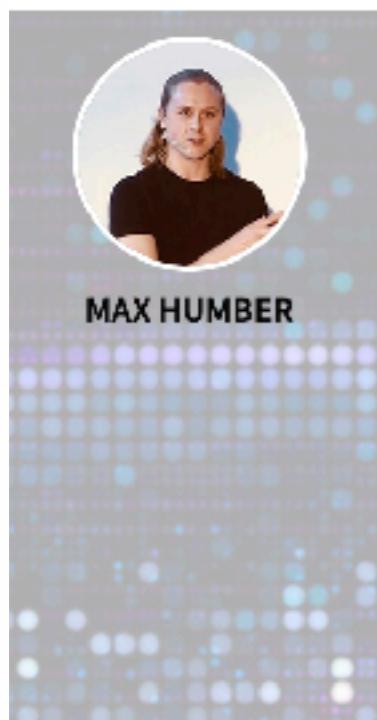


Wi-Fi LIVE ONLINE TRAINING

Data Engineering for Data Scientists

Build resilient pipelines to support stronger models



MAX HUMBER



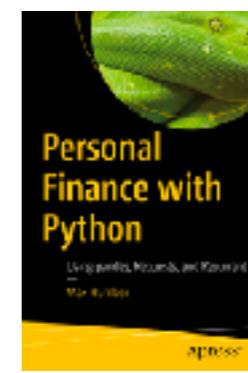
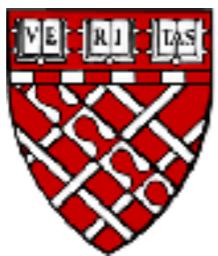
August 6, 2019

6:00pm – 8:00pm CEST

Repo:

