Tribuo

an introduction to a Java ML Library

Mani Sarkar
@theNeomatrix369

Use Hashtag:
#makeitconf2020

Get the slide deck, it is live at

https://bit.ly/intro-to-tribuo-slides

Download the PDF for clickable links in the slides

About me

Mani Sarkar

More about me

Freelance Software, Data, ML Engineer

Java / JVM

Cloud / Infra / DevOps

Polyglot developer

Code quality, testing, performance, DevOps, deep affinity for AI/ML/DL/NLP, NN...

LJC, Devoxx, developer communities

Strengthening teams and helping them accelerate

JCP member, F/OSS projects: @adoptopenjdk @graalvm @truffleruby

Java Champion, Oracle Groundbreaker Ambassador,
Software Crafter, Blogger, Speaker

Disclaimer

- Sharing my ideas, but ***YMMV***
- Possibly missed one or more things or made mistakes, *I ask for forgiveness*
- Not clear for some, *my apologies*
- Lots of info and resources, but we *skipped* a lot
- Lots of takeaway, please go home and do some more research
- Please contribute and share, please tweet!

Citation

The respective authors and creators are, and remain the true owners of the images and other artifacts used in this presentation.

Thank you for your creations!

Agenda

Introduction	Get Started Learn Features
Demo: walk-thru	Resources Summary Q&A

Thank you!

Urban, his team and *YOU*!

And organisers at Make IT WEEK 2020

Use Hashtag:
#makeitconf2020

Celebrating 25 years of Java

Lots of apps we all want to know about...

"The 25 greatest Java apps ever written" blog post by **Alexa Weber Morales**

...for more go here

Introduction

Fun facts about the name “Tribuo”

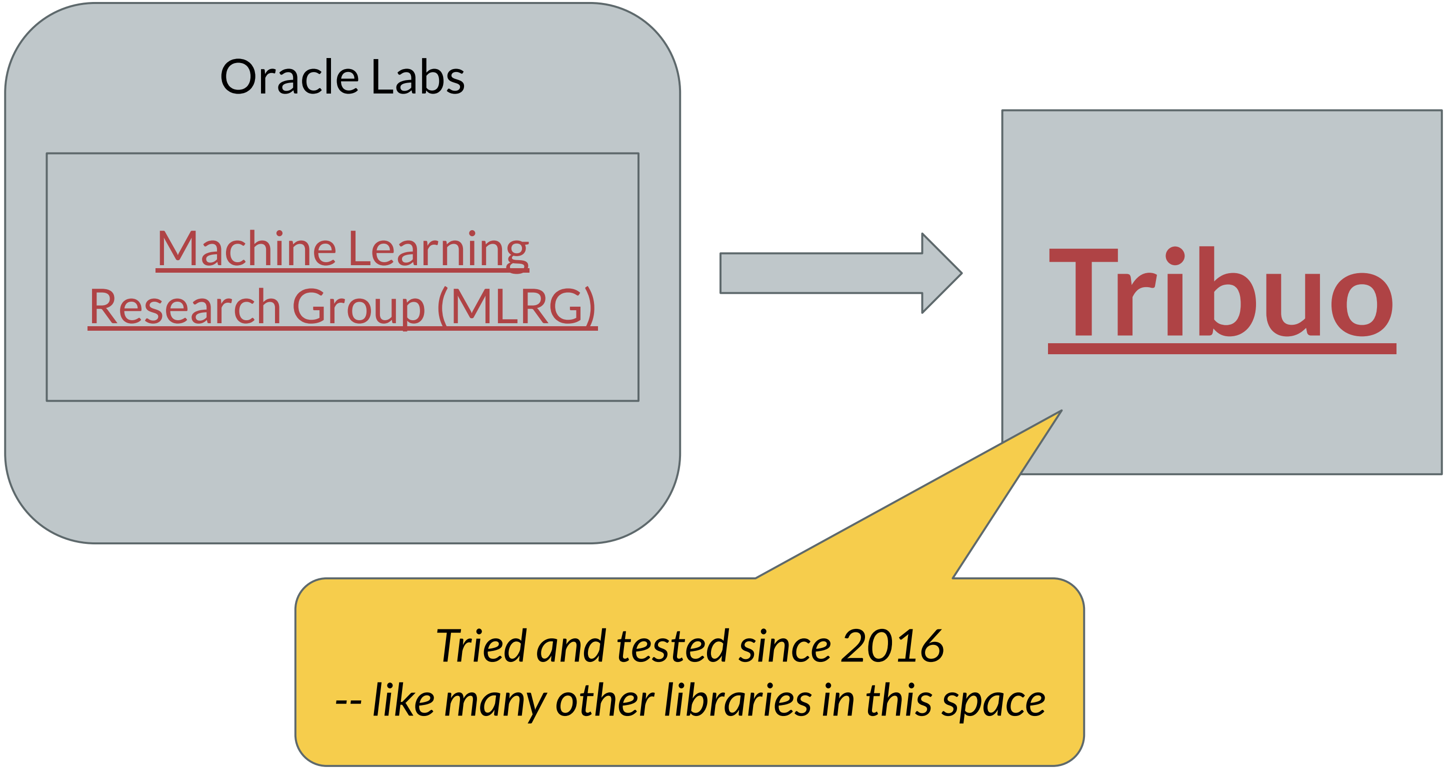
pronounced

| trib-you-oh |

in latin

| to assign, apportion |

Inception



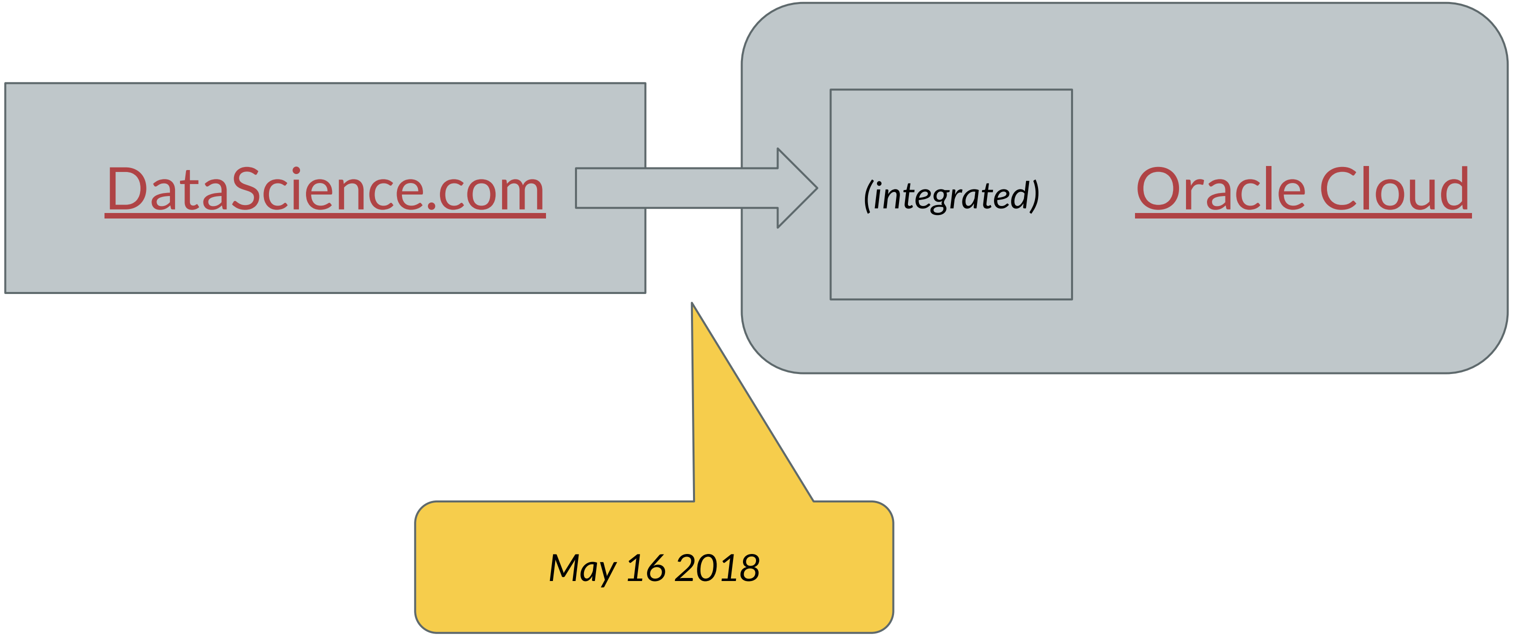
Oracle Labs

Machine Learning Research Group (MLRG)

