, obfuscated measurements</td>
</tr>
<tr>
<td>Prediction &amp; Inferences: data representation, perform. variations</td>
</tr>
<tr>
<td>Observational studies: peer effects, select. bias, ignorability</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metrics: e.g., reliability, lack of domain insights</td>
</tr>
<tr>
<td>Interpretation: e.g., contextual validity, generalizability</td>
</tr>
<tr>
<td>Disclaimers: e.g., lack of negative results and reproducibility</td>
</tr>
</tbody>
</table>

A process for documenting datasets
Based on common practice in the electronics industry, medicine
Purpose, provenance, creation, composition, distribution: Does the dataset relate to people? Does the dataset identify any subpopulations?

Datasheets for Dataset, Gebru et al., (2019).
MODEL CARDS

Model Card - Toxicity in Text

Model Details
- The TOXICITY classifier provided by Perspective API [32], trained to predict the likelihood that a comment will be perceived as toxic.
- Convolutional Neural Network.

Intended Use
- Intended to be used for a wide range of use cases such as supporting human moderation and providing feedback to comment authors.
- Not intended for fully automated moderation.
- Not intended to make judgments about specific individuals.

Factors
- Identity terms referencing frequently attacked groups, focusing on sexual orientation, gender identity, and race.

Metrics
- Pinned AUC, as presented in [11], which measures threshold-agnostic separability of toxic and non-toxic comments for each group, within the context of a background distribution of other groups.

Ethical Considerations
- Following [31], the Perspective API uses a set of values to guide their work. These values are Community, Transparency, Inclusivity, Privacy, and Topic-neutrality. Because of various considerations, the model has certain limitations.

Training Data
- Proprietary from Perspective API. Following details in [11] and [32], this includes comments from online forums such as Wikipedia and New York Times, with crowdsourced labels of whether the comment is "toxic".
- "Toxic" is defined as "a rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion."

Evaluation Data
- A synthetic test set generated using a template-based approach, as suggested in [11], where identity terms are swapped into a variety of template sentences.
- Synthetic data is valuable here because [11] shows that real data often has disproportionate amounts of toxicity directed at specific groups. Synthetic data ensures that we evaluate on data that represents both toxic and non-toxic statements referencing a variety of groups.

Caveats and Recommendations
- Synthetic test data covers only a small set of very specific comments. While these are designed to be representative of common use cases and concerns, it is not comprehensive.

see also https://modelcards.withgoogle.com/about

HOMEWORK 6: FAIRNESS

(credit scoring + recommendation, model + system)
Required reading: Data Skeptic Podcast Episode “Black Boxes are not Required” with Cynthia Rudin (32min) or Rudin, Cynthia. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead." Nature Machine Intelligence 1, no. 5 (2019): 206-215.

LEARNING GOALS

• Understand the importance of and use cases for interpretability
• Explain the tradeoffs between inherently interpretable models and post-hoc explanations
• Measure interpretability of a model
• Select and apply techniques to debug/provide explanations for data, models and model predictions
• Eventuate when to use interpretable models rather than ex-post explanations
DETECTING ANOMALOUS COMMITS

IS THIS RECIDIVISM MODEL FAIR?

IF age between 18–20 and sex is male THEN predict arrest
ELSE
IF age between 21–23 and 2–3 prior offenses THEN predict arrest
ELSE
IF more than three priors THEN predict arrest
ELSE predict no arrest

WHAT FACTORS GO INTO PREDICTING STROKE RISK?

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Congestive Heart Failure</td>
<td>1 point</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Hypertension</td>
<td>1 point</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Age ≥ 75</td>
<td>1 point</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Diabetes Mellitus</td>
<td>1 point</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Prior Stroke or Transient Ischemic Attack</td>
<td>2 points</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ADD POINTS FROM ROWS 1–5  SCORE

<table>
<thead>
<tr>
<th>SCORE</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>STROKE RISK</td>
<td>1.9%</td>
<td>2.8%</td>
<td>4.0%</td>
<td>5.9%</td>
<td>8.5%</td>
<td>12.5%</td>
<td>18.2%</td>
</tr>
</tbody>
</table>

IS THERE AN ACTUAL PROBLEM? HOW TO FIND OUT?
WHAT'S HAPPENING HERE?
The European Union General Data Protection Regulation extends the automated decision-making rights in the 1995 Data Protection Directive to provide a legally disputed form of a right to an explanation: "[the data subject should have] the right ... to obtain an explanation of the decision reached"

US Equal Credit Opportunity Act requires to notify applicants of action taken with specific reasons: "The statement of reasons for adverse action required by paragraph (a)(2)(i) of this section must be specific and indicate the principal reason(s) for the adverse action."

