“nn” things every Java Developer should know about AI/ML/DL

Mani Sarkar
@theNeomatrix369

Oracle Groundbreakers
APAC Virtual Tour 2020

Use Hashtag:
#GroundbreakersTour
#GBT2020

Get the slide deck (just now) at

https://bit.ly/gba-apac-tour-2020-slides

Download the PDF for clickable links in the slides

About me

- Member of the ***London Java Community***
- Java / JVM, Polyglot Developer
- Freelance Software, Data, ML Engineer
- Code quality, testing, performance, DevOps, deep affinity for AI/ML/DL, NN, topics
- ***Strengthening* teams and helping them *accelerate***
- Involved in F/OSS i.e. GraalVM, AdoptOpenJDK and other projects and developer communities

Mani Sarkar
@theNeomatrix369

Use Hashtag:
#GroundbreakersTour
#GBT2020

Also...

- Java Champion
- And Oracle Groundbreaker Ambassador

Click here for more info on Freebies!

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
- Lots of takeaway, please go home and do some more research
- Please contribute and share, please tweet!

Agenda

Why “nn” things?

Demo(s), maybe some code!

Timeline: how it started for me...

Insights, shares and tips...

What we won't cover...

Pros or cons of ...

What is the best library / framework to do "xyz" in?

We may name some topics but we won't go into the details of them. We name them or express an opinion so we have a starting point in our learning journey!

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

“nn” things every Java Developer should know about AI/ML/DL

Mani Sarkar
@theNeomatrix369

Use Hashtag:
#jOnConf

Gave this talk in
***July 2020** at **jOnConf 2020**,*
see slides and video

Why this talk or such talks?

They are all doing an awesome job and a service to us!

Many can show how to install or configure “xyz”!

Many have shown us how to code super fast and at a presentation!

Many have talked about frameworks, APIs, algorithms, theories, and the works!

But not many talks about how to get there, how each one of us can become better?

I could just show you some code

Share some links and videos

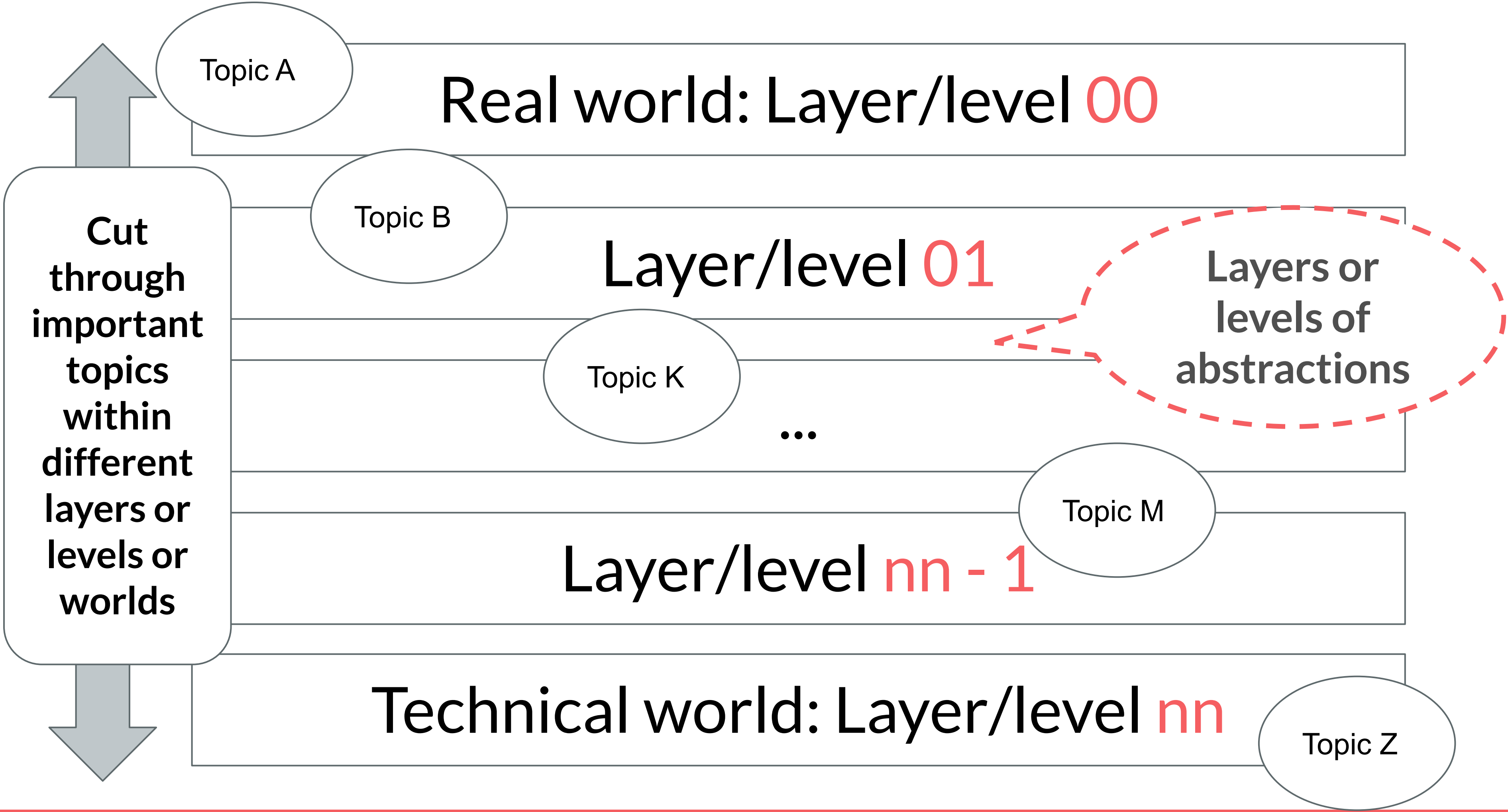
Show some demos!

And share some tips!

Instead I’ll try to share perspectives, on how things sit together in my experience!

Share approaches and how I try to go around things!

Share methods and maybe some of my work! In additional to all other useful resources!



Topic A

Real world: Layer/level 00

Topic B

Layer/level 01

Layers or levels of abstractions

Topic K

...

Topic M

Layer/level nn - 1

Technical world: Layer/level nn

Topic Z

Cut through important topics within different layers or levels or worlds

Why “nn” things?

Each time we
know “n” things...

We find that there
are “n” plus “more”
things to know

It can give rise to
something like
“imposter syndrome”
or just “overwhelm” us!

**Only way forward
is to remove such
impediments!**

What is AI / ML / DL ?

We could look for it’s meanings on Wikipedia, but that does not serve our purpose fully.

Mish-mash of many topics, techniques, methods! They can overlap yet they can be different and separate

Which kind of model to build maybe an incorrect question!