<https://github.com/maxhumber/air>

\$ whoami



Talks



Agenda

1

Extend

2

Save

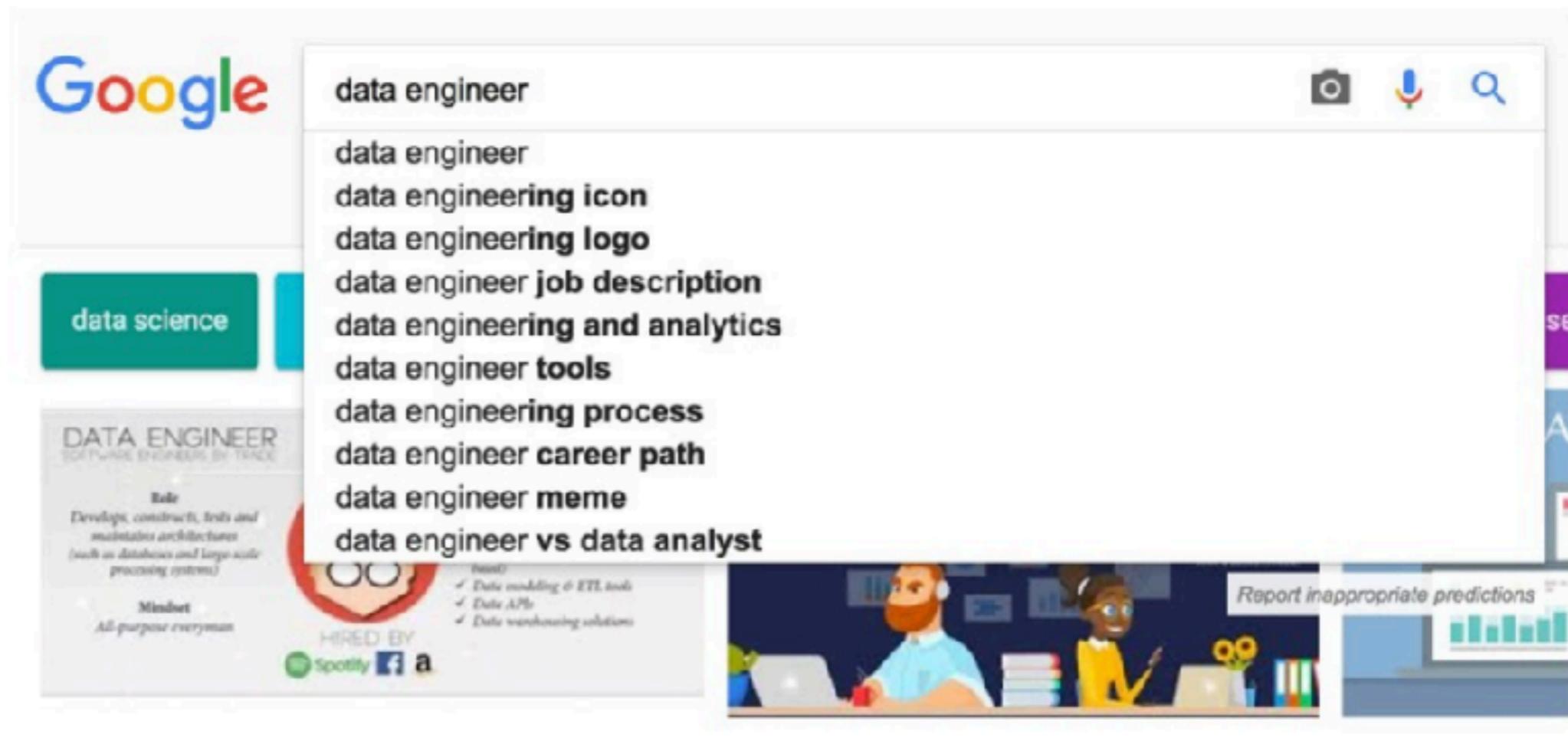