See also https://en.wikipedia.org/wiki/Right_to_explanation
DEBUGGING

- Why did the system make a wrong prediction in this case?
- What does it actually learn?
- What kind of data would make it better?
- How reliable/robust is it?
- How much does the second model rely on the outputs of the first?
- Understanding edge cases
CURIOSITY, LEARNING, DISCOVERY, SCIENCE

- What drove our past hiring decisions? Who gets promoted around here?
- What factors influence cancer risk? Recidivism?
- What influences demand for bike rentals?
- Which organizations are successful at raising donations and why?
Interpretability is the degree to which a human can understand the cause of a decision.

Interpretability is the degree to which a human can consistently predict the model’s result.

(No mathematical definition)
GOOD EXPLANATIONS ARE CONTRASTIVE

Counterfactuals. *Why this, rather than a different prediction?*

*Your loan application has been declined. If your savings account had had more than $100 your loan application would be accepted.*

Partial explanations often sufficient in practice if contrastive
INHERENTLY INTERPRETABLE MODELS: SPARSE LINEAR MODELS

\[ f(x) = \alpha + \beta_1 x_1 + \ldots + \beta_n x_n \]

Truthful explanations, easy to understand for humans

Easy to derive contrastive explanation and feature importance

Requires feature selection/regularization to minimize to few important features (e.g. Lasso); possibly restricting possible parameter values

<table>
<thead>
<tr>
<th></th>
<th>Congestive Heart Failure</th>
<th>1 point</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>Hypertension</td>
<td>1 point</td>
<td>+</td>
</tr>
<tr>
<td>3.</td>
<td>Age ( \geq 75 )</td>
<td>1 point</td>
<td>+</td>
</tr>
<tr>
<td>4.</td>
<td>Diabetes Mellitus</td>
<td>1 point</td>
<td>+</td>
</tr>
<tr>
<td>5.</td>
<td>Prior Stroke or Transient Ischemic Attack</td>
<td>2 points</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td><strong>ADD POINTS FROM ROWS 1–5</strong></td>
<td></td>
<td><strong>SCORE</strong></td>
</tr>
<tr>
<td></td>
<td><strong>=</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCORE</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>STROKE RISK</td>
<td>1.9%</td>
<td>2.8%</td>
<td>4.0%</td>
<td>5.9%</td>
<td>8.5%</td>
<td>12.5%</td>
<td>18.2%</td>
</tr>
</tbody>
</table>
INHERENTLY INTERPRETABLE MODELS: DECISION TREES

Easy to interpret up to a size

Possible to derive counterfactuals and feature importance

Unstable with small changes to training data

IF age between 18–20 and sex is male THEN predict arrest
ELSE IF age between 21–23 and 2–3 prior offenses THEN predict arrest
ELSE IF more than three priors THEN predict arrest
ELSE predict no arrest
POST-HOC EXPLANATIONS OF BLACK-BOX MODELS

(large research field, many approaches, much recent research)

GLOBAL SURROGATES

1. Select dataset X (previous training set or new dataset from same distribution)
2. Collect model predictions for every value \( y_i = f(x_i) \)
3. Train inherently interpretable model \( g \) on \( (X,Y) \)
4. Interpret surrogate model \( g \)

Can measure how well \( g \) fits \( f \) with common model quality measures, typically \( R^2 \)

Advantages? Disadvantages?
LIME EXAMPLE

PARTIAL DEPENDENCE PLOT EXAMPLE

Bike rental in DC

Source: Christoph Molnar. "Interpretable Machine Learning." 2019
INDIVIDUAL CONDITIONAL EXPECTATION (ICE)

Similar to PDP, but not averaged; may provide insights into interactions

Source: Christoph Molnar. "Interpretable Machine Learning." 2019
FEATURE IMPORTANCE EXAMPLE

Source: Christoph Molnar. "Interpretable Machine Learning." 2019
## EXAMPLE: ANCHORS