Multiple meanings, overlapping meanings, misconceptions and hype

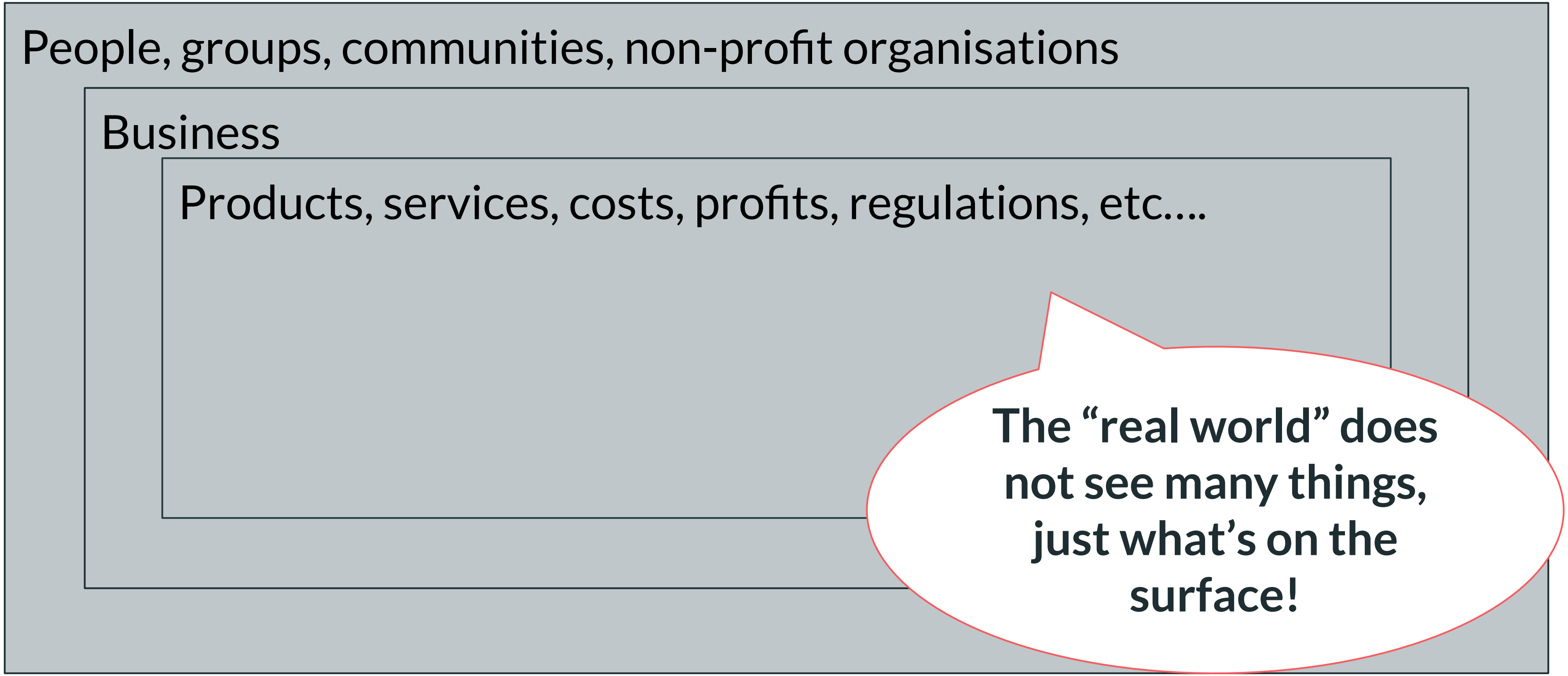
Understanding data over model building, see talks

Do your own research! Don’t just follow the hype!

Interpretable and explainable models over black box models!

Levels of understanding, abstractions, and views

Real world



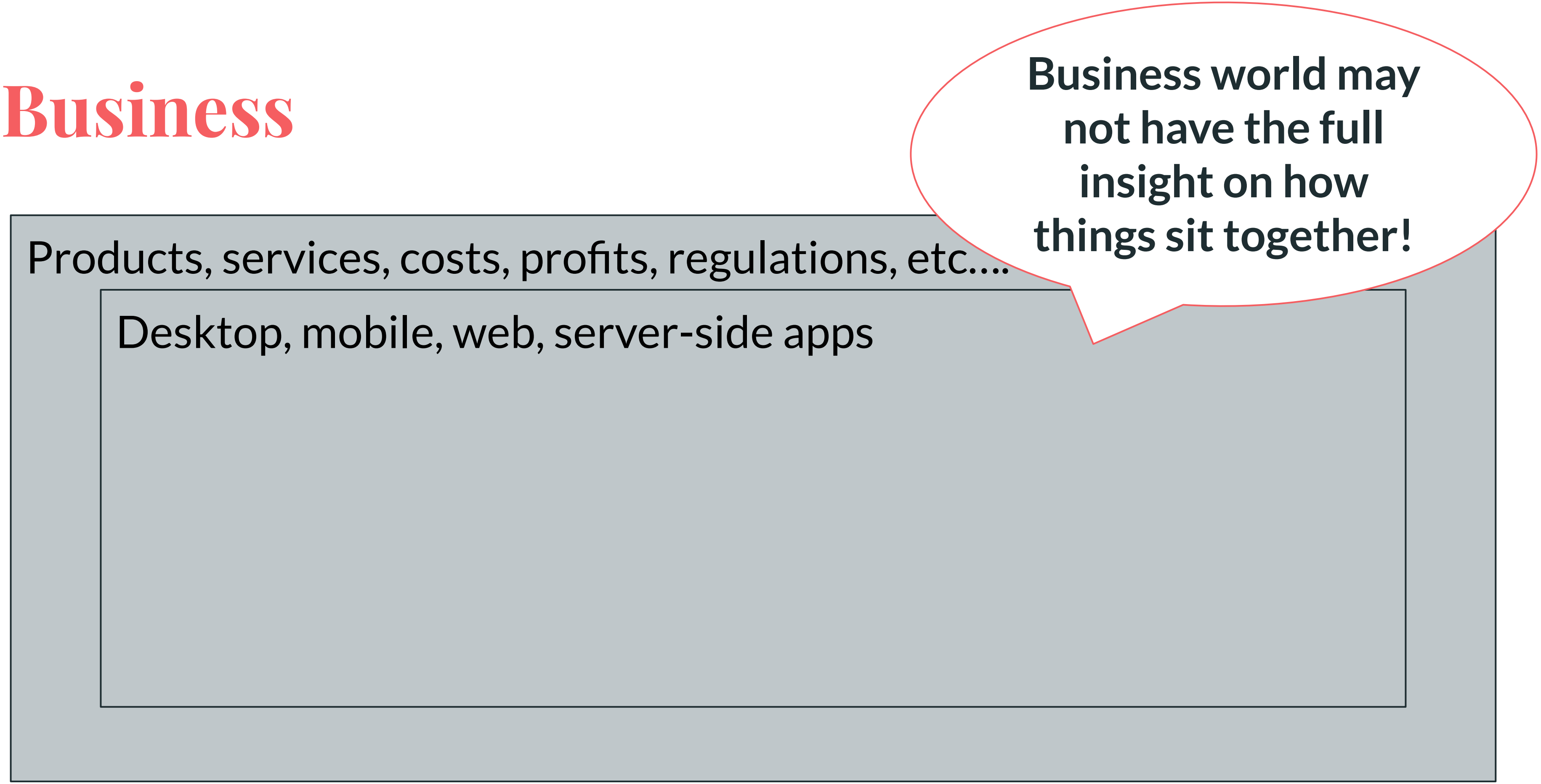
People, groups, communities, non-profit organisations

Business

Products, services, costs, profits, regulations, etc.…

The “real world” does not see many things, just what’s on the surface!

