An Open Source Toolkit for R to Mitigate Discrimination and Bias in Machine Learning Models

Stacey Ronaghan, Gabriela de Queiroz, Saishruthi Swaminathan

slides: bit.ly/openup-summit

Agenda

Responsible AI

AI Fairness 360

Demo

Q&A

Responsible AI

- AI Opportunities
  - Increased Revenue
  - Efficiencies
- AI Risks
  - Harm to Users
  - Harm to Business
- A Solution
  - Regulation
  - Ethical & Moral Practices

slides: bit.ly/openup-summit

DESIGN

- Human Centric
- Optimization Metrics

DATA

- Representative
- Protected

MODEL

- Interpretable
- Fair

MONITORING

- Staged rollout
- Feedback loop

ACCOUNTABILITY

- Transparency
- Responsibilities

Responsible ML Pipeline

Responsible AI Benefits

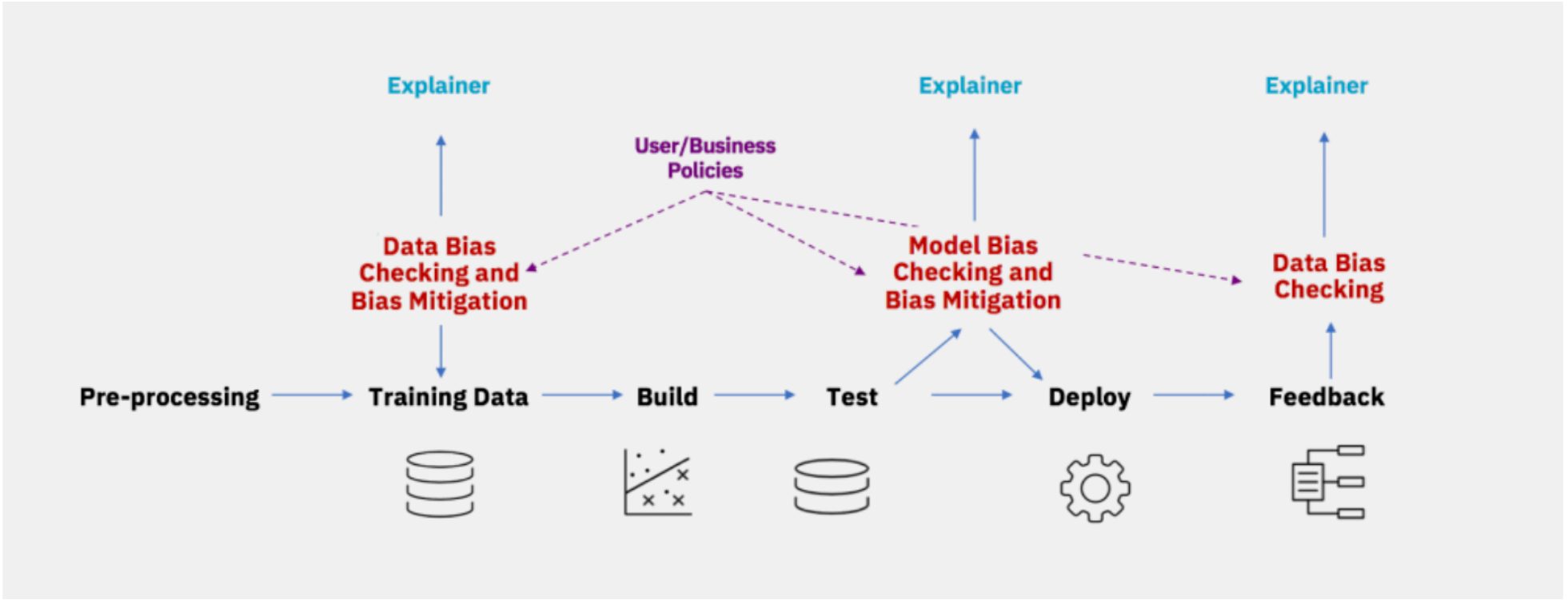
- Prevent harm
- Build an inclusive product
- Delightful customer experiences
- Responsible branding