Tribuo

Tried and tested since 2016
-- like many other libraries in this space

Another event



May 16 2018

Get Started

Fork/clone the repo

```
git clone https://github.com/oracle/tribuo.git
```

also Watch and Star the repo

Maven

```xml
<dependency>
    <groupId>org.tribuo</groupId>
    <artifactId>tribuo-all</artifactId>
    <version>4.0.1</version>
    <type>pom</type>
</dependency>
```

Gradle

```
dependencies {
    implementation ("org.tribuo:tribuo-all:4.0.1@pom") {
          // --- for Groovy uncomment below line
          // transitive = true
          // --- for Kotlin uncomment below line
          // isTransitive = true
    }
}
```

Learn

Documentation

- Features
- Architecture
- Package overview
- Security considerations
- JavaDocs

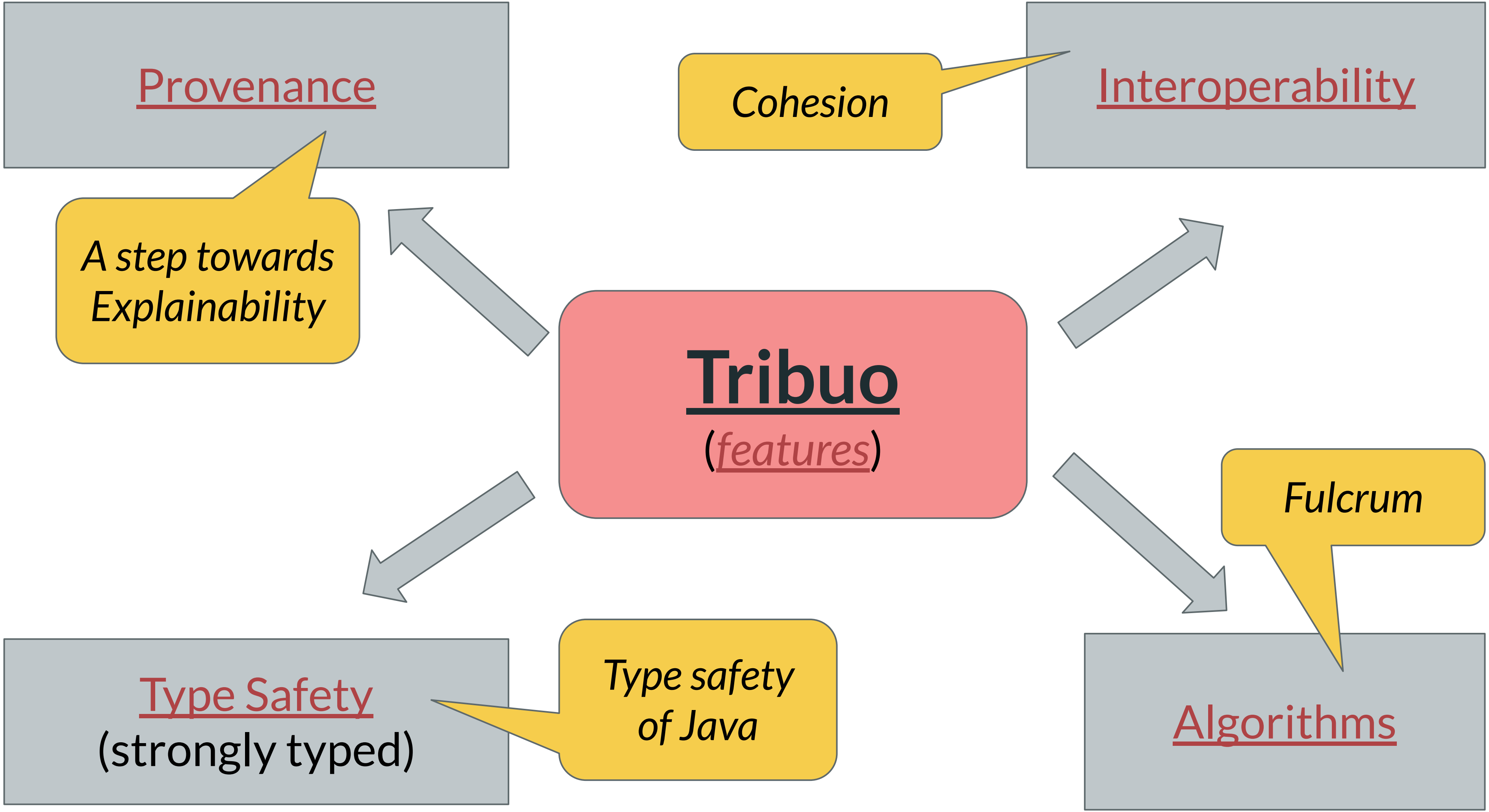
Tutorial

- Classification
- Regression
- Clustering
- Anomaly detection
- Configuration

FAQ: Frequently Asked Questions

- Project Overview
- General

Features



Tribuo

(*features*)