Business

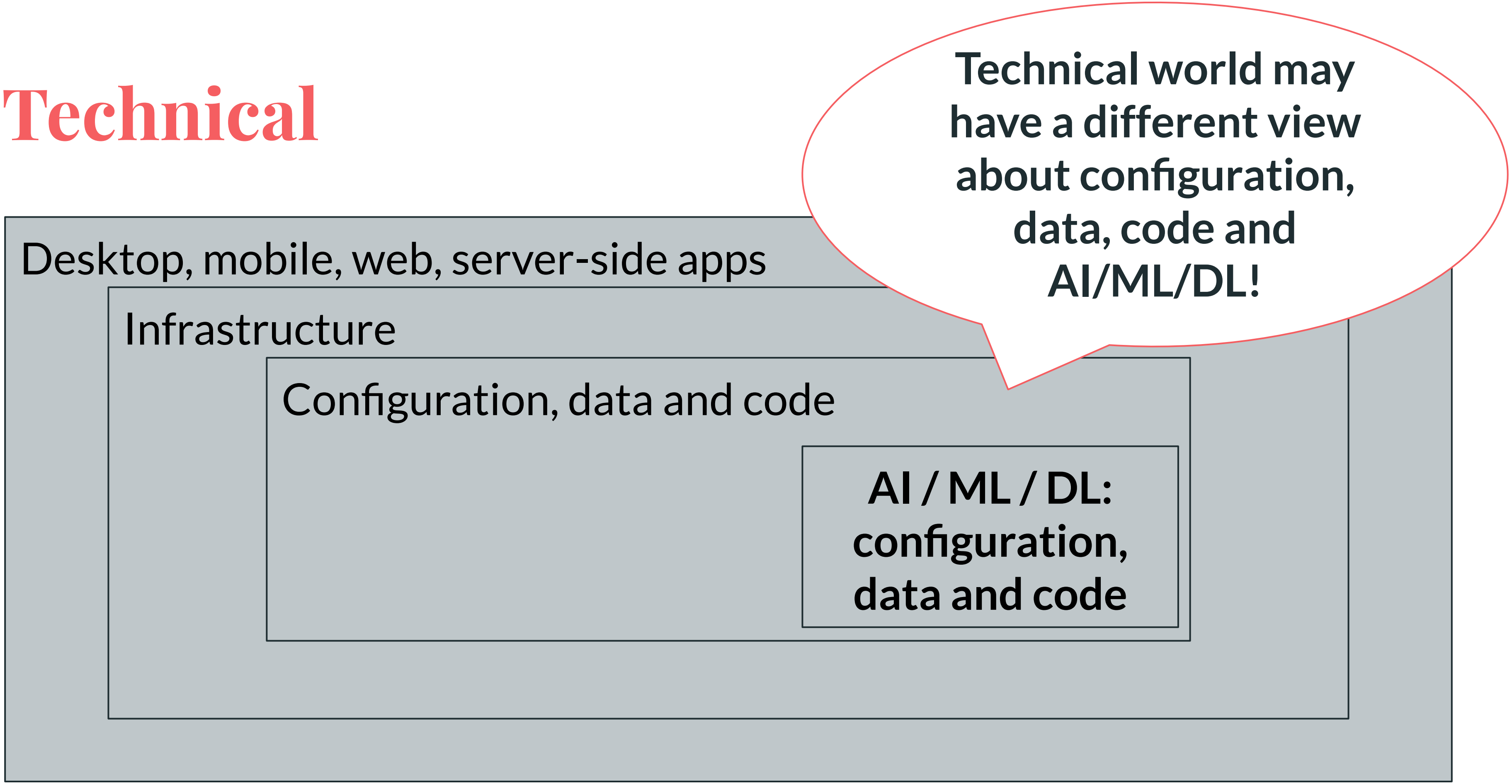


Products, services, costs, profits, regulations, etc….

Desktop, mobile, web, server-side apps

Business world may not have the full insight on how things sit together!

Technical



Desktop, mobile, web, server-side apps

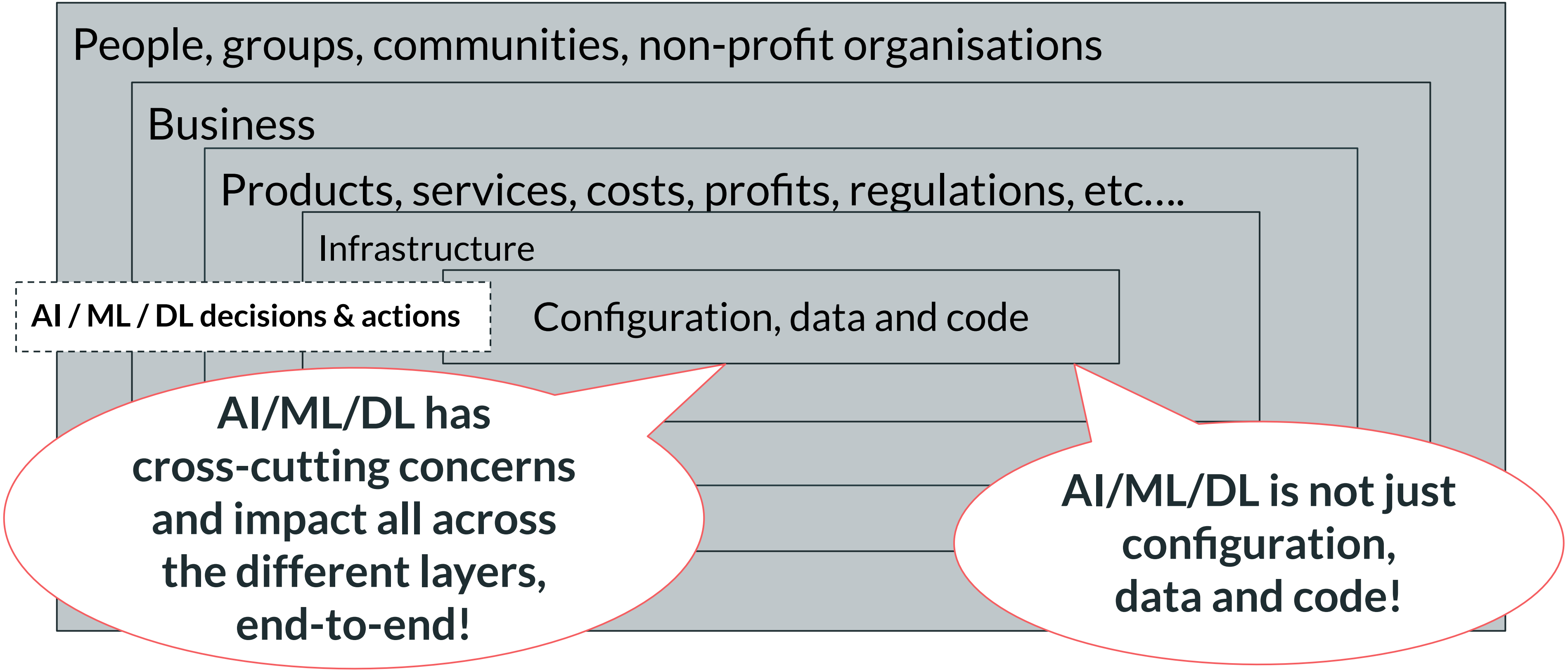
Infrastructure

Configuration, data and code

AI / ML / DL: configuration, data and code

Technical world may have a different view about configuration, data, code and AI/ML/DL!

In reality



People, groups, communities, non-profit organisations

Business

Products, services, costs, profits, regulations, etc....