<table>
<thead>
<tr>
<th>If</th>
<th>Predict</th>
</tr>
</thead>
<tbody>
<tr>
<td>adult</td>
<td></td>
</tr>
<tr>
<td>No capital gain or loss, never married</td>
<td>$\leq 50K$</td>
</tr>
<tr>
<td>Country is US, married, work hours $&gt;$ 45</td>
<td>$&gt; 50K$</td>
</tr>
<tr>
<td>rcdv</td>
<td></td>
</tr>
<tr>
<td>No priors, no prison violations and crime not against property</td>
<td>Not rearrested</td>
</tr>
<tr>
<td>Male, black, 1 to 5 priors, not married, and crime not against property</td>
<td>Re-arrested</td>
</tr>
<tr>
<td>lending</td>
<td></td>
</tr>
<tr>
<td>FICO score $\leq 649$</td>
<td>Bad Loan</td>
</tr>
<tr>
<td>649 $\leq$ FICO score $\leq$ 699 and $$5,400 \leq$ loan amount $\leq$ $$10,000$</td>
<td>Good Loan</td>
</tr>
</tbody>
</table>
EXAMPLE: ANCHORS

COUNTERFACTUAL EXPLANATIONS

if X had not occurred, Y would not have happened

Your loan application has been declined. If your savings account had had more than $100 your loan application would be accepted.

-> Smallest change to feature values that result in given output
MULTIPLE COUNTERFACTUALS

Often long or multiple explanations

Your loan application has been declined. If your savings account...

Your loan application has been declined. If your lived in...

Report all or select "best" (e.g. shortest, most actionable, likely values)

(Rashomon effect)
GAMING/ATTACKING THE MODEL WITH EXPLANATIONS?

Does providing an explanation allow customers to 'hack' the system?

- Loan applications?
- Apple FaceID?
- Recidivism?
- Auto grading?
- Cancer diagnosis?
- Spam detection?
GAMING THE MODEL WITH EXPLANATIONS?
EXAMPLE: PROTOTYPES AND CRITICISMS

EXAMPLE: INFLUENTIAL INSTANCE

Source: Christoph Molnar. "Interpretable Machine Learning." 2019
WHAT DISTINGUISHES AN INFLUENTIAL INSTANCE FROM A NON-INFLUENTIAL INSTANCE?

Compute influence of every data point and create new model to explain influence in terms of feature values

Which features have a strong influence but little support in the training data?

Source: Christoph Molnar. "Interpretable Machine Learning." 2019
Tell the user when a lack of data might mean they’ll need to use their own judgment. Don’t be afraid to admit when a lack of data could affect the quality of the AI recommendations.

Source: People + AI Guidebook, Google
CASE STUDY: FACEBOOK'S FEED CURATION
"STOP EXPLAINING BLACK BOX MACHINE LEARNING MODELS FOR HIGH STAKES DECISIONS AND USE INTERPRETABLE MODELS INSTEAD."

Microsoft AI principles

We put our responsible AI principles into practice through the Office of Responsible AI (ORA) and the AI, Ethics, and Effects in Engineering and Research (Aether) Committee. The Aether Committee advises our leadership on the challenges and opportunities presented by AI innovations. ORA sets our rules and governance processes, working closely with teams across the company to enable the effort.

Learn more about our approach

**Fairness**
AI systems should treat all people fairly
- Play video on fairness

**Inclusiveness**
AI systems should empower everyone and engage people
- Play video on inclusiveness

**Reliability & Safety**
AI systems should perform reliably and safely
- Play video on reliability

**Transparency**
AI systems should be understandable
- Play video on transparency

**Privacy & Security**
AI systems should be secure and respect privacy
- Play video on privacy

**Accountability**
People should be accountable for AI systems
- Play video on accountability
The world of artificial intelligence is constantly evolving, and certainly so is the legal and regulatory environment.
VERSIONING, PROVENANCE, AND REPRODUCABILITY

Christian Kaestner

LEARNING GOALS

• Judge the importance of data provenance, reproducibility and explainability for a given system
• Create documentation for data dependencies and provenance in a given system
• Propose versioning strategies for data and models
• Design and test systems for reproducibility
DATA PROVENANCE

- Track origin of all data
  - Collected where?
  - Modified by whom, when, why?
  - Extracted from what other data or model or algorithm?
- ML models often based on data derived from many sources through many steps, including other models
VERSIONING DATASETS

- Store copies of entire datasets (like Git)
- Store deltas between datasets (like Mercurial)
- Offsets in append-only database (like Kafka offset)
- History of individual database records (e.g. S3 bucket versions)
  - some databases specifically track provenance (who has changed what entry when and how)
  - specialized data science tools eg Hangar for tensor data
- Version pipeline to recreate derived datasets ("views", different formats)
  - e.g. version data before or after cleaning?