3

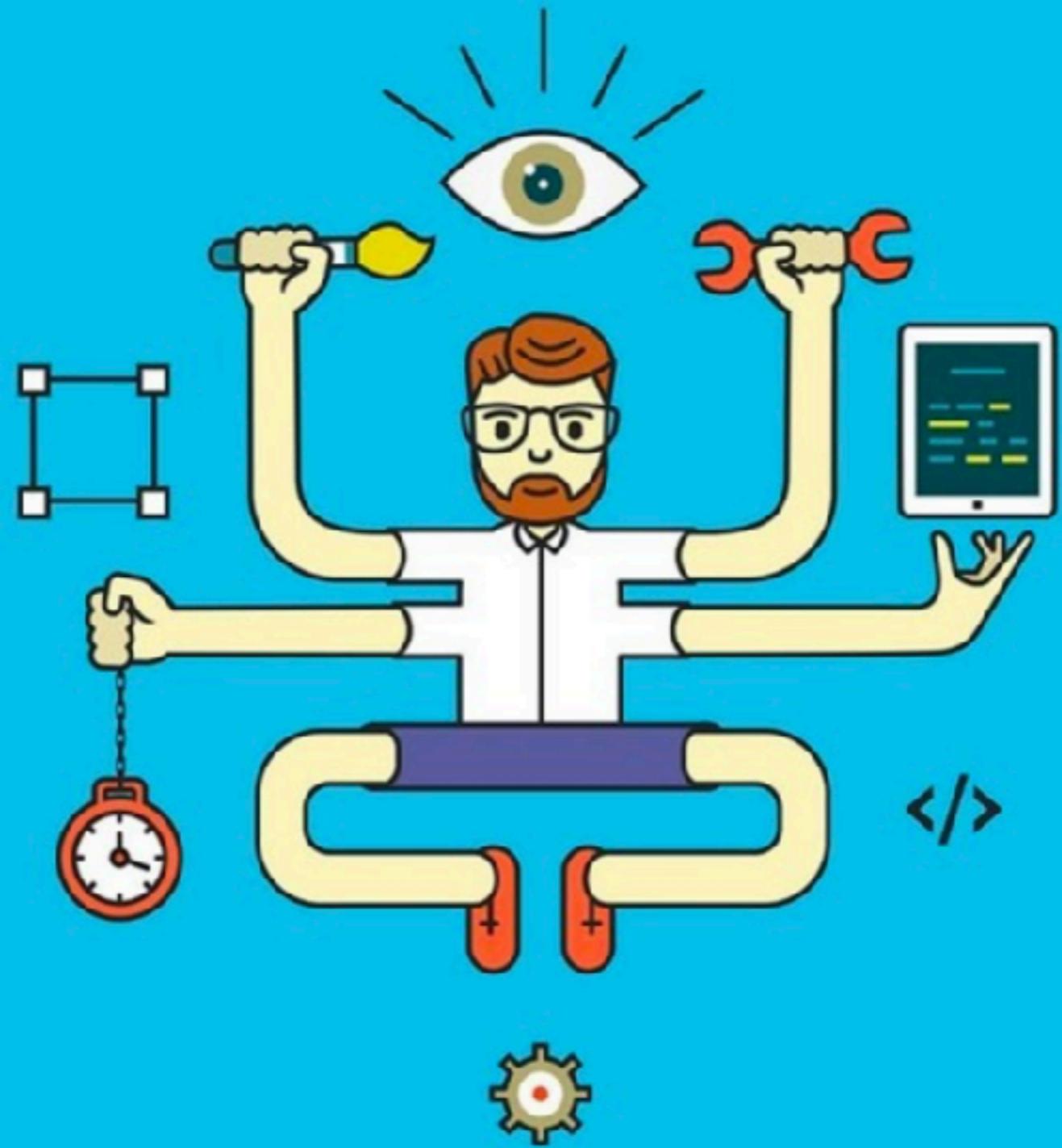
Schedule

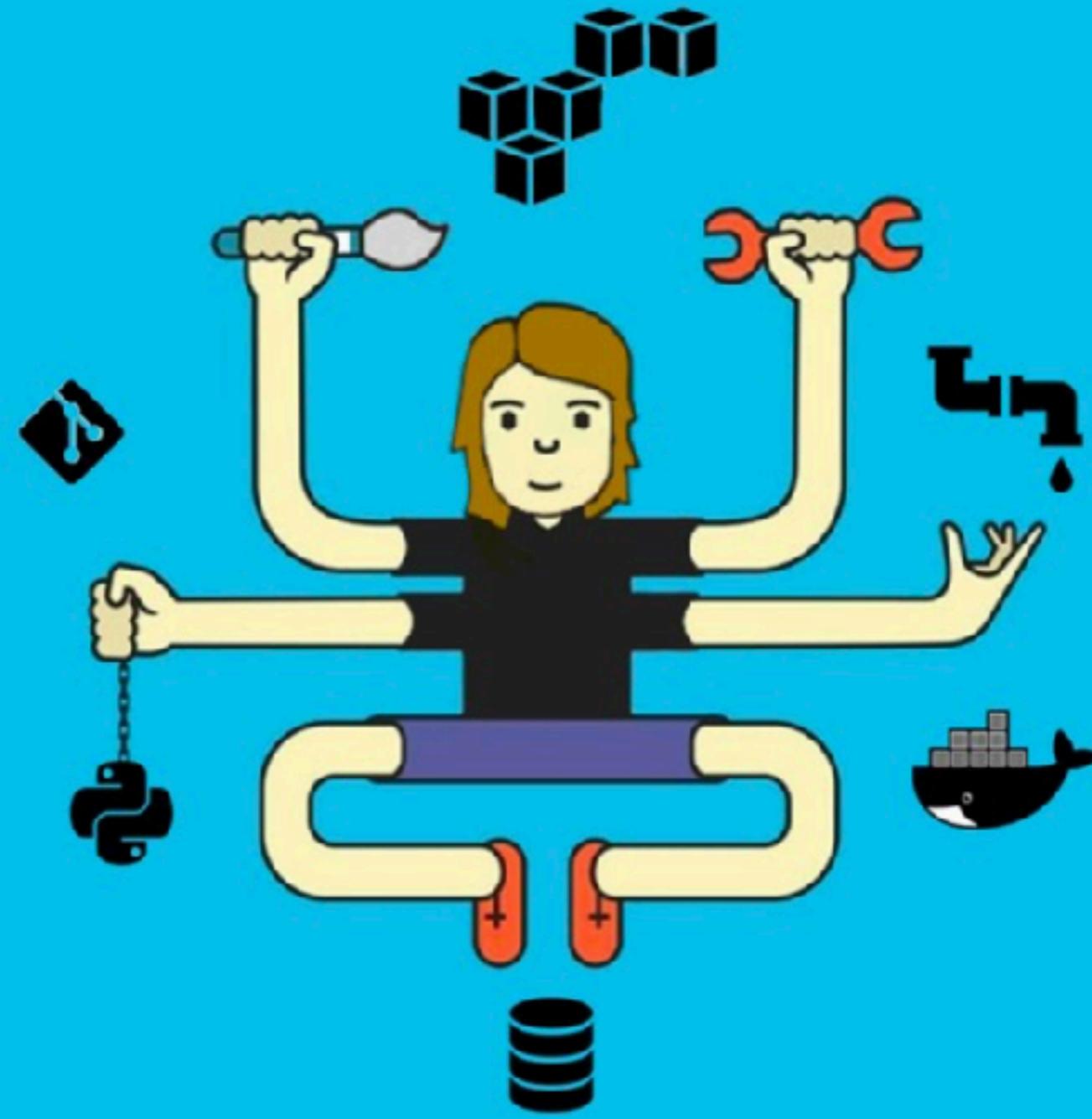
Polls

- 1. Do you have an ML model in production?**
- 2. Do you own the production pipeline?**
- 3. How many DS/DEs are on your team? {1,5,10,++}**