AIF 360 Library

slides: bit.ly/openup-summit

AIF360

- AIF360 toolkit is an open-source library to help detect and remove bias in machine learning models.
- AIF360 translates algorithmic research from the lab into practice.
- Applicable domains include finance, human capital management, healthcare, and education.
- Toolbox
  - Fairness metrics
  - Fairness metric explanations
  - Bias mitigation algorithms



Mitigating bias throughout the AI application lifecycle

slides: bit.ly/openup-summit

LF AI

IBM donated AIF360 to LFAI

Blog Post

IBM and LFAI move forward on trustworthy and responsible AI

☆ Save 👍 Like

IBM donates Trusted AI toolkits to the Linux Foundation AI

By Todd Moore, Sriram Raghavan, Aleksandra Mojsilovic

Published June 29, 2020

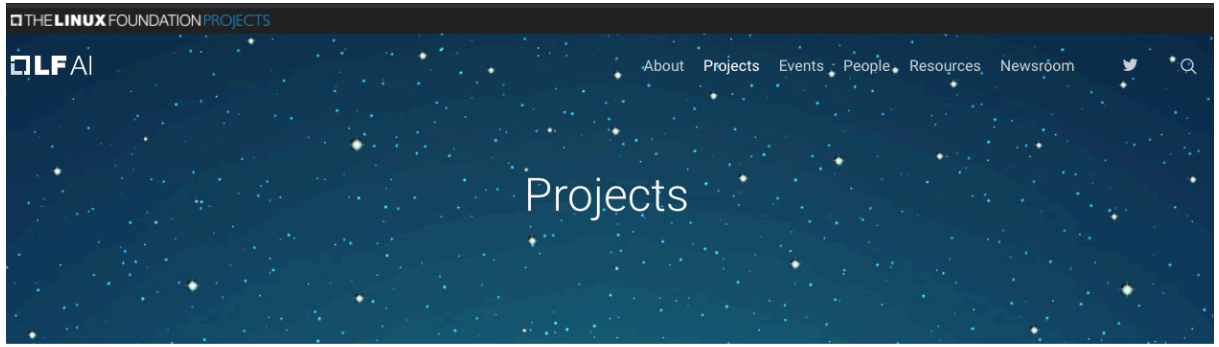
For over a century, IBM has created technologies that profoundly changed how humans work and live: the personal computer, ATM, magnetic tape, Fortran Programming Language, floppy disk, scanning tunneling microscope, relational database, and most recently, quantum computing, to name a few. With trust as one of our core principles, we've spent the past century creating products our clients can trust and depend on, guiding their responsible adoption and use, and respecting the needs and values of all users and communities we serve.

Our current work in artificial intelligence (AI) is bringing a transformation of similar scale to the world today. We infuse these guiding principles of trust and transparency into all of our work in AI. Our responsibility is to not only make the technical breakthroughs required to make AI trustworthy and ethical, but to ensure these trusted algorithms work as intended in real-world AI deployments.

https://developer.ibm.com/technologies/artificial-intelligence/blogs/ibm-and-lfai-move-forward-on-trustworthy-and-responsible-ai/

LF AI

AIF360 is being incubated under Linux Foundation AI



Acumos AI

Acumos AI

Open source framework to build, share and deploy AI applications

Acumos is an open source platform, which supports design, integration and deployment of AI models. Furthermore, it offers an AI marketplace that empowers data scientists to publish adaptive AI models, while shielding them from the need to custom develop fully integrated solutions.

Learn More

ADLIK

Adlik

Open source toolkit for accelerating deep learning inference

Adlik is an end-to-end optimizing framework for deep learning models. The goal of Adlik is to accelerate deep learning inference process both on cloud and embedded environments.