- Often in cloud storage, distributed
- Checksums often used to uniquely identify versions
- Version also metadata
VERSIONING PIPELINES

- data
- hyperparameters
- pipeline
  - model
EXAMPLE: DVC

dvc add images
dvc run -d images -o model.p cnn.py
dvc remote add myrepo s3://mybucket
dvc push

- Tracks models and datasets, built on Git
- Splits learning into steps, incrementalization
- Orchestrates learning in cloud resources

https://dvc.org/
EXAMPLE: MODELDB

https://github.com/mitdbg/modeldb
EXAMPLE: MLFLOW

- Instrument pipeline with *logging* statements
- Track individual runs, hyperparameters used, evaluation results, and model files
### Listing Price Prediction

**Experiment ID:** 0  
**Artifact Location:** /Users/matei/mlflow/demo/mlruns/0

**Search Runs:** metrics.R2 > 0.24

**Filter Params:** alpha, lr  
**Filter Metrics:** rmse, r2

<table>
<thead>
<tr>
<th>Time</th>
<th>User</th>
<th>Source</th>
<th>Version</th>
<th>Parameters</th>
<th>MAE</th>
<th>R2</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>17:37</td>
<td>matei</td>
<td>linear.py</td>
<td>3a1995</td>
<td>alpha 0.5, I1_ratio 0.2</td>
<td>84.27</td>
<td>0.277</td>
<td>158.1</td>
</tr>
<tr>
<td>17:37</td>
<td>matei</td>
<td>linear.py</td>
<td>3a1995</td>
<td>alpha 0.2, I1_ratio 0.5</td>
<td>84.08</td>
<td>0.264</td>
<td>159.6</td>
</tr>
<tr>
<td>17:37</td>
<td>matei</td>
<td>linear.py</td>
<td>3a1995</td>
<td>alpha 0.5, I1_ratio 0.5</td>
<td>84.12</td>
<td>0.272</td>
<td>158.6</td>
</tr>
<tr>
<td>17:37</td>
<td>matei</td>
<td>linear.py</td>
<td>3a1995</td>
<td>alpha 0, I1_ratio 0</td>
<td>84.49</td>
<td>0.249</td>
<td>161.2</td>
</tr>
</tbody>
</table>

---

DEFINITIONS

- **Reproducibility**: the ability of an experiment to be repeated with minor differences from the original experiment, while achieving the same qualitative result.
- **Replicability**: ability to reproduce results exactly, achieving the same quantitative result; requires determinism.

In science, reproducing results under different conditions are valuable to gain confidence:
- "conceptual replication": evaluate same hypothesis with different experimental procedure or population
- many different forms distinguished "... replication" (e.g. close, direct, exact, independent, literal, nonexperimental, partial, retest, sequential, statistical, varied, virtual)

NONDETERMINISM

- Some machine learning algorithms are nondeterministic
  - Recall: Neural networks initialized with random weights
  - Recall: Distributed learning
- Many notebooks and pipelines contain nondeterminism
  - Depend on snapshot of online data (e.g., stream)
  - Depend on current time
  - Initialize random seed
- Different library versions installed on the machine may affect results
- (Inference for a given model is usually deterministic)
PROJECT M3: MONITORING AND CONTINUOUS DEPLOYMENT

(containization, monitoring, canary releases, provenance)
SECURITY, ADVERSARIAL LEARNING, AND PRIVACY

Christian Kaestner

with slides from Eunsuk Kang


LEARNING GOALS

- Explain key concerns in security (in general and with regard to ML models)
- Analyze a system with regard to attacker goals, attack surface, attacker capabilities
- Describe common attacks against ML models, including poisoning attacks, evasion attacks, leaking IP and private information
- Measure robustness of a prediction and a model
- Understand design opportunities to address security threats at the system level
- Identify security requirements with threat modeling
- Apply key design principles for secure system design
- Discuss the role of AI in securing software systems
SECURITY AT THE MODEL LEVEL

- Various attack discussions, e.g. poisoning attacks
- Model robustness
- Attack detection
- ...