Provenance

A step towards Explainability

Interoperability

Cohesion

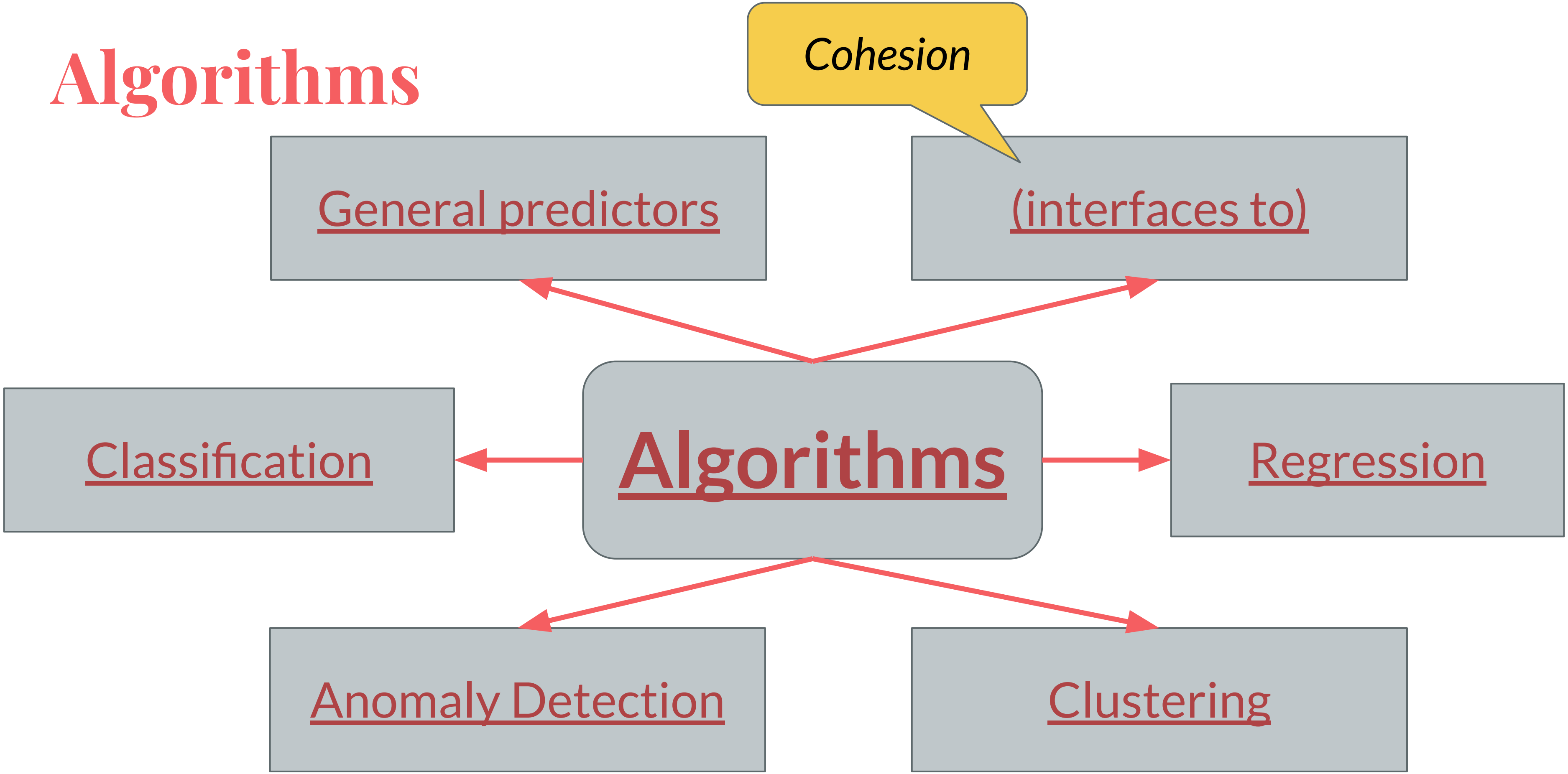
Type Safety
(strongly typed)

Type safety of Java

Algorithms

Fulcrum

Algorithms



Cohesion

General predictors

(interfaces to)

Classification

Algorithms

Regression

Anomaly Detection

Clustering

General Predictors

- Native (Tribuo)
  - Bagging
  - Random Forest
  - K-NN (k Nearest Neighbourhood)
- Neural Networks via Tensorflow

Classification

- Native (Tribuo)
  - Linear models
  - CART
  - SVM-SGD
  - Adaboost.SAMME
  - Multinomial Naive Bayes
- SVM via LibSVM
- Gradient Boosted Decision Trees via XGBoost
- Regularised Linear Models via LibLinear

Support Vector Machine

Stochastic Gradient Descent

Regression

- Native (Tribuo)
  - CART
  - Linear models
  - Lasso
  - ElasticNet
- Gradient Boosted Decision Trees via XGBoost
- SVM via LibSVM
- Regularised Linear Models via LibLinear

Clustering

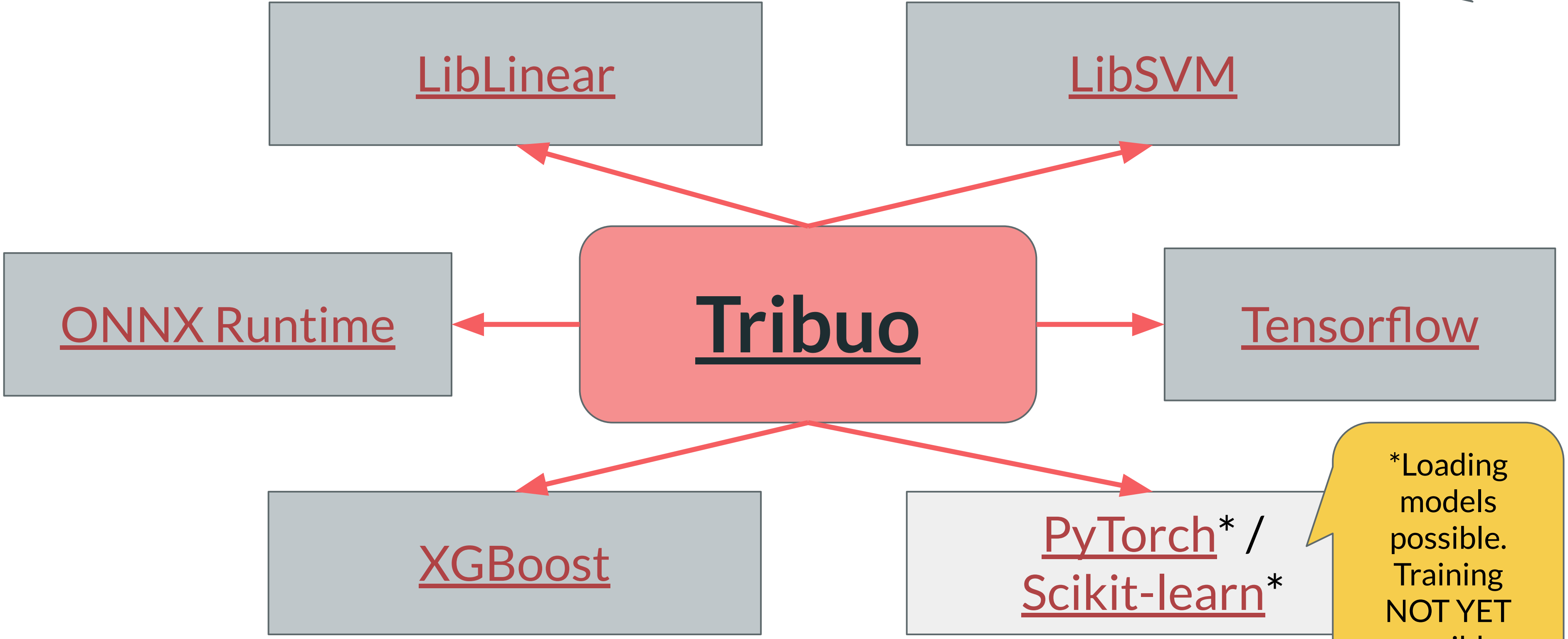
- Native (Tribuo)
    - K-Means

Anomaly Detection

- One-class SVM via LibSVM

Interfaces to

Interface your own library to Tribuo



Tribuo

LibLinear

LibSVM

ONNX Runtime

Tensorflow

XGBoost

PyTorch* / Scikit-learn*

*Loading models possible. Training NOT YET possible.