Learn More

Adversarial Robustness Toolbox

Adversarial Robustness Toolbox

Open source tools to evaluate, defend, certify and verify Machine Learning models and applications against adversarial threats

Adversarial Robustness Toolbox (ART) provides tools that enable developers and researchers to evaluate, defend, certify and verify Machine Learning models and applications against the adversarial threats.

Learn More

AI Explainability 360

AI Explainability 360

Open source toolkit that can help users better understand the ways that machine learning models predict labels

AI Explainability 360 is an open source toolkit that can help users better understand the ways that machine learning models predict labels using a wide variety of techniques throughout the AI application lifecycle.

Learn More

AI Fairness 360

AI Fairness 360

Open source toolkit that can help users understand and mitigate bias in machine learning models throughout the AI application lifecycle

AI Fairness 360 is an extensible open source toolkit that can help users understand and mitigate bias in machine learning models throughout the AI application lifecycle.

Learn More

Metrics

A quantification of unwanted bias in training data or models.

- Individual vs. Group Fairness, or Both

  Equal treatment under protected attributes

- Group Fairness: Data vs Model

  Measure at different points in ML pipeline: pre-,in-,post-processing

- Group Fairness: We're All Equal vs What You See is What You Get

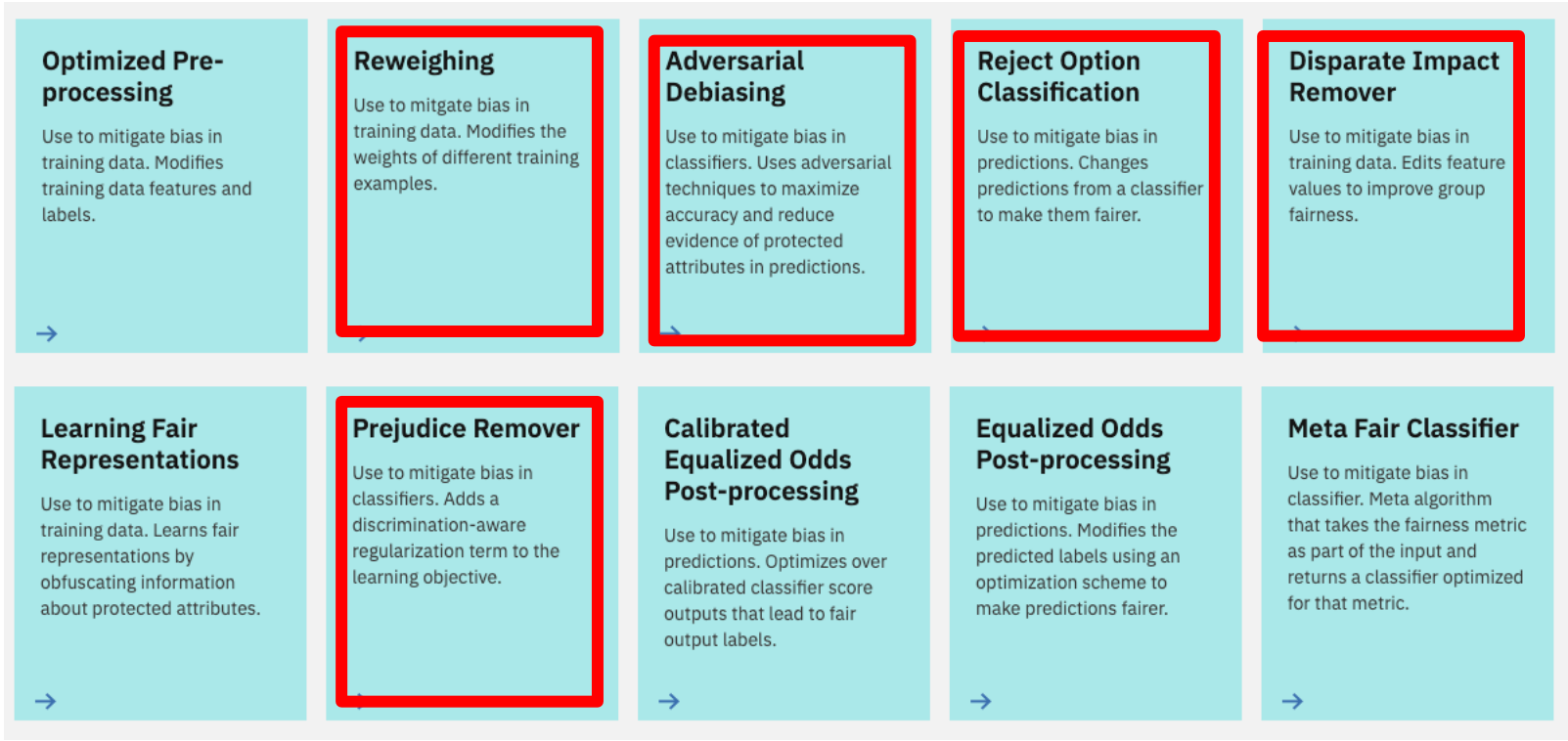
  - WAE: Predicted future performance is influenced by bias in measurement.
  - WISYWIG: Predicted future performance correlates only with raw score.