Infrastructure

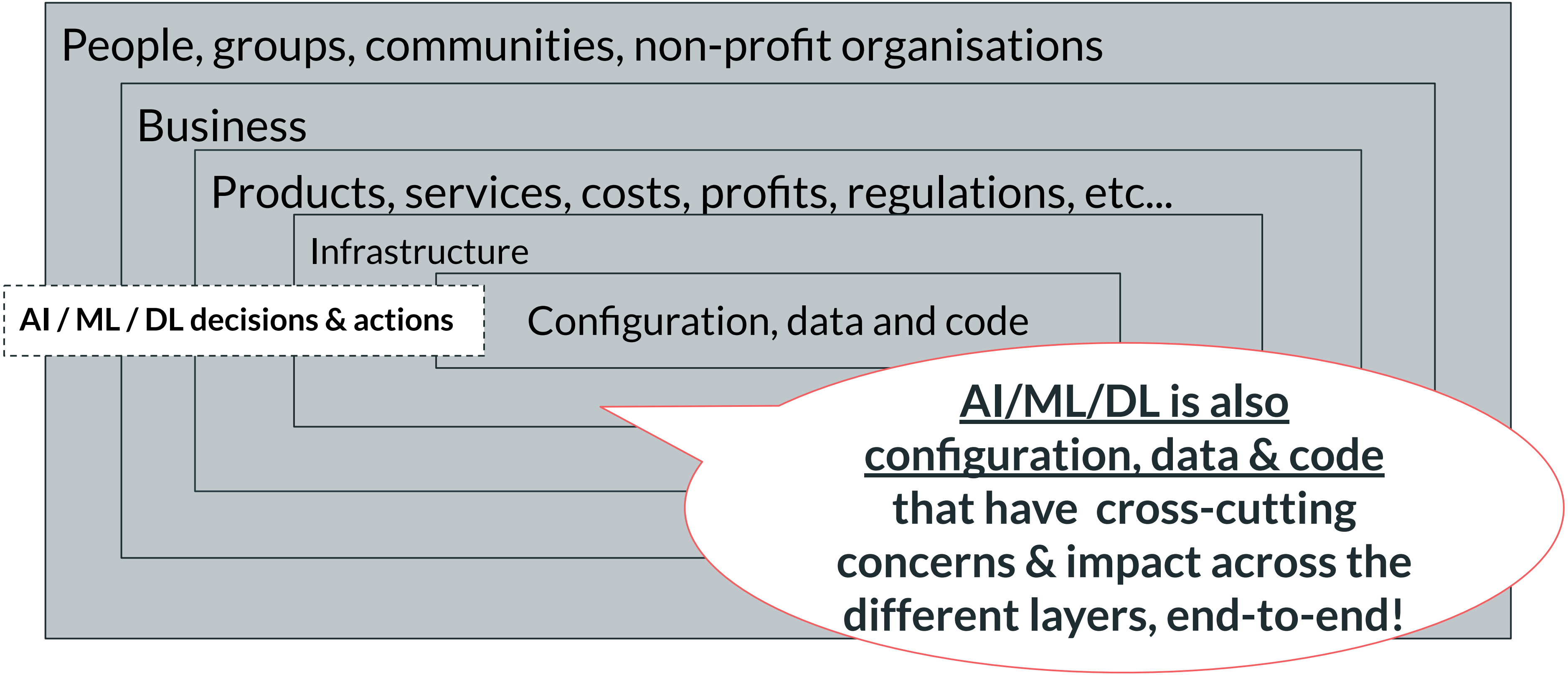
Configuration, data and code

AI / ML / DL decisions & actions

AI/ML/DL has cross-cutting concerns and impact all across the different layers, end-to-end!

AI/ML/DL is not just configuration, data and code!

In reality



People, groups, communities, non-profit organisations

Business

Products, services, costs, profits, regulations, etc...

Infrastructure

Configuration, data and code

AI / ML / DL decisions & actions

AI/ML/DL is also configuration, data & code that have cross-cutting concerns & impact across the different layers, end-to-end!

And then there are other realities!

How it all started for me?

Timeline

- January to November 2018: collecting links
- 2018 December: creation of **awesome-ai-ml-dl**
- Early 2019: Better NLP library
- 2019: collaboration with Virgilio (reviewing guides)
- May 2019: talk on data, see talks slide
- Mid 2019: another Better NLP presentation
- Rest of 2019: blogs on AI/ML/DL and example projects
- Late 2019, early 2020: online Data/ML competitions
- Mid 2020: created the NLP Profiler library
- 2020: more links added to **awesome-ai-ml-dl**, talks delivered

Thanks to Yolande Poirier for posting all those AI/ML/DL links on @java between 2016 and 2018

Thanks to the **Meet-a-Mentor** initiative by **LJC** during 2018-19, for holding the **ML Study Group**!

Thanks to **Yaz** for the Tensorflow meetups in London, UK during 2018 and 2019

And many others in this journey...

How do I learn?

Top-down, outside in learning, see Rachel Thomas’ talk (fast.ai)

Mind-maps, gathering reading lists, organising them, categorising them

Nodes of Knowledge (see Appendix, previous talks)

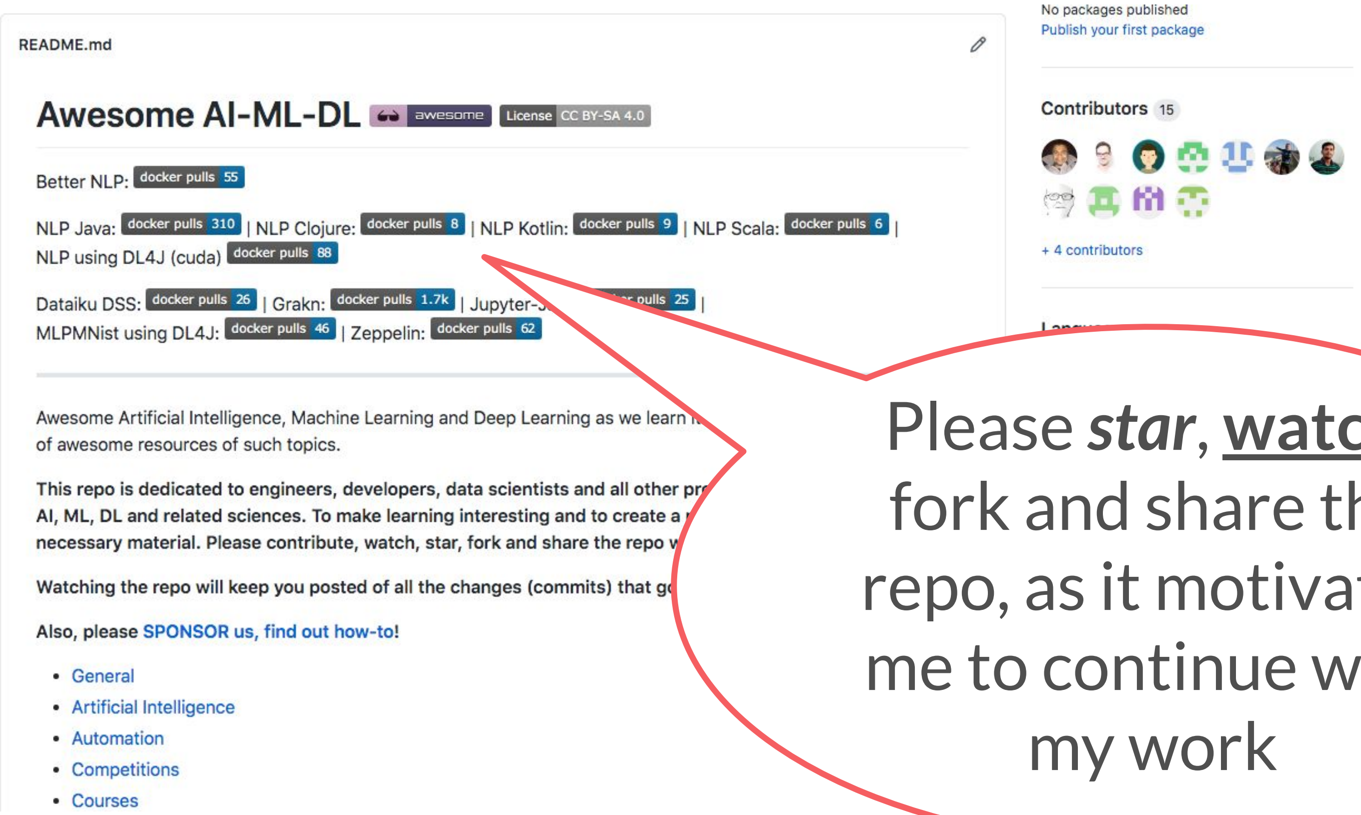
Automatic chunking, Learning to Learn course by Coursera

List building: helps know the landscape and understand the context: birds-eye-view

List building: helps prioritise, order, remove or add topics you like, or don’t like or don’t know

AI/ML/DL git repo

https://github/neomatrix369/awesome-ai-ml-dl



README.md

Awesome AI-ML-DL

Better NLP: docker pulls 55

NLP Java: docker pulls 310 | NLP Clojure: docker pulls 8 | NLP Kotlin: docker pulls 9 | NLP Scala: docker pulls 6 |
NLP using DL4J (cuda) docker pulls 88

Dataiku DSS: docker pulls 26 | Grakn: docker pulls 1.7k | Jupyter-J er pulls 25 |
MLPMNist using DL4J: docker pulls 46 | Zeppelin: docker pulls 62

Awesome Artificial Intelligence, Machine Learning and Deep Learning as we learn i
of awesome resources of such topics.

**This repo is dedicated to engineers, developers, data scientists and all other pr
AI, ML, DL and related sciences. To make learning interesting and to create a
necessary material. Please contribute, watch, star, fork and share the repo w**

Watching the repo will keep you posted of all the changes (commits) that g

Also, please SPONSOR us, find out how-to!

- General
- Artificial Intelligence
- Automation
- Competitions
- Courses

No packages published
Publish your first package

Contributors 15

+ 4 contributors

Please ***star***, **watch,**
fork and share the
repo, as it motivates
me to continue with
my work

AI/ML/DL highlights

My first post on AI/ML/DL!