Demo: walk-thru

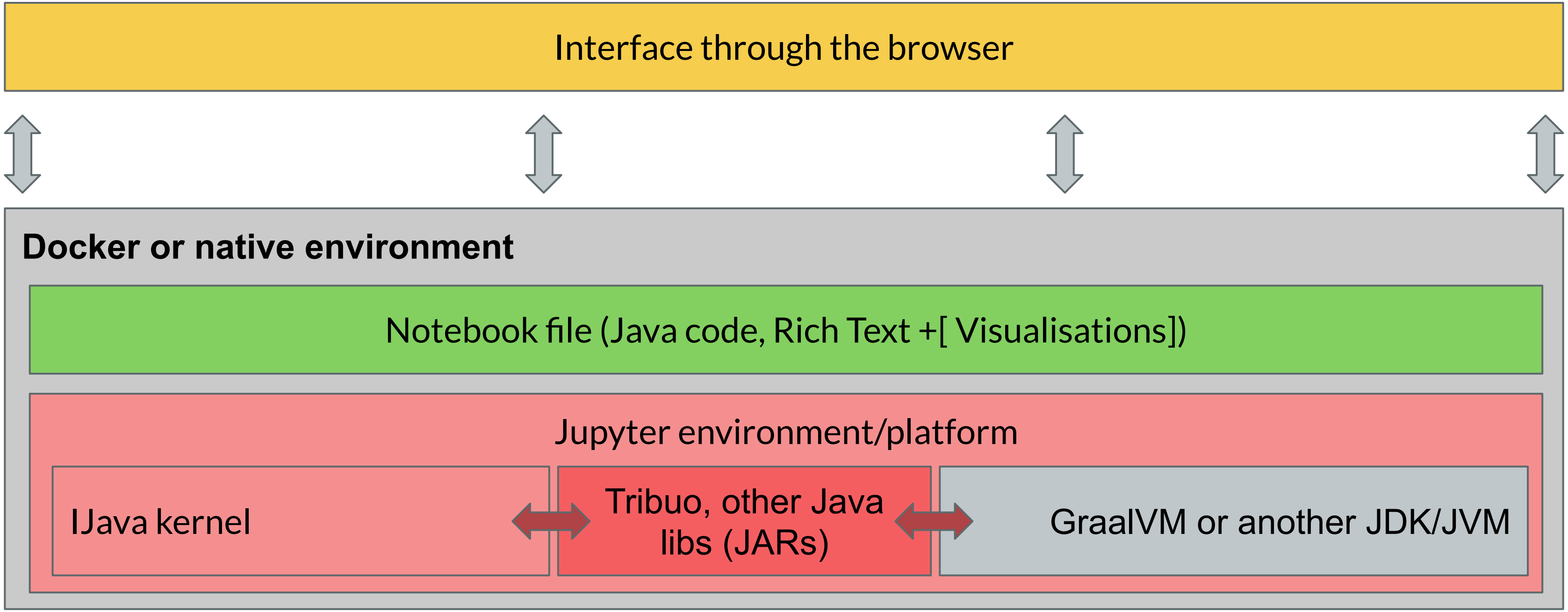
About the demo

- **Code** on **GitHub (example):**
  https://github.com/neomatrix369/awesome-ai-ml-dl/tree/master/examples/tribuo

  (follow steps in the README)

- **Classification notebook** on **GitHub:**
  https://github.com/oracle/tribuo/blob/main/tutorials/irises-tribuo-v4.ipynb
- Other **Notebooks** on **GitHub:**
  https://github.com/oracle/tribuo/blob/main/tutorials/
- **Tribuo docker image:**
  https://hub.docker.com/r/neomatrix369/tribuo

A bit more about the demo environment



Interface through the browser

Docker or native environment

Notebook file (Java code, Rich Text +[Visualisations])

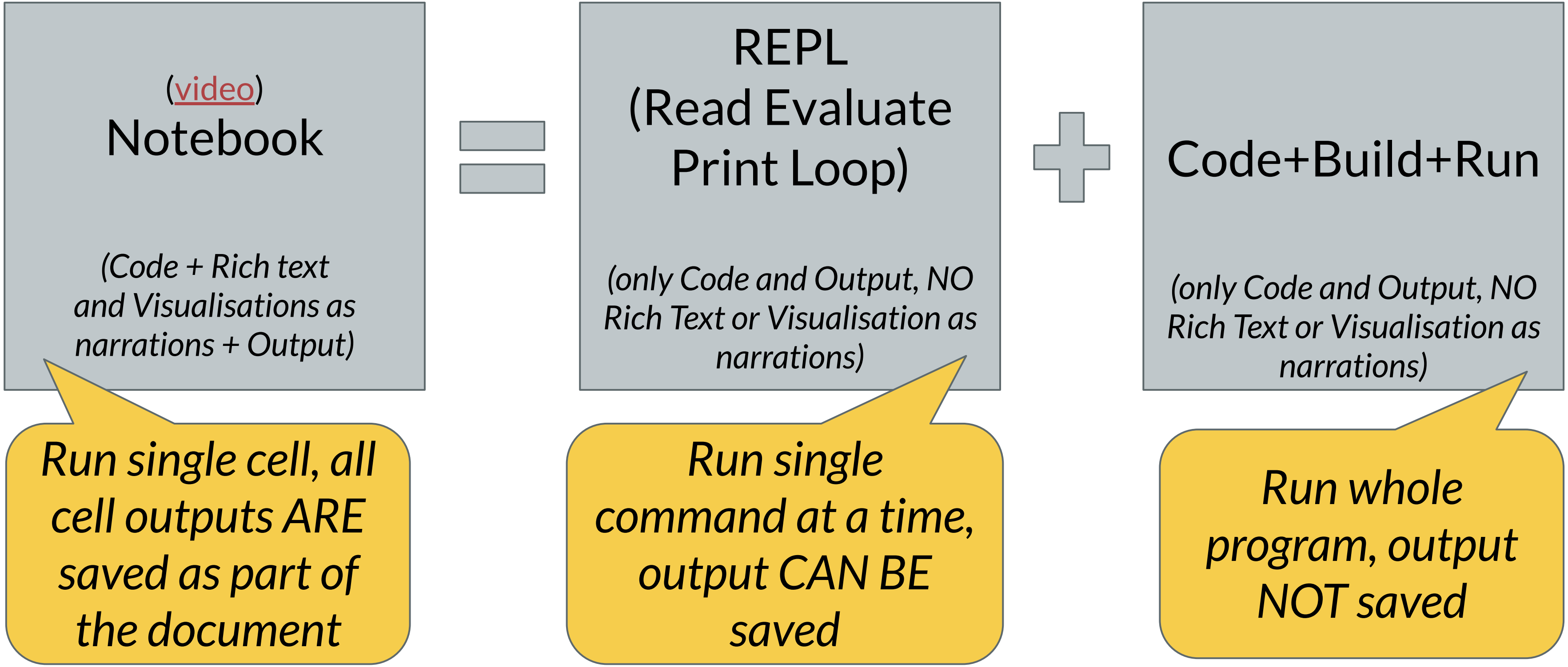
Jupyter environment/platform

IJava kernel

Tribuo, other Java libs (JARs)

GraalVM or another JDK/JVM

What is a “notebook”? Best of both worlds!



(video)
Notebook

(Code + Rich text and Visualisations as narrations + Output)

=

REPL
(Read Evaluate Print Loop)

(only Code and Output, NO Rich Text or Visualisation as narrations)