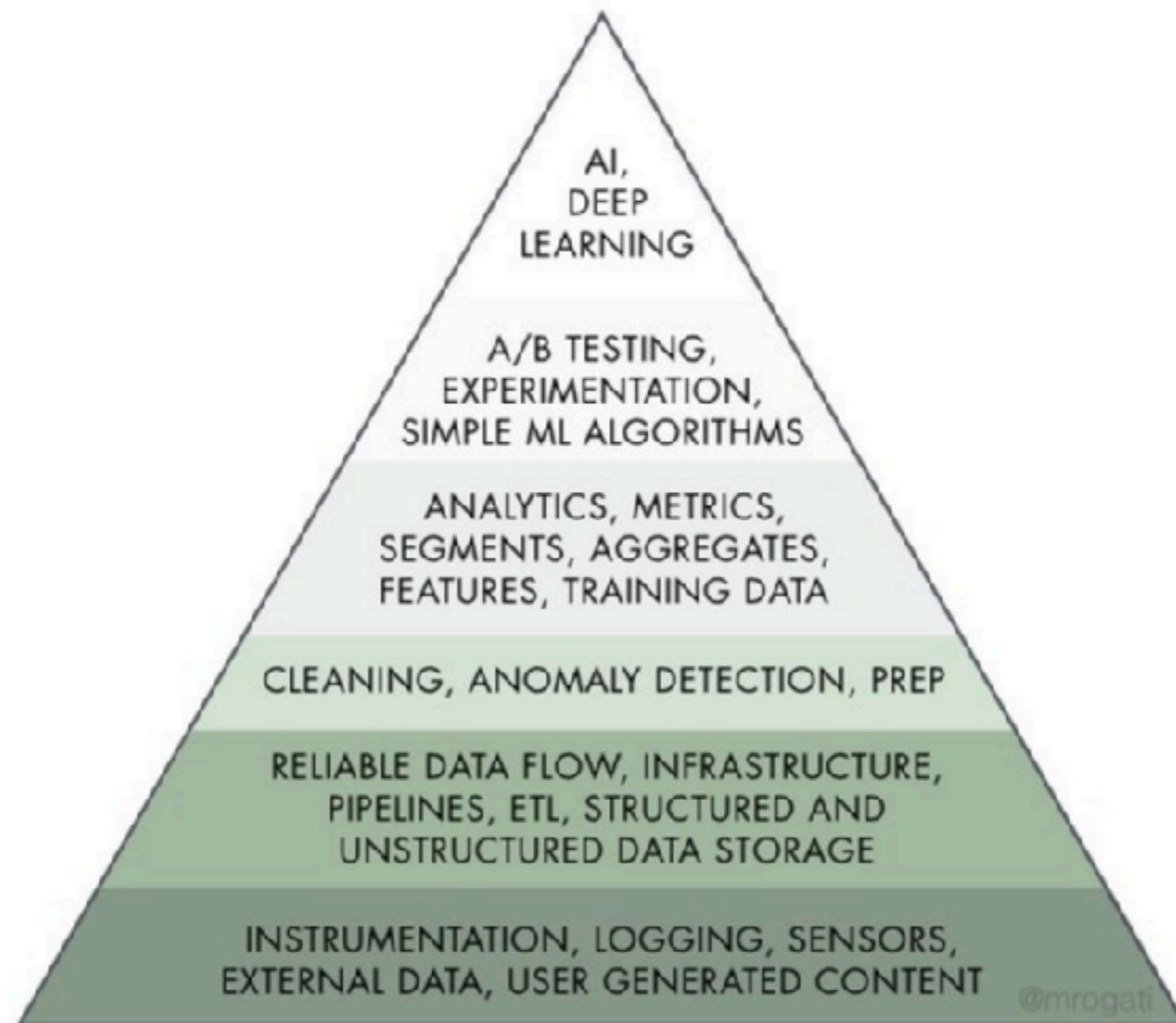
Data Engineering