Courses

ML on Code

Data Science

Mathematics, Statistics, Probability & Probabilistic programming

Neural Networks

NLP

Graphs

Genetic Algorithms

JavaScript

Natural Language Processing

Machine Learning

Competition

Cloud, DevOps, Infra

Visualisation

Time-series / anomaly detection

Data

AI/ML/DL & Java highlights

Classifiers and Decisions trees

Data Science

Deep Learning

Neural Networks

Tools, libraries & resources

Genetic Algorithms

Machine Learning

JSRs

Natural Language Processing

Clojure resources

Scala resources

Java related projects and technologies

AI/ML/DL artefacts

Cheatsheets

Notebooks

Presentations

Examples

Virgilio

Guides

Docker containers

Study Notes

Things to know list

Learning by example

See Appendix: Learning by example

- DL4J example
- NLP example
- Jupyter Notebook example
- Apache Zeppelin example
- graql-to-english, english-to-graql
- grCuda example
- grPython examples

12+ months
worth of
coding work

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)

- **Regression notebook** on **GitHub:**
  https://github.com/oracle/tribuo/blob/main/tutorials/regression-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

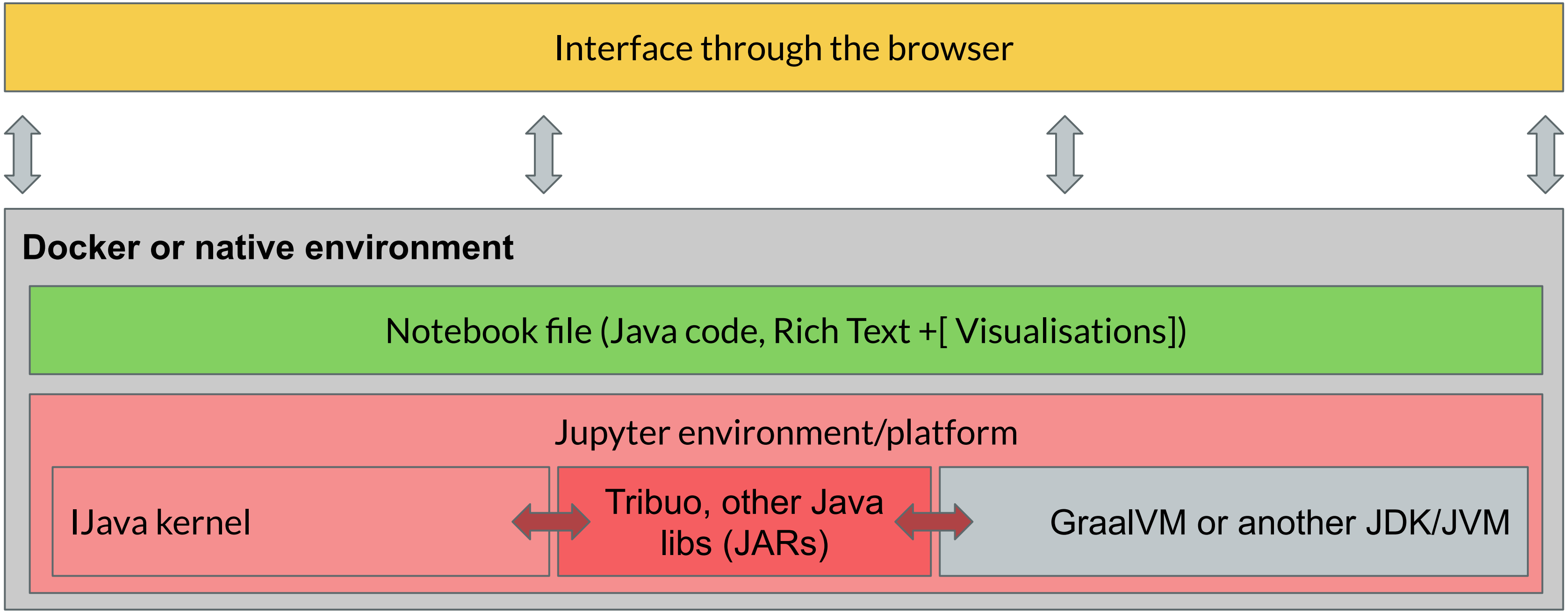
Tribuo Classification Tutorial

See my previous talk

Tribuo: an introduction to a Java ML Library

(for slides and video)

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)

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

=

REPL
(Read Evaluate Print Loop)

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

Run single command at a time, output CAN BE saved

+

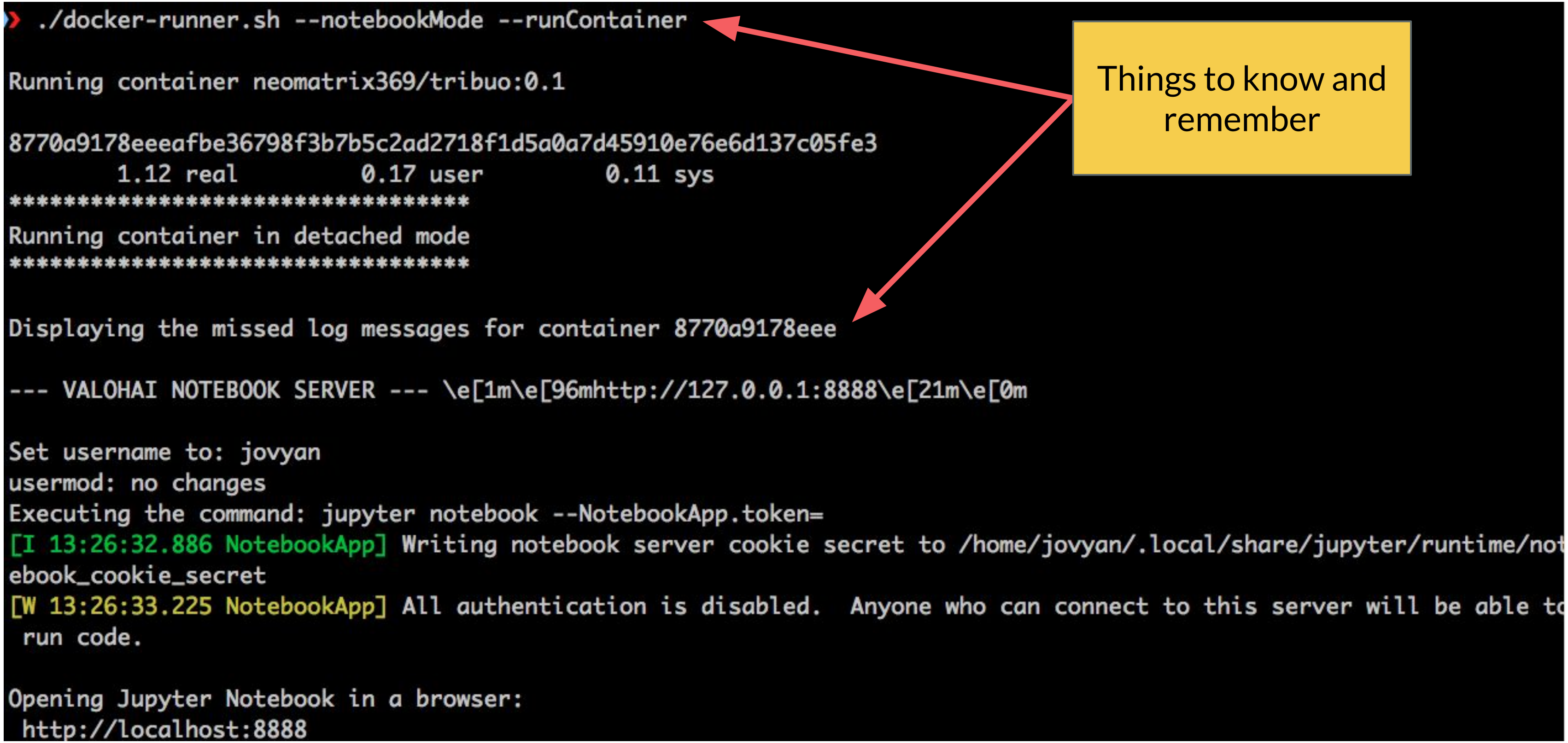
Code+Build+Run

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

Run whole program, output NOT saved

Video
(Java Jupyter Notebook)