+

Code+Build+Run

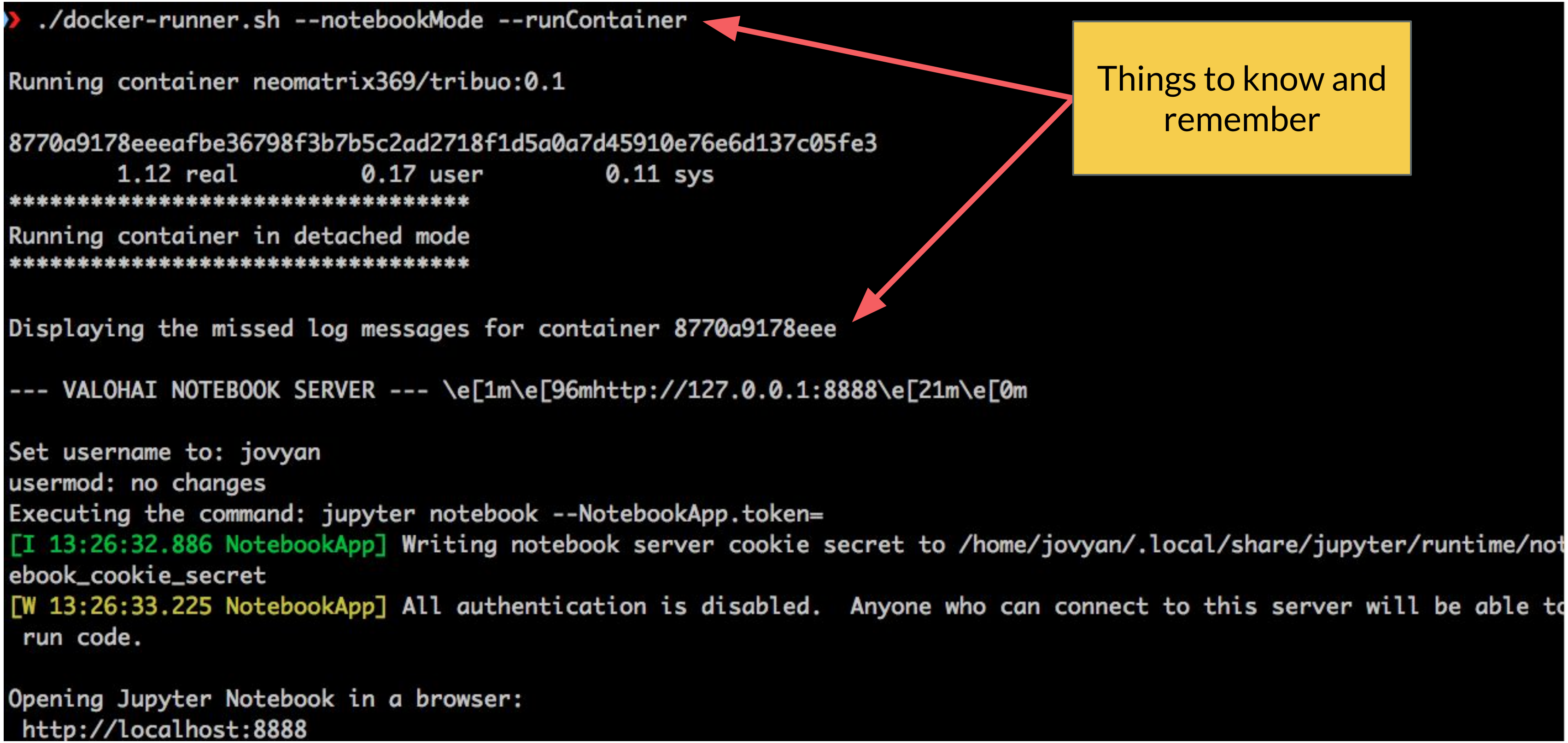
(only Code and Output, NO Rich Text or Visualisation as narrations)

Run single cell, all cell outputs ARE saved as part of the document

Run single command at a time, output CAN BE saved

Run whole program, output NOT saved

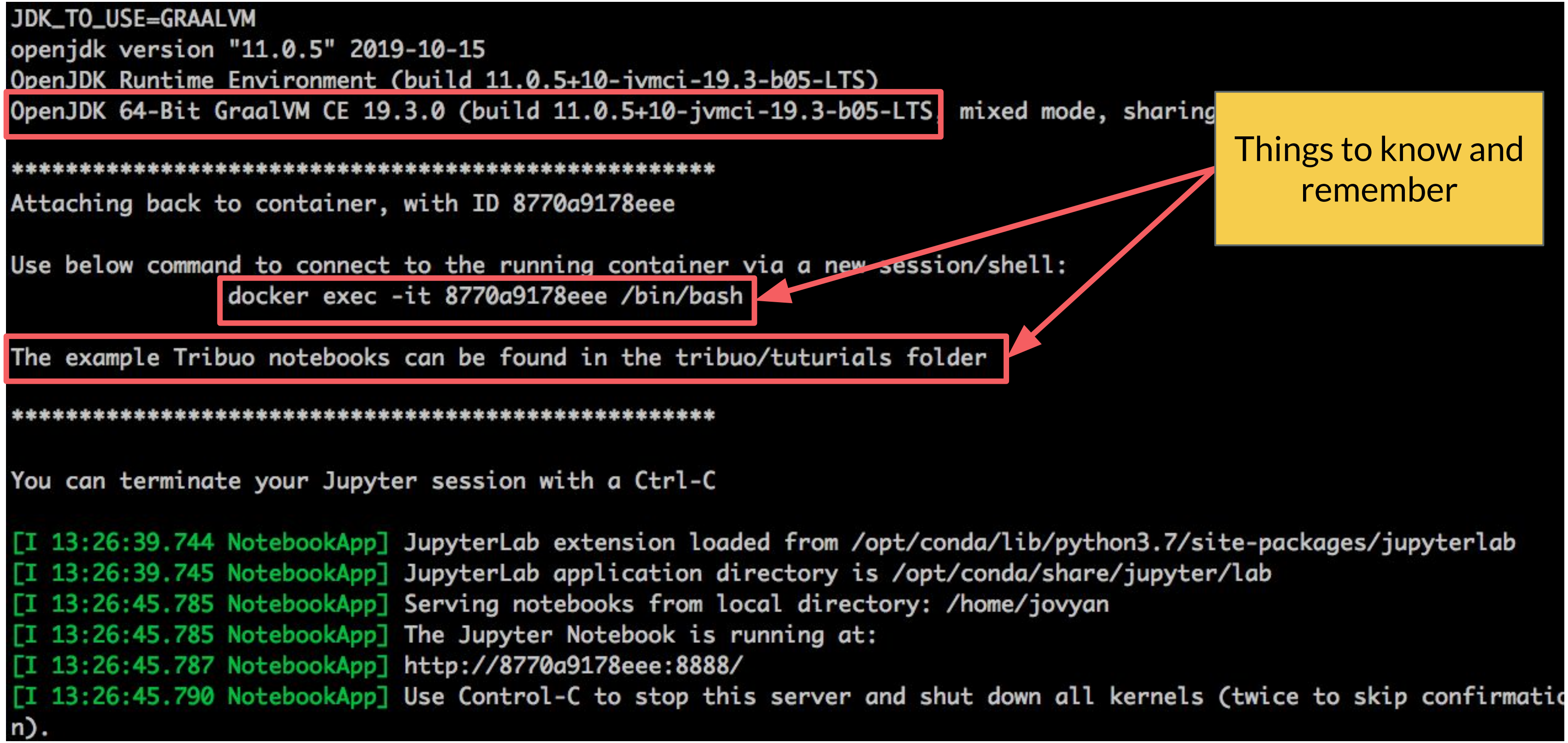
Running the IJava Jupyter notebook



```
❯ ./docker-runner.sh --notebookMode --runContainer

Running container neomatrix369/tribuo:0.1

8770a9178eeeafbe36798f3b7b5c2ad2718f1d5a0a7d45910e76e6d137c05fe3
        1.12 real         0.17 user         0.11 sys
**********************************

Running container in detached mode
**********************************


Displaying the missed log messages for container 8770a9178eee

--- VALOHAI NOTEBOOK SERVER --- \e[1m\e[96mhttp://127.0.0.1:8888\e[21m\e[0m

Set username to: jovyan
usermod: no changes
Executing the command: jupyter notebook --NotebookApp.token=
[I 13:26:32.886 NotebookApp] Writing notebook server cookie secret to /home/jovyan/.local/share/jupyter/runtime/not
ebook_cookie_secret
[W 13:26:33.225 NotebookApp] All authentication is disabled.  Anyone who can connect to this server will be able to
 run code.

Opening Jupyter Notebook in a browser:
 http://localhost:8888
```