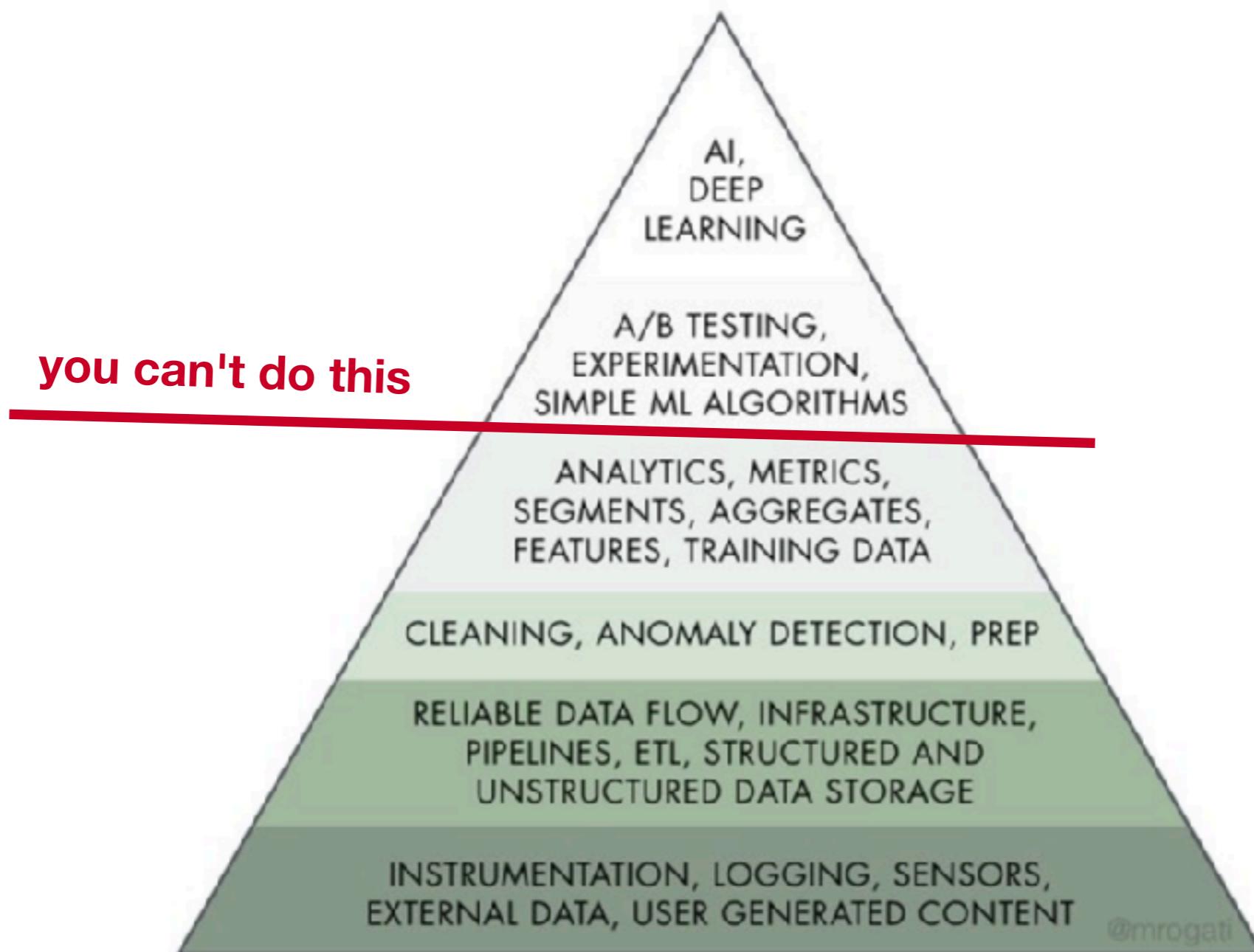




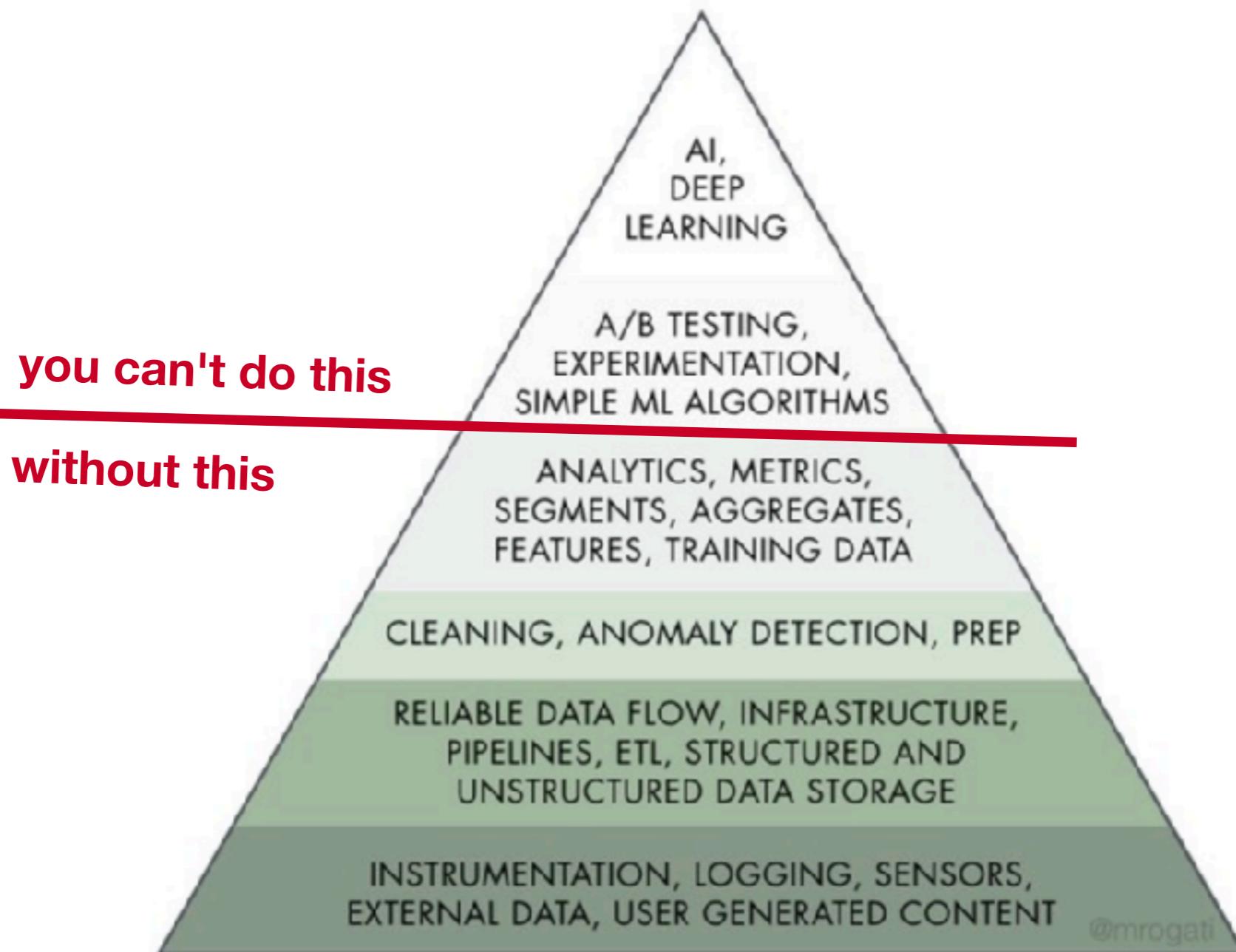