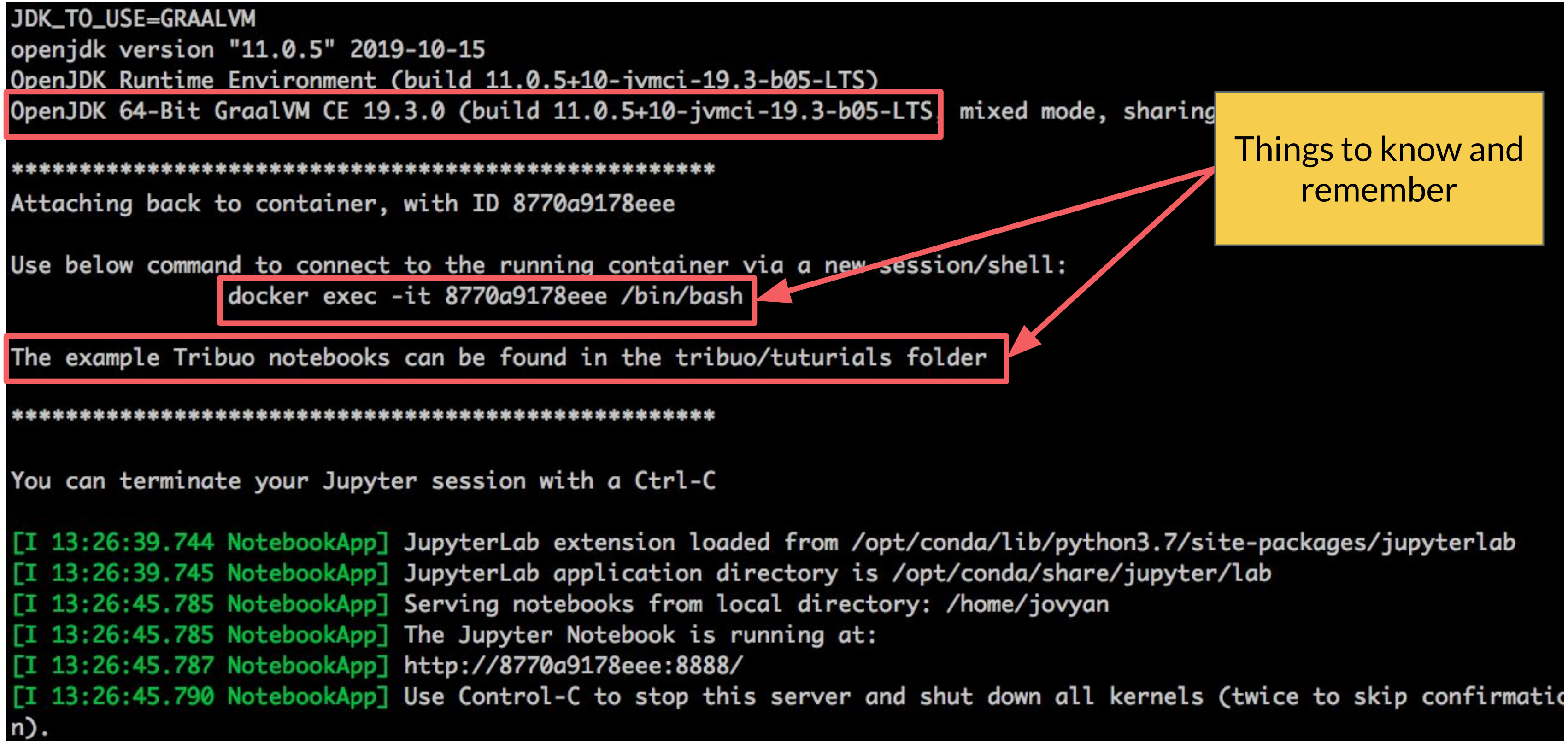
Running the IJava Jupyter notebook



```
❯ ./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

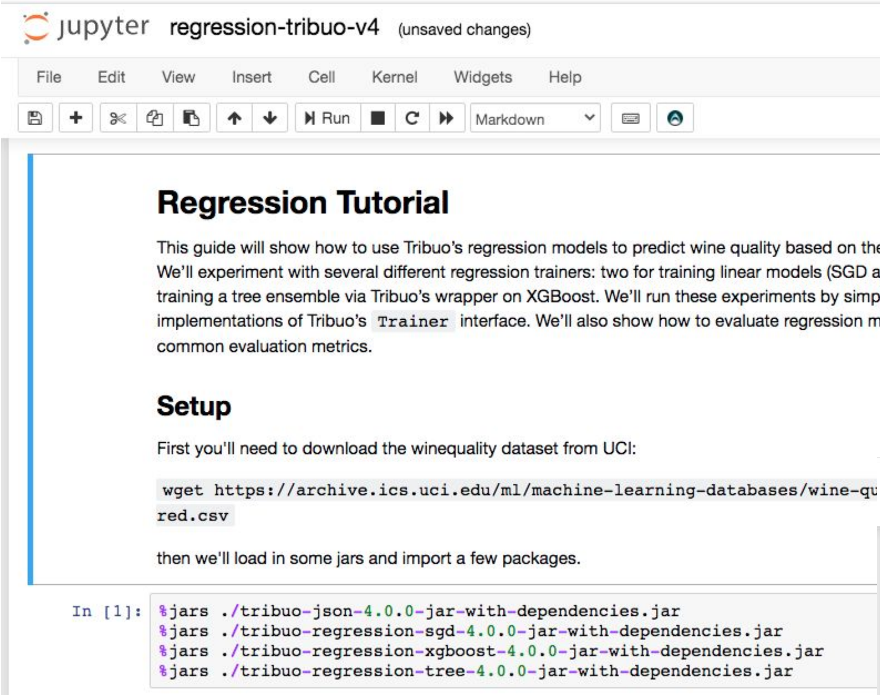
Running the IJava Jupyter notebook



```
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

Regression Tutorial:
regression-tribuo-v4.ipynb


jupyter regression-tribuo-v4 (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Run
Markdown

Regression Tutorial

This guide will show how to use Tribuo's regression models to predict wine quality based on the UCI Wine Quality data set. We'll experiment with several different regression trainers: two for training linear models (SGD and Adagrad) and one for training a tree ensemble via Tribuo's wrapper on XGBoost. We'll run these experiments by simply swapping in different implementations of Tribuo's `Trainer` interface. We'll also show how to evaluate regression models and describe some common evaluation metrics.

Setup

First you'll need to download the winequality dataset from UCI:

`wget https://archive.ics.uci.edu/ml/machine-learning-databases/wine-qu`
`red.csv`

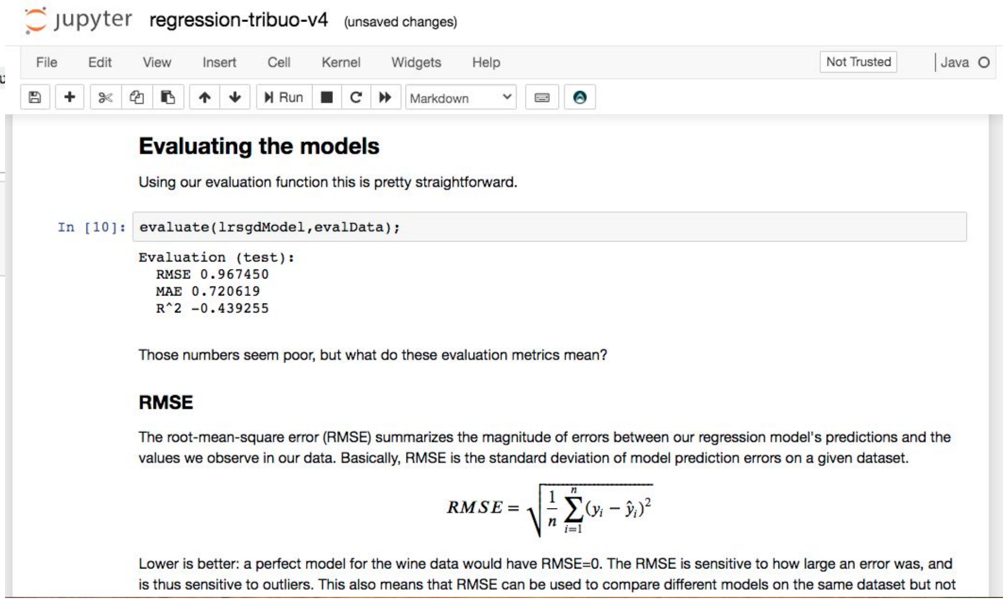
then we'll load in some jars and import a few packages.