Things to know and remember

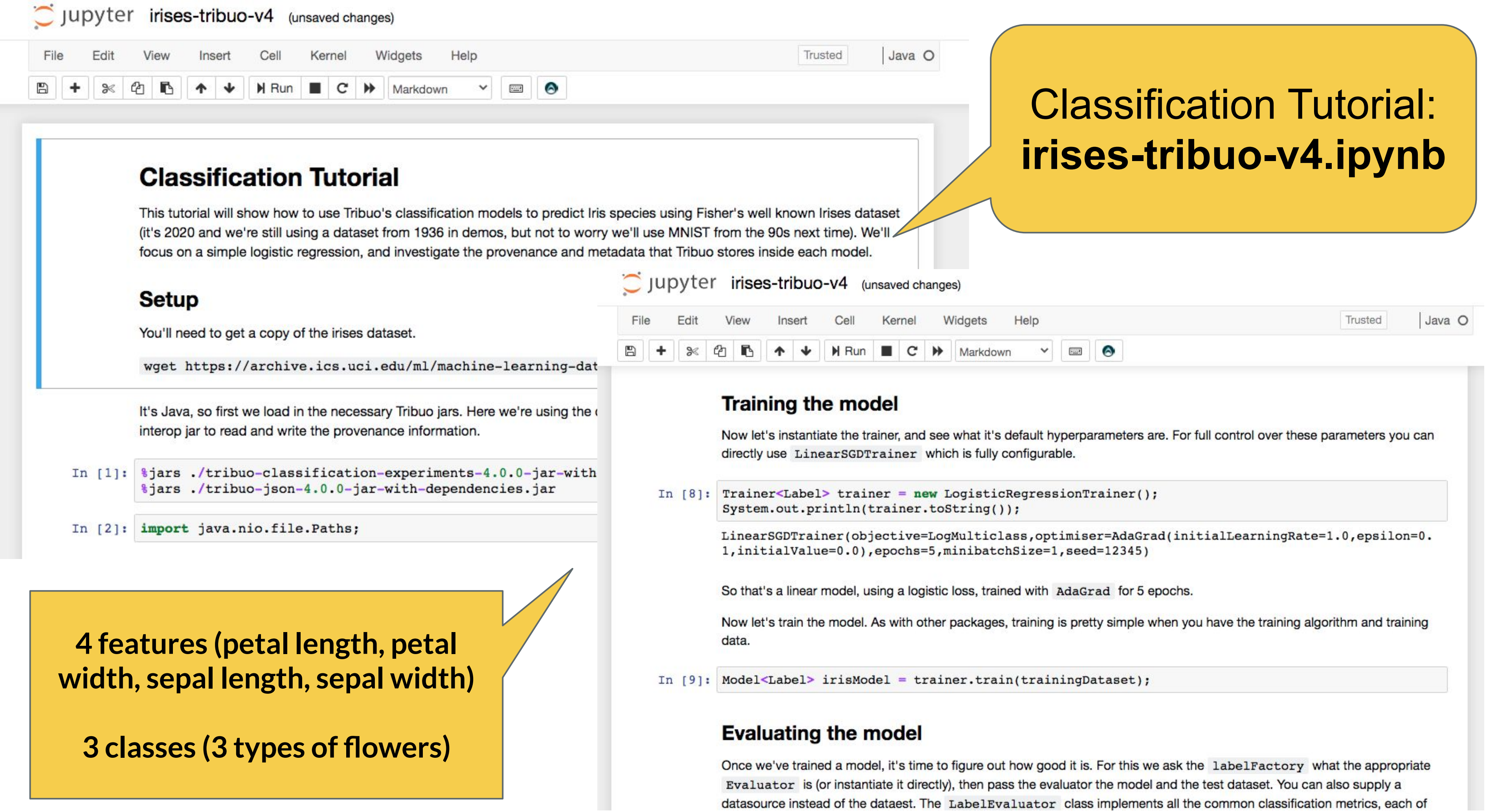
Running the IJava Jupyter notebook



```
JDK_TO_USE=GRAALVM
openjdk version "11.0.5" 2019-10-15
OpenJDK Runtime Environment (build 11.0.5+10-jvmci-19.3-b05-LTS)
OpenJDK 64-Bit GraalVM CE 19.3.0 (build 11.0.5+10-jvmci-19.3-b05-LTS mixed mode, sharing

****************************************************
Attaching back to container, with ID 8770a9178eee

Use below command to connect to the running container via a new session/shell:
               docker exec -it 8770a9178eee /bin/bash

The example Tribuo notebooks can be found in the tribuo/tuturials folder

****************************************************


You can terminate your Jupyter session with a Ctrl-C

[I 13:26:39.744 NotebookApp] JupyterLab extension loaded from /opt/conda/lib/python3.7/site-packages/jupyterlab
[I 13:26:39.745 NotebookApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 13:26:45.785 NotebookApp] Serving notebooks from local directory: /home/jovyan
[I 13:26:45.785 NotebookApp] The Jupyter Notebook is running at:
[I 13:26:45.787 NotebookApp] http://8770a9178eee:8888/
[I 13:26:45.790 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmati
n).
```

Things to know and remember

Classification Tutorial: **irises-tribuo-v4.ipynb**


jupyter irises-tribuo-v4 (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Java
Run
Markdown

Classification Tutorial

This tutorial will show how to use Tribuo's classification models to predict Iris species using Fisher's well known Irises dataset (it's 2020 and we're still using a dataset from 1936 in demos, but not to worry we'll use MNIST from the 90s next time). We'll focus on a simple logistic regression, and investigate the provenance and metadata that Tribuo stores inside each model.

Setup

You'll need to get a copy of the irises dataset.

```
wget https://archive.ics.uci.edu/ml/machine-learning-dat
```

It's Java, so first we load in the necessary Tribuo jars. Here we're using the c
interop jar to read and write the provenance information.

In [1]:
```
%jars ./tribuo-classification-experiments-4.0.0-jar-with
%jars ./tribuo-json-4.0.0-jar-with-dependencies.jar
```

In [2]:
```
import java.nio.file.Paths;
```

Training the model

Now let's instantiate the trainer, and see what it's default hyperparameters are. For full control over these parameters you can directly use `LinearSGDTrainer` which is fully configurable.

In [8]:
```
Trainer<Label> trainer = new LogisticRegressionTrainer();
System.out.println(trainer.toString());
```

```
LinearSGDTrainer(objective=LogMulticlass,optimiser=AdaGrad(initialLearningRate=1.0,epsilon=0.
1,initialValue=0.0),epochs=5,minibatchSize=1,seed=12345)
```

So that's a linear model, using a logistic loss, trained with `AdaGrad` for 5 epochs.

Now let's train the model. As with other packages, training is pretty simple when you have the training algorithm and training data.

In [9]:
```
Model<Label> irisModel = trainer.train(trainingDataset);
```

Evaluating the model

Once we've trained a model, it's time to figure out how good it is. For this we ask the `labelFactory` what the appropriate `Evaluator` is (or instantiate it directly), then pass the evaluator the model and the test dataset. You can also supply a datasource instead of the dataest. The `LabelEvaluator` class implements all the common classification metrics, each of

4 features (petal length, petal width, sepal length, sepal width)

3 classes (3 types of flowers)

(Demo)

```
var evaluator = new LabelEvaluator();
var evaluation = evaluator.evaluate(irisModel,
testingDataset);
System.out.println(evaluation.toString());
```

Classification metrics

n = number of observations per class | **tp** = True Positive | **fn** = False Negative | **fp** = False Positive

```
Class                       n          tp          fn          fp    recall      prec        f1
Iris-versicolor            16          15           1           0     0.938     1.000     0.968
Iris-virginica             15          15           0           1     1.000     0.938     0.968
Iris-setosa                14          14           0           0     1.000     1.000     1.000
Total                      45          44           1           1

Accuracy                                                              0.978
Micro Average                                                         0.978     0.978     0.978
Macro Average                                                         0.979     0.979     0.978
Balanced Error Rate                                                   0.021
```