Data Hierarchy of Needs



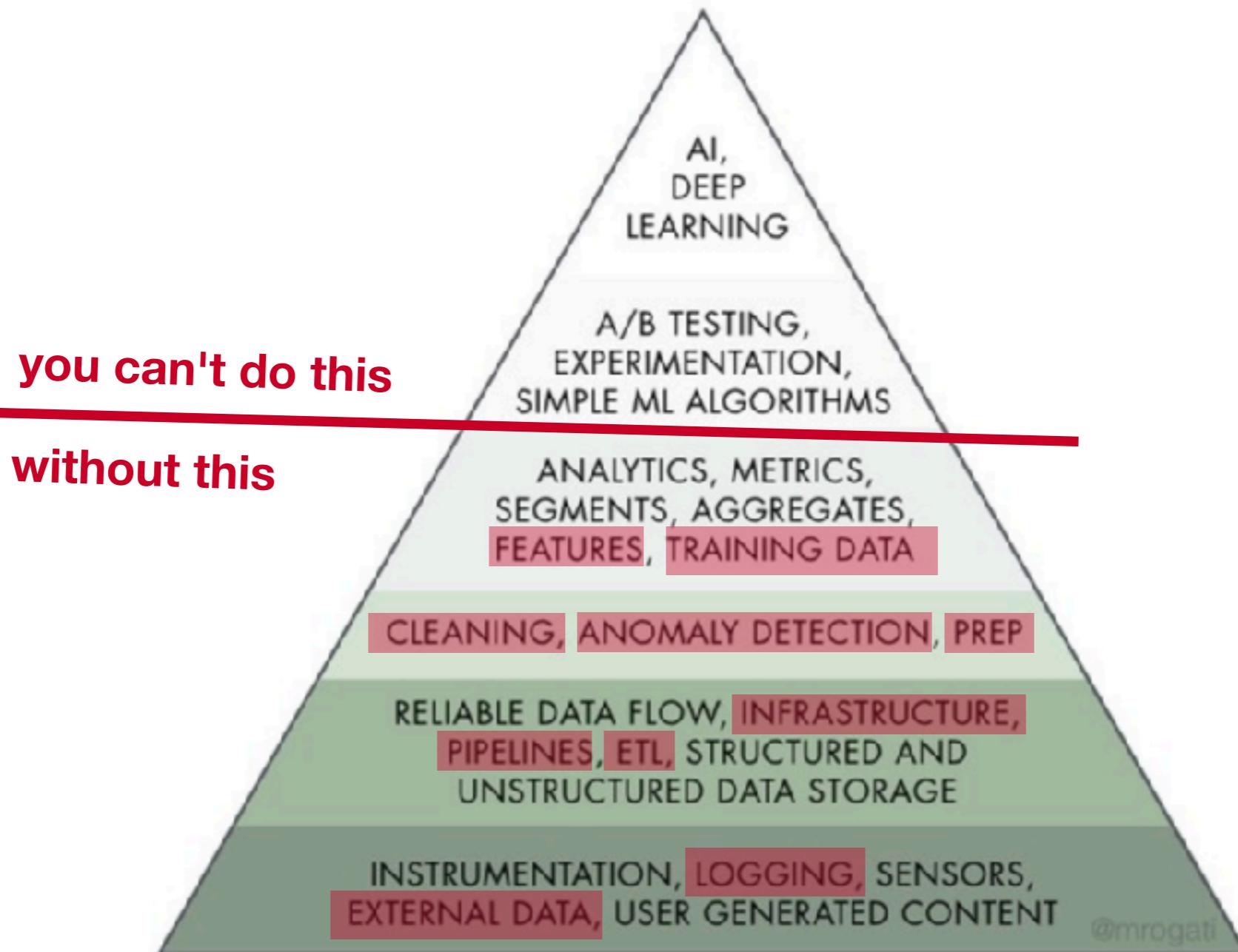
Data Hierarchy of Needs