SECURITY AT THE SYSTEM LEVEL

- Requirements analysis
- System-level threat modeling
- Defense strategies beyond the model
- Security risks beyond the model
- ...

[213]
"CIA triad" of information security

- **Confidentiality**: Sensitive data must be accessed by authorized users only
- **Integrity**: Sensitive data must be modifiable by authorized users only
- **Availability**: Critical services must be available when needed by clients
ATTACKER GOALS AND INCENTIVES

- What is the attacker trying to achieve? Undermine one or more security requirements
- Why does the attacker want to do this?

Example goals and incentives in Garmin/college admission scenario?
• Availability: Inject mislabeled training data to damage model quality
  ▪ 3% poisoning => 11% decrease in accuracy (Steinhardt, 2017)
• Attacker must have some access to the training set
  ▪ models trained on public data set (e.g., ImageNet)
  ▪ retrained automatically on telemetry
POISONING ATTACK: INTEGRITY

- Insert training data with seemingly correct labels
- More targeted than availability attacks
  - Cause misclassification from one specific class to another

*Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks*, Shafahi et al. (2018)
POISONING ATTACK IN WEB SHOP?

Antique Box Ugears, 3D Mechanical Treasure Models, Self-Assembling Precut Wooden Gift, DIY Craft Set

🌟🌟🌟🌟🌟 261

$41.90 $44.90
FREE Delivery for Prime members
Only 1 left in stock - order soon.
More Buying Choices
$38.89 (44 new offers)
Ages: 14 years and up

ROKR 3D Wooden Puzzle for Adults-Mechanical Train Model Kits-Brain Teaser Puzzles-Vehicle Building Kits-Unique Gi...

🌟🌟🌟🌟🌟 44

$22.99
✓prime FREE One-Day
Get it Tomorrow, Jul 26
Ages: 14 years and up

Wooden Puzzles for Toddlers, Aitey Wooden Alphabet Number Puzzles Toddler Learning Puzzle Toys for Kids Ages 2 3 4 (Set of...

🌟🌟🌟🌟🌟 283
✓prime FREE One-Day
Get it Tomorrow, Jul 26
More Buying Choices
$23.99 (2 used & new offers)
Ages: 12 months and up
ROKR 3D Assembly Wooden Puzzle Brain Teaser Game Mechanical Gears Set Model Kit Marble Run Set Unique Craft...

Unidragon Wooden Jigsaw Puzzles - Unique Shape Jigsaw Pieces Best Gift for Adults and Kids Alluring Fox 7 x 9.2 in (18...

KINGZHOU Hexagon Tangram Classic Handmade Wooden Puzzle for Children and Adults Challenging Puzzles Brain...

$20.99

$49.99

$9.98
DEFENSE AGAINST POISONING ATTACKS

ATTACKS ON INPUT DATA (EVASION ATTACKS, ADVERSARIAL EXAMPLES)

- Add noise to an existing sample & cause misclassification
  - achieve specific outcome (evasion attack)
  - circumvent ML-based authentication like FaceID (impersonation attack)
- Attack at inference time
Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition, Sharif et al. (2016).
GENERATING ADVERSARIAL EXAMPLES

- see counterfactual explanations
- Find similar input with different prediction
  - targeted (specific prediction) vs untargeted (any wrong prediction)
- Many similarity measures (e.g., change one feature vs small changes to many features)
  - \[ x^* = x + \arg\min\{ |z| : f(x + z) = t \} \]
- Attacks more affective which access to model internals, but also black-box attacks (with many queries to the model) feasible
  - With model internals: follow the model's gradient
  - Without model internals: learn surrogate model
  - With access to confidence scores: heuristic search (eg. hill climbing)
NO MODEL IS FULLY ROBUST

- Every useful model has at least one decision boundary (ideally at the real task decision boundary)
- Predictions near that boundary are not (and should not) be robust
ASSURING ROBUSTNESS

- Much research, many tools and approaches (especially for DNN)
- Formal verification
  - Constraint solving or abstract interpretation over computations in neuron activations
  - Conservative abstraction, may label robust inputs as not robust
  - Currently not very scalable
- Sampling
  - Sample within distance, compare prediction to majority prediction
  - Probabilistic guarantees possible (with many queries, e.g., 100k)
PRACTICAL USE OF ROBUSTNESS

- Defense and safety mechanism at inference time
  - Check robustness of each prediction at runtime
  - Handle inputs with non-robust predictions differently (e.g. discard, low confidence)
  - Significantly raises cost of prediction (e.g. 100k model inferences or constraint solving at runtime)