- Group Fairness: Ratios vs Differences

Algorithms

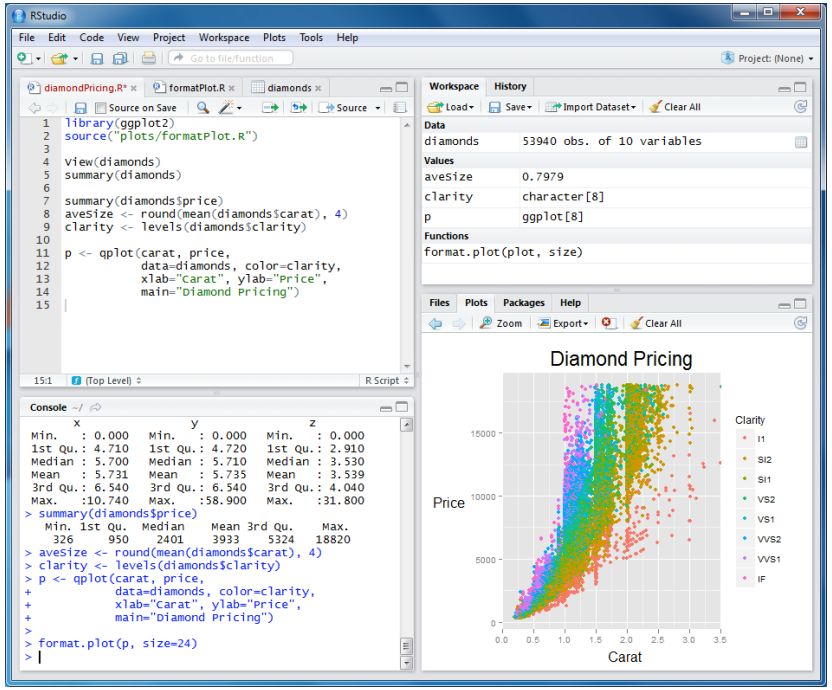
- Bias mitigation algorithms attempt to improve the fairness metrics by modifying the training data, the learning algorithm, or the predictions.
- These algorithm categories are known as pre-processing, in-processing, and post-processing, respectively.