In [1]:

```
%jars ./tribuo-json-4.0.0-jar-with-dependencies.jar
%jars ./tribuo-regression-sgd-4.0.0-jar-with-dependencies.jar
%jars ./tribuo-regression-xgboost-4.0.0-jar-with-dependencies.jar
%jars ./tribuo-regression-tree-4.0.0-jar-with-dependencies.jar
```

features: 9 to 10 characteristics of wine

target (label): continuous values (wine quality)


jupyter regression-tribuo-v4 (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Java
Run
Markdown

Evaluating the models

Using our evaluation function this is pretty straightforward.

In [10]:

```
evaluate(lrsgdModel,evalData);
```

```
Evaluation (test):
  RMSE 0.967450
  MAE 0.720619
  R^2 -0.439255
```

Those numbers seem poor, but what do these evaluation metrics mean?

RMSE

The root-mean-square error (RMSE) summarizes the magnitude of errors between our regression model's predictions and the values we observe in our data. Basically, RMSE is the standard deviation of model prediction errors on a given dataset.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Lower is better: a perfect model for the wine data would have RMSE=0. The RMSE is sensitive to how large an error was, and is thus sensitive to outliers. This also means that RMSE can be used to compare different models on the same dataset but not

(Demo)

```
var lrsgdModel = train("Linear Regression (SGD)",lrsgd,trainData);
evaluate(lrsgdModel,evalData);
```

Training and test Evaluation metrics

(test sometimes referred to as validation)

```
Evaluation (train):
  RMSE 0.979522
  MAE 0.741870
  R^2 -0.471611
```

```
Evaluation (test):
  RMSE 0.967450
  MAE 0.720619
  R^2 -0.439255
```

RMSE: root mean square error
(std. dev.)
--
MAE: mean absolute error
--
R^2: R-squared
(mean deviation, 0 and 1)

See the ***Evaluating the models*** section of the notebook for further details on each of the above metrics

Training and test Evaluation metrics

(test sometimes referred to as validation)

```
Training Linear Regression (AdaGrad) took (00:00:00:045)
Evaluation (train):
  RMSE 0.735311
  MAE 0.575096
  R^2 0.170709
Evaluation (test):
  RMSE 0.737994
  MAE 0.585709
  R^2 0.162497
```

```
Training CART took (00:00:00:076)
Evaluation (train):
  RMSE 0.545205
  MAE 0.406670
  R^2 0.544085
Evaluation (test):
  RMSE 0.657900
  MAE 0.494812
  R^2 0.334420
```

```
Training XGBoost took (00:00:00:375)
Evaluation (train):
  RMSE 0.143871
  MAE 0.097167
  R^2 0.968252
Evaluation (test):
  RMSE 0.599478
  MAE 0.426673
  R^2 0.447378
```

AdaGrad optimiser

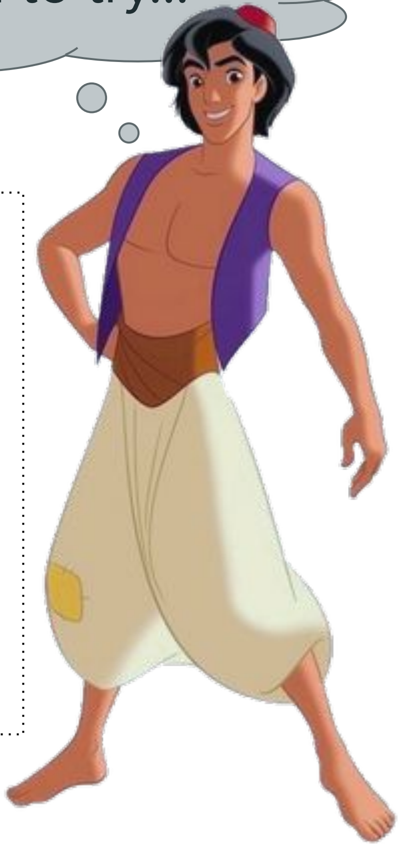
CART (trees) models

XGBoost (ensemble) models

(Demo: try yourself)

As Java CLI app

For you to try...



Go to this link:

Tribuo Regression example: as a Java CLI app

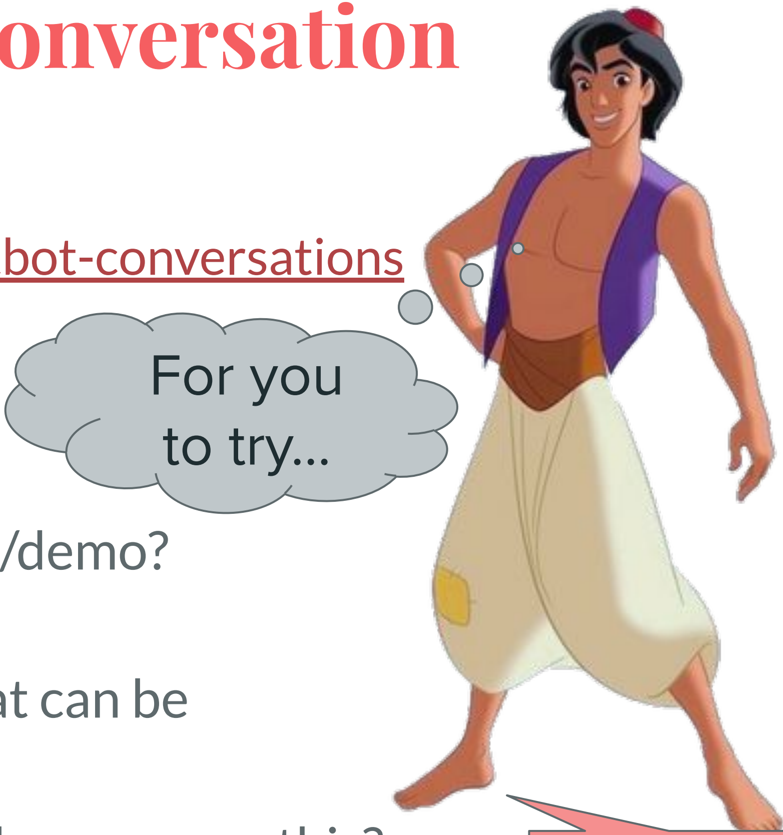
And please perform the steps mentioned there.