- Testing and debugging
  - Identify training data near model's decision boundary (i.e., model robust around all training data?)
  - Check robustness on test data
  - Evaluate distance for adversarial attacks on test data

(most papers on the topic focus on techniques and evaluate on standard benchmarks like handwritten numbers, but do not discuss practical scenarios)
Google Catches Bing Copying; Microsoft Says 'So What?'
what would bing do
what would bing do bnet
what would bing crosby do

Google Search  I'm Feeling Lucky
NetFlix Cancels Recommendation Contest After Privacy Lawsuit
Netflix is canceling its second $1 million Netflix Prize to settle a legal challenge that it breached customer privacy as part of the first contest's race for a better movie-recommendation engine.
GENERATIVE ADVERSARIAL NETWORKS

Real images → Sample → Discriminator → Disc. loss

Generator → Sample → Discriminator → Gen. loss
PROTOTYPICAL INPUTS WITH GANS
SECURITY AT THE SYSTEM LEVEL

security is more than model robustness
defenses go beyond hardening models
Fun to Build Detailed Steam Engine Model
Reviewed in the United States on September 17, 2019
Verified Purchase

The wooden steam engine model made by ROKR is called a “3D Puzzle Kit.” I completed without great difficulty it over the span of two days. The model is made from laser cut wood parts that need to be punched out (carefully) from eight large flat wooden panels. The individual parts are labeled by board and number. There is no glue used, the pieces are all pressed together (again carefully.)

The model is fairly large at 14 inches long, 9 1/2 inches high and 2 inches wide. It weighs almost 3
2 Comments

Add a public comment...

Highlighted comment

Blaise Norman 3 weeks ago
Good videos. You deserve more subscribers. Check FollowSM main channel to promote my videos.

Like Dislike Heart Reply

MALEK97 3 days ago
Thank you so much for sharing this amazing content!

Like Reply

Options:
- Pin
- Remove
- Report
- Hide user from channel
Too many attempts. Try again later.
ARCHITECTURE DIAGRAM FOR THREAT MODELING

- Dynamic and physical architecture diagram
- Describes system components and users and their interactions
- Describe thrust boundaries
On-going arms race (mostly among researchers)
- Defenses proposed & quickly broken by noble attacks

Assume ML component is likely vulnerable
- Design your system to minimize impact of an attack

Remember: There may be easier ways to compromise system
- e.g., poor security misconfiguration (default password), lack of encryption, code vulnerabilities, etc.,
SECURE DESIGN PRINCIPLES

• Principle of Least Privilege
  ■ A component should be given the minimal privileges needed to fulfill its functionality
  ■ Goal: Minimize the impact of a compromised component

• Isolation
  ■ Components should be able to interact with each other no more than necessary
  ■ Goal: Reduce the size of trusted computing base (TCB)
  ■ TCB: Components responsible for establishing a security requirement(s)
  ■ If any of TCB compromised => security violation
  ■ Conversely, a flaw in non-TCB component => security still preserved!
  ■ In poor system designs, TCB = entire system
30 COMPANIES MERGING AI AND CYBERSECURITY TO KEEP US SAFE AND SOUND

Alyssa Schroer
July 12, 2019  Updated: July 15, 2020

...by the year 2021, cybercrime losses will
SAFETY

Christian Kaestner

With slides from Eunsuk Kang

Required Reading
LEARNING GOALS

• Understand safety concerns in traditional and AI-enabled systems
• Apply hazard analysis to identify risks and requirements and understand their limitations
• Discuss ways to design systems to be safe against potential failures
• Suggest safety assurance strategies for a specific project
• Describe the typical processes for safety evaluations and their limitations
SAFETY
CASE STUDY: SELF-DRIVING CAR
CHALLENGE: EDGE/UNKNOWN CASES

- Gaps in training data; ML will unlikely to cover all unknown cases
- Why is this a unique problem for AI? What about humans?
WHAT IS HAZARD ANALYSIS?

- **Hazard**: A condition or event that may result in undesirable outcome
  - e.g., "Ego vehicle is in risk of a collision with another vehicle."
- **Safety requirement**: Intended to eliminate or reduce one or more hazards
  - "Ego vehicle must always maintain some minimum safe distance to the leading vehicle."
- **Hazard analysis**: Methods for identifying hazards & potential root causes
ROBUSTNESS IN A SAFETY SETTING

- Does the model reliably detect stop signs?
- Also in poor lighting? In fog? With a tilted camera?
- With stickers taped to the sign?