Accuracy scores

For more details

```
var evaluator = new LabelEvaluator();
var evaluation = evaluator.evaluate(irisModel,
testingDataset);
System.out.println(evaluation.toString());
```

Classification metrics

n = number of observations per class | **tp** = True Positive | **fn** = False Negative | **fp** = False Positive

```
Class                          n          tp          fn          fp    recall        prec          f1
Iris-versicolor               16          15           1           0     0.938       1.000       0.968
Iris-virginica                15          15           0           1     1.000       0.938       0.968
Iris-setosa                   14          14           0           0     1.000       1.000       1.000
Total                         45          44           1           1

Accuracy                                                                 0.978
Micro Average                                                            0.978       0.978       0.978
Macro Average                                                            0.979       0.979       0.978
Balanced Error Rate                                                      0.021
```

recall = tp / (tp + fn)

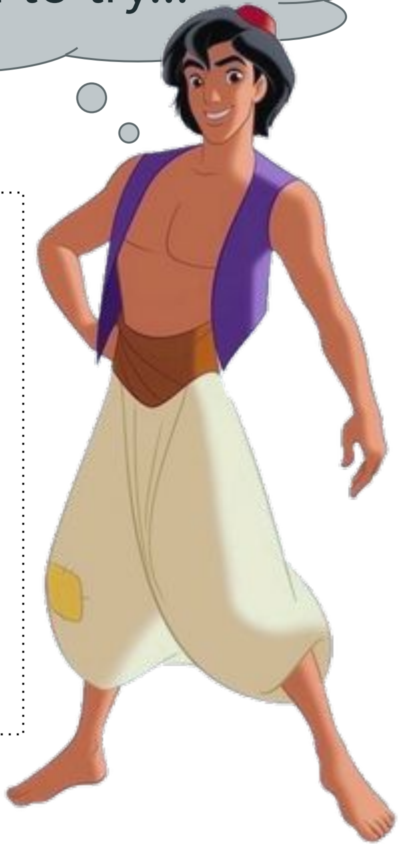
prec (precision) = tp / (tp + fp)

f1 (F1 Score) = 2 * ((precision / recall) / (precision + recall))

For more details

As Java CLI app

For you to try...



Go to this link:

Tribuo Classification example: as a Java CLI app

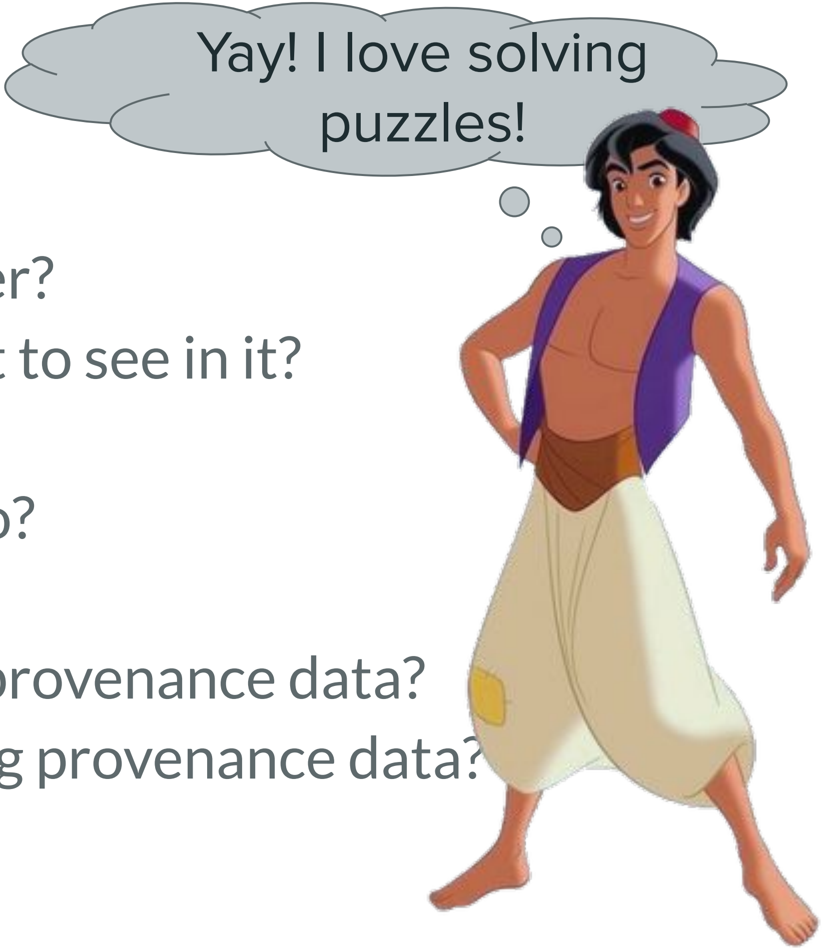
And please perform the steps mentioned there.

You should see an output like **this**.

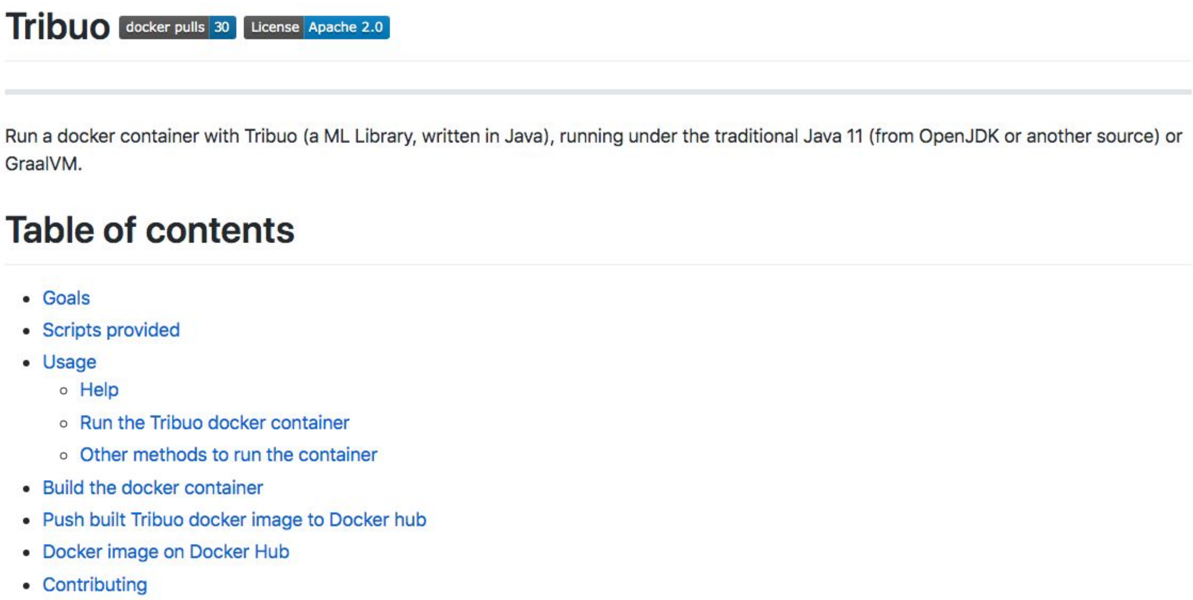
- *Build and run JAR from CLI*
- *Build and run GraalVM native-image from CLI*

Puzzles: Tribuo

- What can we do to make it better?
- What more features do we want to see in it?
- Any other ideas come to mind?
- What else can we do with Tribuo?
- Ideas about integrating with …?
- Can we recreate a model using provenance data?
- Can we recreate a e2e flow using provenance data?



Tribuo example code



Tribuo docker pulls 30 License Apache 2.0

Run a docker container with Tribuo (a ML Library, written in Java), running under the traditional Java 11 (from OpenJDK or another source) or GraalVM.