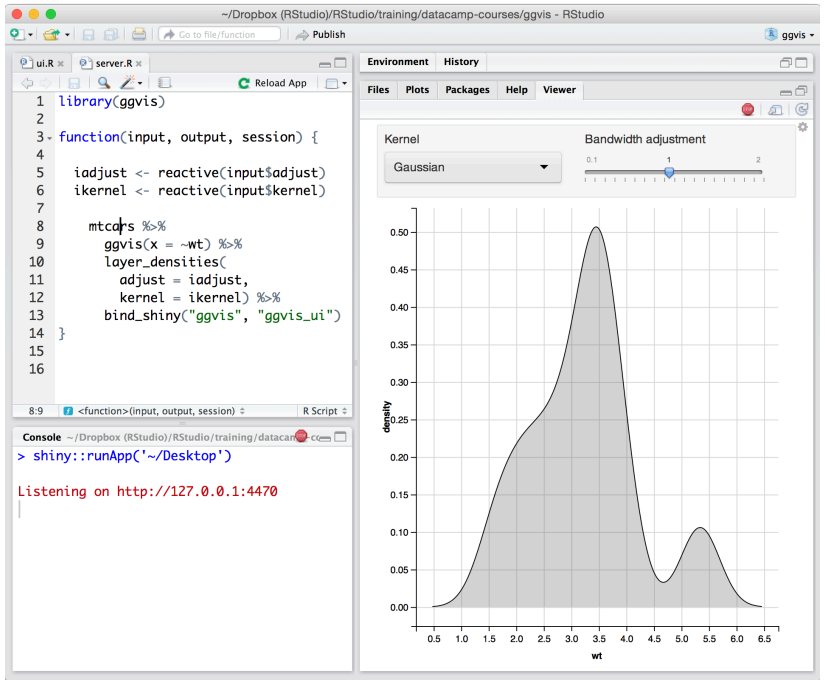
Algorithms

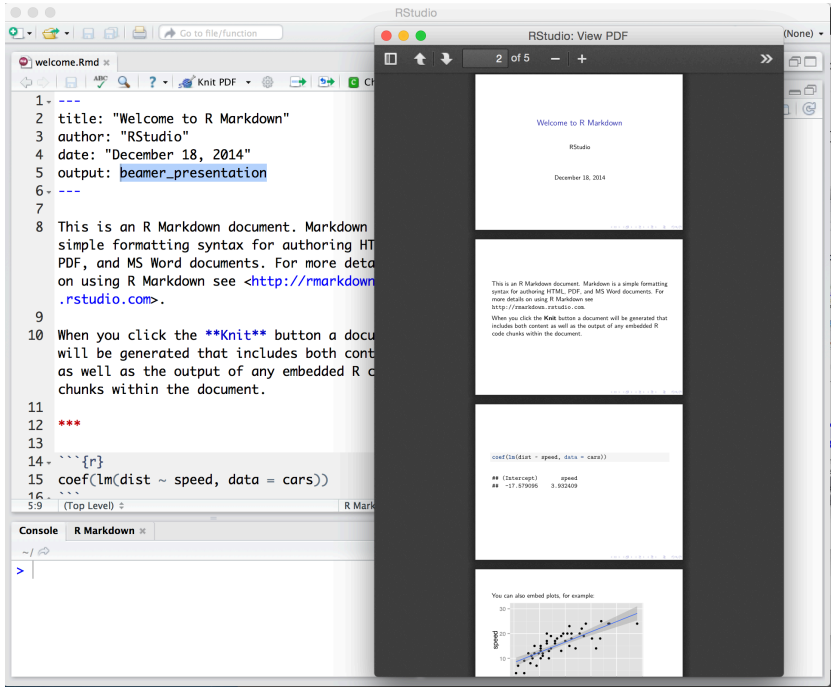

Optimized Pre-processing
Use to mitigate bias in training data. Modifies training data features and labels.
Reweighing
Use to mitgate bias in training data. Modifies the weights of different training examples.
Adversarial Debiasing
Use to mitigate bias in classifiers. Uses adversarial techniques to maximize accuracy and reduce evidence of protected attributes in predictions.
Reject Option Classification
Use to mitigate bias in predictions. Changes predictions from a classifier to make them fairer.
Disparate Impact Remover
Use to mitigate bias in training data. Edits feature values to improve group fairness.
Learning Fair Representations
Use to mitigate bias in training data. Learns fair representations by obfuscating information about protected attributes.
Prejudice Remover
Use to mitigate bias in classifiers. Adds a discrimination-aware regularization term to the learning objective.
Calibrated Equalized Odds Post-processing
Use to mitigate bias in predictions. Optimizes over calibrated classifier score outputs that lead to fair output labels.
Equalized Odds Post-processing
Use to mitigate bias in predictions. Modifies the predicted labels using an optimization scheme to make predictions fairer.
Meta Fair Classifier
Use to mitigate bias in classifier. Meta algorithm that takes the fairness metric as part of the input and returns a classifier optimized for that metric.