TESTING FOR SAFETY

- Curate data sets for critical scenarios (see model quality lecture)
- Create test data for difficult settings (e.g. fog)
- Simulation feasible? Shadow deployment feasible?
NEGATIVE SIDE EFFECTS
Welcome to Universal Paperclips

> AutoClippers available for purchase

**Paperclips: 148**

**Business**

Available Funds: $9.50
Unsold Inventory: 89
Price per Clip: $0.25
Public Demand: 32%

**Marketing**

Level: 1
Cost: $100.00

**Manufacturing**

Clips per Second: 1

Wire: 852 inches
Cost: $26

**AutoClippers**

1
Cost: $6.10
PlayFun algorithm pauses the game of Tetris indefinitely to avoid losing.

When about to lose a hockey game, the PlayFun algorithm exploits a bug to make one of the players on the opposing team disappear from the map, thus forcing a draw.

Self-driving car rewarded for speed learns to spin in circles.

Self-driving car figures out that it can avoid getting penalized for driving too close to other cars by exploiting certain sensor vulnerabilities so that it can’t “see” how close it is getting.
ELEMENTS OF SAFE DESIGN

- **Assume**: Components will fail at some point
- **Goal**: Minimize the impact of failures on safety
- **Detection**
  - Monitoring
- **Control**
  - Graceful degradation (fail-safe)
  - Redundancy (fail over)
- **Prevention**
  - Decoupling & isolation
SAE J3016™ LEVELS OF DRIVING AUTOMATION

What does the human in the driver’s seat have to do?

**SAE LEVEL 0**
You are driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering.

**SAE LEVEL 1**
You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety.

**SAE LEVEL 2**
You are not driving when these automated driving features are engaged – even if you are seated in “the driver’s seat”.

**SAE LEVEL 3**
When the feature requests, you must drive.

**SAE LEVEL 4**
These automated driving features will not require you to take over driving.

**SAE LEVEL 5**

What do these features do?

**These are driver support features**
- These features are limited to providing warnings and momentary assistance.
- These features provide steering OR brake/acceleration support to the driver.
- These features provide steering AND brake/acceleration support to the driver.
- Example Features:
  - automatic emergency braking
  - blind spot warning
  - lane departure warning

**These are automated driving features**
- These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met.
- Example Features:
  - traffic jam chauffeur
  - local driverless taxi
  - pedals/steering wheel may or may not be installed

For a more complete description, please download a free copy of SAE J3016: https://www.sae.org/standards/content/J3016_201806/
<table>
<thead>
<tr>
<th>Being able to apply ML in safety-critical applications will be important to my organization in the future</th>
</tr>
</thead>
<tbody>
<tr>
<td>V&amp;V of features that rely on ML is recognized as a particularly challenging area in my organization</td>
</tr>
<tr>
<td>My organization is well-prepared for a future in which V&amp;V of safety-critical ML is commonplace</td>
</tr>
</tbody>
</table>

SAFETY ASSURANCE WITH ML COMPONENTS

- Consider ML components as unreliable, at most probabilistic guarantees
- Testing, testing, testing (+ simulation)
  - Focus on data quality & robustness
- *Adopt a system-level perspective!*
- Consider safe system design with unreliable components
  - Traditional systems and safety engineering
  - Assurance cases
- Understand the problem and the hazards
  - System level, goals, hazard analysis, world vs machine
  - Specify *end-to-end system behavior* if feasible
- Recent research on adversarial learning and safety in reinforcement learning
Recall: Legal vs ethical
Safety analysis not only for regulated domains (nuclear power plants, medical devices, planes, cars, ...)
Many end-user applications have a safety component

Examples?
ADDICTION
Robinhood Has Gamified Online Trading Into an Addiction

Tech’s obsession with addiction will hurt us all

Warning: This post contains a discussion of suicide.

Addiction is the inability to stop consuming a chemical or pursuing an activity although it’s causing harm.