Table of contents

- Goals
- Scripts provided
- Usage
  - Help
  - Run the Tribuo docker container
  - Other methods to run the container
- Build the docker container
- Push built Tribuo docker image to Docker hub
- Docker image on Docker Hub
- Contributing

https://github.com/neomatrix369/awesome-ai-ml-dl/tree/master/examples/tribuo#tribuo--

Other libraries

- There are a few AI/ML Java libraries out there
- Eyal’s recent presentation on AI/ML Java libraries
- Zoran’s blog posts on AI/ML libraries: [1] [2]
- My Previous talks on related topics
- Other AI/ML Java talks at this event

Release & License

Tribuo library

Latest Release

tag/v4.0.1

Apache 2.0

Developer and community friendly License

Community

Community

- Oracle Labs
- Oracle Labs: Machine Learning Research Group
- Discussion List: Archive | Subscribe
- Issues | Pull Requests on GitHub
- Contribution Guidelines
- Security Reporting Guidelines

Resources

Resources

- Tribuo | GitHub
- Java on Awesome AI/ML/DL
- Awesome AI/ML/DL
- Awesome Graal
- Awesome Java | Awesome JVM

See Appendix: Learning by example

- DL4J example
- NLP example
- Jupyter Notebook example
- Apache Zeppelin example
- grCuda example
- grPython examples

12+ months
worth of
coding work

Summary

- Unique approach to Machine Learning
- Produces predictions (types instead of floating-point arrays)
- Focus on provenance (tracks metadata)
    - For posterity
    - First step to explainability
    - Recreate model
    - Recreate end-to-end flow

Summary

- Focus on security unlike many other libraries and packages
- Tried and tested for a long time
- Community friendly license
- We just scraped the surface, lots of resources to learn from and neat documentation to get started with

Thank you!

Urban, his team and *YOU*!

And organisers at Make IT WEEK 2020

Use Hashtag:
#makeitconf2020

Social media links

- twitter: @theNeomatrix369
- medium: https://medium.com/@neomatrix369
- github: https://github.com/neomatrix369/
- linkedin: https://uk.linkedin.com/pub/mani-sarkar/71/a77/39b
- slideshare: https://www.slideshare.net/neomatrix369/
- youtube: My Channel | Videos playlist

Use Hashtag:
#makeitconf2020

Questions & feedback

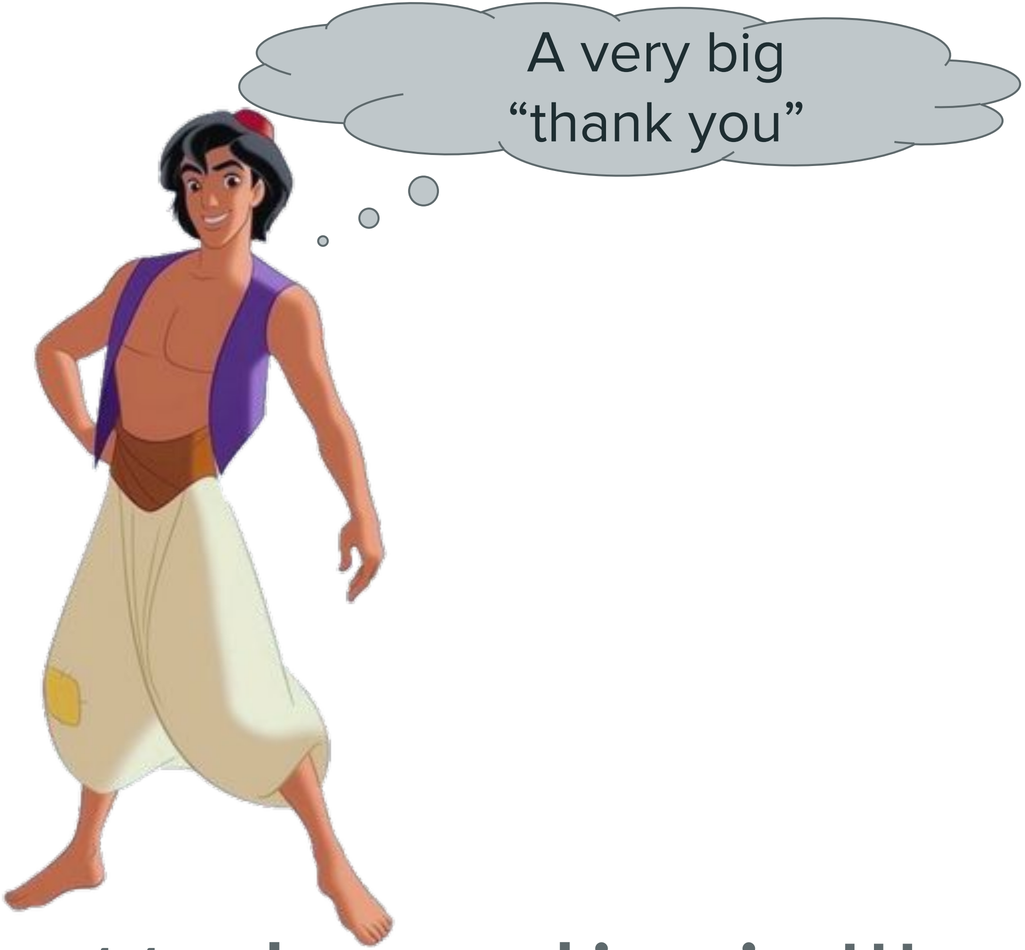
Use Hashtag:
#makeitconf2020

Please share your questions and feedback at

@theNeomatrix369

or on the video stream

Keep in mind...



It's your turn next to share and inspire!!!

Appendix

Freebies!

Get $500 worth of free cloud credits on Oracle Cloud

People doing some great work with AI & Java

Eyal Wirsansky, Zoran Sevarac, Suyash Joshi, Adam Pocock, Frank Greco, Johan Vos and many more...

(please share more names and examples with me so I can add them here)

Java and AI/ML/DL

Machine Learning Best Practices

o

Top 5 machine learning libraries for Java

o

10 Popular Java Machine Learning Tools & Libraries

o

What are machine learning libraries in Java?

Previous talks

- I recently gave a talk: **NLP Profiler**: A simple profiler, to profile textual datasets
- From backend development to machine learning
- “nn” things every Java developer should know about AI/ML/DL
- Naturally, getting productive, my journey with Grakn and Graql
- Do we know our data as well as our tools?
- Java N.n: What to know? How to learn?
- Some of my other talks a can be found here and here (and others on Slideshare)

Learning by example

DL4J example

- Github
- Blog post

NLP examples

- Example 1
  - Github
  - Blog post
- Example 2
  - Blog post
- Example 3
  - Blog post
- Better NLP

Jupyter Notebook example

- Example 1
  - Github
  - Blog: Exploring NLP concepts using Apache OpenNLP inside a Jupyter notebook
- Example 2
  - Blog post
- Example 3
  - Github
  - Blog post

Apache Zeppelin example

- Github
- Blog posts
    - Apache Zeppelin: stairway to notes* haven!
    - Running Apache Zeppelin on Oracle Cloud Infrastructure

grCuda example

- Blog posts
  - grCUDA: A Polyglot Language Binding for CUDA in GraalVM. NVIDIA Developer Blog, November 2019.
  - grCUDA: A Polyglot Language Binding. Presentation at Oracle CodeOne 2019, September 2019.
  - Simplifying GPU Access. Presentation at NVIDIA GTC 2020, March 2020
  - Optimizing Machine Learning Performance at Netsuite with GraalVM and NVIDIA GPUs
- Github

graalPython examples

- Blog posts
    - Introduction to the Python implementation for GraalVM
    - Moving from Jython to GraalVM
    - Running Python on GraalVM
- Github