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

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

Making chatbot have a conversation

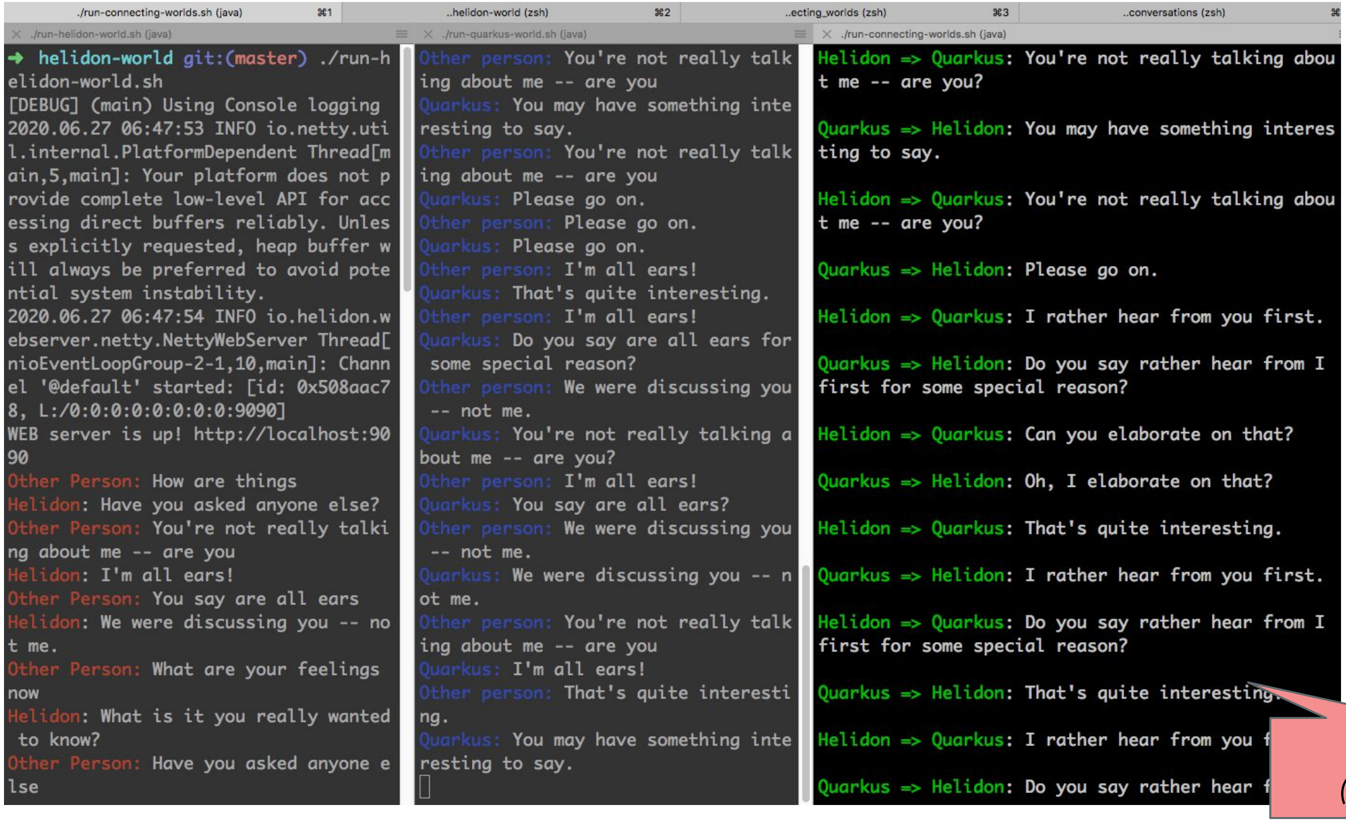
- Github repo: https://github.com/neomatrix369/chatbot-conversations
- Questions
    - Is this a real conversation?
    - Would this pass the Turing test?
    - What is good about this example/demo?
- Puzzles to solve
    - What are the different things that can be improved?
    - What new ideas come to mind when you see this?



Video

(Chatbots demo)

Making chatbot have a conversation


./run-connecting-worlds.sh (java) ⌘1
..helidon-world (zsh) ⌘2
..ecting_worlds (zsh) ⌘3
..conversations (zsh) ⌘
./run-helidon-world.sh (java)
./run-quarkus-world.sh (java)
./run-connecting-worlds.sh (java)

```
➜  helidon-world git:(master) ./run-h
elidon-world.sh
[DEBUG] (main) Using Console logging
2020.06.27 06:47:53 INFO io.netty.uti
l.internal.PlatformDependent Thread[m
ain,5,main]: Your platform does not p
rovide complete low-level API for acc
essing direct buffers reliably. Unles
s explicitly requested, heap buffer w
ill always be preferred to avoid pote
ntial system instability.
2020.06.27 06:47:54 INFO io.helidon.w
ebserver.netty.NettyWebServer Thread[
nioEventLoopGroup-2-1,10,main]: Chann
el '@default' started: [id: 0x508aac7
8, L:/0:0:0:0:0:0:0:0:9090]
WEB server is up! http://localhost:90
90
Other Person: How are things
Helidon: Have you asked anyone else?
Other Person: You're not really talki
ng about me -- are you
Helidon: I'm all ears!
Other Person: You say are all ears
Helidon: We were discussing you -- no
t me.
Other Person: What are your feelings
now
Helidon: What is it you really wanted
 to know?
Other Person: Have you asked anyone e
lse
```

```
Other person: You're not really talk
ing about me -- are you
Quarkus: You may have something inte
resting to say.
Other person: You're not really talk
ing about me -- are you
Quarkus: Please go on.
Other person: Please go on.
Quarkus: Please go on.
Other person: I'm all ears!
Quarkus: That's quite interesting.
Other person: I'm all ears!
Quarkus: Do you say are all ears for
 some special reason?
Other person: We were discussing you
 -- not me.
Quarkus: You're not really talking a
bout me -- are you?
Other person: I'm all ears!
Quarkus: You say are all ears?
Other person: We were discussing you
 -- not me.
Quarkus: We were discussing you -- n
ot me.
Other person: You're not really talk
ing about me -- are you
Quarkus: I'm all ears!
Other person: That's quite interesti
ng.
Quarkus: You may have something inte
resting to say.
▯
```

```
Helidon => Quarkus: You're not really talking abou
t me -- are you?

Quarkus => Helidon: You may have something interes
ting to say.

Helidon => Quarkus: You're not really talking abou
t me -- are you?

Quarkus => Helidon: Please go on.

Helidon => Quarkus: I rather hear from you first.

Quarkus => Helidon: Do you say rather hear from I
first for some special reason?

Helidon => Quarkus: Can you elaborate on that?

Quarkus => Helidon: Oh, I elaborate on that?

Helidon => Quarkus: That's quite interesting.

Quarkus => Helidon: I rather hear from you first.

Helidon => Quarkus: Do you say rather hear from I
first for some special reason?

Quarkus => Helidon: That's quite interesting

Helidon => Quarkus: I rather hear from you f

Quarkus => Helidon: Do you say rather hear f
```

Video
(Chatbots demo)

Java is a tool, a programming language.

We can use it to create anything that are created by other programming languages!

Resources

Learning websites

- Awesome AI/ML/DL
- Better NLP
- Awesome Graal
- Virgilio | GitHub
- MadeWithML

All you need to know, pick and choose, make your own collections / checklists / playlists of resources to study and practice from.

Java AI/ML/DL Resources

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

Other Java ML 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
- Some Java & AI/ML talks this week...

More resources

See Appendix section

Summary

- My journey and how I learnt the topics
- We get better results when we organise ourselves
- Create something simple from scratch, similar to our demo and other examples
- A simple idea can go very far, inspire others with creative solutions

Summary

- Java is versatile and can be used to create amazing things just like others are creating
- We didn’t cover even more number of Java and AI/ML/DL related topics, libraries, framework -- I’m hoping that the resources shared here hopefully will lead up to them
- No need to feel overwhelmed and let’s try to avoid “imposter syndrome”

Summary

The pathway to mastery or championing a subject means we must be different and take different approaches, in addition to the ones that have already been taken!

Thank you!

You

And organisers of the

Oracle Groundbreakers APAC Virtual Tour 2020

Use Hashtag:
#GroundbreakersTour
#GBT2020

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!

Questions & feedback

Use Hashtag:
#GroundbreakersTour
#GBT2020

Please share your questions and feedback at

@theNeomatrix369

or on the video stream

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:
#GroundbreakersTour
#GBT2020

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 others…

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

Accelerated Processing

Java on the GPU by Mitia

Plenty of resources on NVIDIA

Nvidia's Developer site | Community | Research | Blog site

Parallel processing

GNU Parallel

https://www.gnu.org/software/parallel/

Parallel / Async programming

Talk by Venkat Subramanian

ASYNC

Reactive programming

https://community.oracle.com/docs/DOC-1006738

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?

Levels of abstractions

- Higher to lower
  - From generic to more specific
- Always know the level of abstract
  - when reading
  - when writing
  - when speaking

Try to analyse this presentation, it's structure, the different levels explored...

See how we can get to know many topics and yet stay high-level, and go deep only when required.

Book references

- How to think like a Scientist!
- Deep work book
- Getting Things Done: Book | Free Resources

Previous talks

- I recently gave a talk: Tribuo: an introduction to a Java ML Library
- **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 (*jOnConf*)
- 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)

Create something simple from scratch

- even if it's as simple or silly idea - like the demos i have shown
- *(it can also be putting existing components together from scratch)*
- this can have a massive impact on us and our communities

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

graql-to-english, english-to-graql example

- Presentation
- Github

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