I engage with almost every substance or behavior associated with addiction: alcohol, drugs, coffee, porn, sex, gambling, work, spending,
Facebook Executives Shut Down Efforts to Make the Site Less Divisive

The social-media giant internally studied how it polarizes users, then largely shelved the research

By Jeff Horwitz and Deepa Seetharaman
May 26, 2020 11:38 am ET
ENVIRONMENTAL: ENERGY CONSUMPTION
Creating an AI can be five times worse for the planet than a car

TECHNOLOGY  6 June 2019

By Donna Lu
FOSTERING INTERDISCIPLINARY TEAMS

(Process and Team Reflections)

Christian Kaestner

LEARNING GOALS

- Plan development activities in an inclusive fashion for participants in different roles
- Describe agile techniques to address common process and communication issues
DATA SCIENCE ROLES AT MICROSOFT

- Polymath
- Data evangelist
- Data preparer
- Data shaper
- Data analyzer
- Platform builder
- 50/20% moonlighter
- Insight actors

OTHER ROLES IN AI SYSTEMS PROJECTS?

- Domain specialists
- Business, management, marketing
- Project management
- Designers, UI experts
- Operations
- Lawyers
- Social scientists, ethics
- ...
HOW TO STRUCTURE TEAMS?

Mobile game; 50ish developers; distributed teams?
MYTHICAL MAN MONTH

Brooks's law: Adding manpower to a late software project makes it later

1975, describing experience at IBM developing OS/360
CONFLICTING GOALS?

Data Scientists

Compliance Lawyers
T-SHAPED PEOPLE

Broad-range generalist + Deep expertise

Figure: Jason Yip. Why T-shaped people?. 2018
MATRIX ORGANIZATION

- mgmt
- Project 1
- Project 2
- Project 3
- System programmers
- Application programmers
- QA
- Security
- Marketing
TEAM ISSUES: GROUPTHINK

WE NEED MORE DISSenting OPINIONS.

WE AGREE 100%
TEAM ISSUES: SOCIAL LOAFING
SUMMARY

(424 slides in 40 min)

Christian Kaestner
<table>
<thead>
<tr>
<th>TODAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Looking back at the semester</td>
</tr>
<tr>
<td>Discussion of future of SE4AI</td>
</tr>
<tr>
<td>Feedback for future semesters</td>
</tr>
</tbody>
</table>
THE FUTURE OF SOFTWARE ENGINEERING FOR AI-ENABLED SYSTEMS?
WHAT ARE NEW SOFTWARE ENGINEERING CHALLENGES?

Gaps? Research needs? Adaptation of existing methods?
ARE SOFTWARE ENGINEERS DISAPPEARING?

see also Andrej Karpathy. Software 2.0. Blog, 2017
Andrej Karpathy is the director of AI at Tesla and coined the term Software 2.0
ARE DATA SCIENTISTS DISAPPEARING?

AutoML 2.0: Is The Data Scientist Obsolete?

Ryohei Fujimaki  Contributor
COGNITIVE WORLD  Contributor Group ©
AI

Ryohei Fujimaki Ph.D. is founder and CEO of dotData
It's an AutoML World

The world of AutoML has been proliferating over the past few years - and with a recession looming, the notion of automating the

Ryohei Fujimaki. AutoML 2.0: Is The Data Scientist Obsolete? Forbes, 2020
However, AutoML does not spell the end of data scientists, as it doesn’t “AutoSelect” a business problem to solve, it doesn’t AutoSelect indicative data, it doesn’t AutoAlign stakeholders, it doesn’t provide AutoEthics in the face of potential bias, it doesn’t provide AutoIntegration with the rest of your product, and it doesn’t provide AutoMarketing after the fact. -- Frederik Bussler

Frederik Bussler. Will AutoML Be the End of Data Scientists?, Blog 2020
SE4AI RESEARCH: MORE SE POWER TO DATA SCIENTISTS?

SE4AI RESEARCH: MORE DS POWER TO SOFTWARE ENGINEERS?
Virtually *everyone* is / will soon be building ML applications. Only few can afford dedicated software engineers to team up with, or SE education for themselves. It would be more inclusive to build SE into the ML processes more fundamentally, so that everyone could build better.
ANALOGY
(better tools don't replace the knowledge to use them)
This is an education problem, more than a research problem.

Interdisciplinary teams, mutual awareness and understanding

Software engineers will play an essential role
DEVOPS AS A ROLE MODEL

Joint responsibilities, joint processes, joint tools, joint vocabulary
FEEDBACK

- What was useful?
- What could be improved?
- Ideas for better remote teaching?
THANK YOU!