Using AIF360 in R

slides: bit.ly/openup-summit







What is R?

R is a programming language used by Data Scientists, Researchers, by anyone working with data.

It has more than 10,000 packages, it is free, open source, powerful and highly extensible

You can run R within RStudio - *integrated development environment (IDE)* that provides an interface by adding many convenient features and tools

R Package Installation

You can install the **aif360** R package in your machine

Or you can use **Docker** for example and install the package

Using Rstudio with Docker

1) Install docker: https://docs.docker.com/get-docker/

2) Go to terminal and run:

```
docker run -e PASSWORD=yourpassword --rm -p 8787:8787 rocker/rstudio
```

3) Open your browser and type: localhost:8787

Username: rstudio
Password: (the one you defined above)

More info: www.rocker-project.org/

Live Demo

```r
## 3) Calculate the mean  difference
metric_train <- binary_label_dataset_metric(data_aif_train,
                                            privileged_groups = privileged_groups,
                                            unprivileged_groups = unprivileged_groups)
metric_train$mean_difference()
# [1] -0.1932321
# The difference between the proportion of positive outcomes for the unprivileged vs
# the  privileged group
# P(Y =1| D =unprivileged) - P (Y =1| D =privileged)



### 4) Apply Adversarial debiasing is an in-processing technique that learns a classifier
## to maximize prediction accuracy and simultaneously reduce an adversary's ability to determine
### the protected attribute from the predictions
sess <- tf$compat$v1$Session()

debiased_model <- adversarial_debiasing(privileged_groups = privileged_groups,
                                        unprivileged_groups = unprivileged_groups,
                                        scope_name = 'debiased_classifier',
                                        debias = TRUE,
                                        sess = sess)

debiased_model$fit(data_aif_train)
# predictions
data_aif_train_debiasing <- debiased_model$predict(data_aif_train)

# Right now we are just caring about fairness
metric_preds <- binary_label_dataset_metric(data_aif_train_debiasing,
                                            privileged_groups = privileged_groups,
                                            unprivileged_groups = unprivileged_groups)

metric_preds$mean_difference()
# [1] -0.08583602 after
# [1] -0.1932321 before
```

adult_dataset.R

Source on Save

```r
### Load the library
library(aif360)
load_aif360_lib()

### Load the data
original_data <- readr::read_csv(
  "https://www.dropbox.com/s/ga8tr1glji7nrgk/adult_data_preprocessed.csv?dl=1"
  )
original_data <- original_data[, -1]
head(original_data)
str(original_data)

# Predict whether income exceeds $50K/yr based on census data.
# Variables:
## sex: 1 male, 0 female
## income binary: 1 > 50k, 0 <= 50k

privileged_groups <- list('sex', 1)
unprivileged_groups <- list('sex', 0)

### 1) Convert the dataframe into the aif360 format ----------------------------
data_aif <- aif_dataset(data_path = original_data,
                        favor_label = 1,
                        unfavor_label = 0,
                        privileged_protected_attribute = 1,
                        unprivileged_protected_attribute = 0,
                        target_column = "Income Binary",
                        protected_attribute = "sex")

### 2) Let's  split in train and test ------------------------------------------
# train should be 70%
# test should be 30%
set.seed(1234)
data_aif_split <- data_aif$split(num_or_size_splits = list(0.70))
data_aif_train <- data_aif_split[[1]]
data_aif_test <- data_aif_split[[2]]
```

A Better Life With Open Source.

OPENUP SUMMIT 2020

Thank You!

linkedin.com/in/staceyronaghan

linkedin.com/in/gabrieladequeiroz

linkedin.com/in/saishruthi-swaminathan

slides: bit.ly/openup-summit