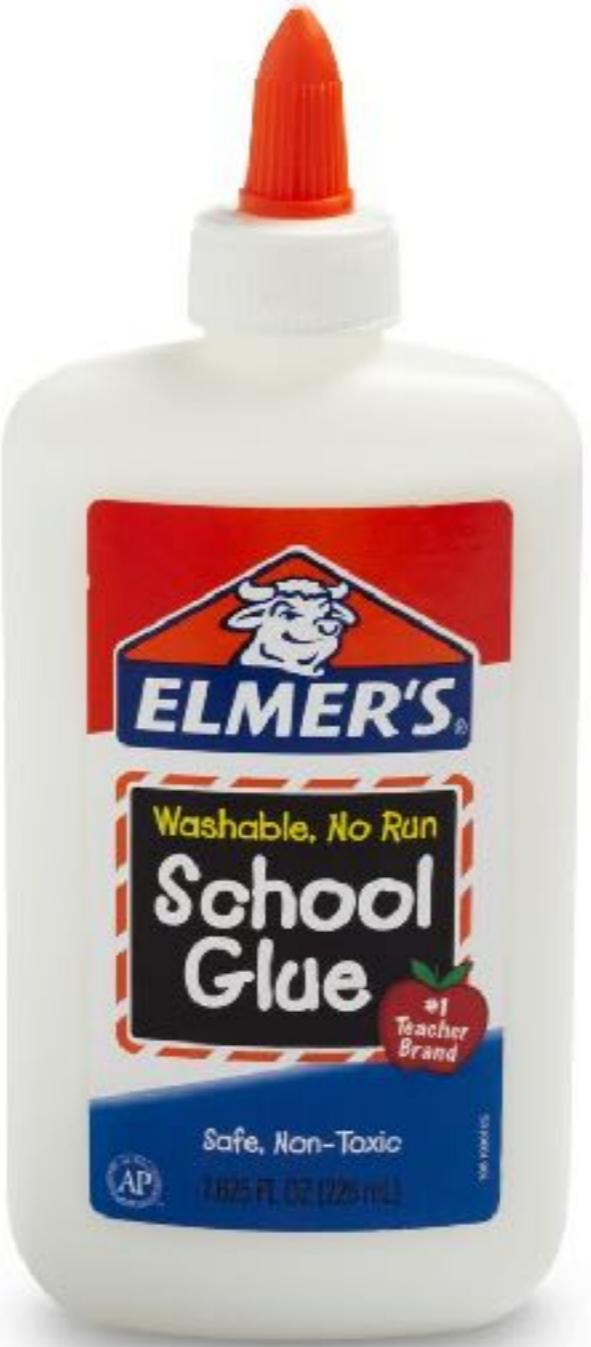


Data Hierarchy of Needs

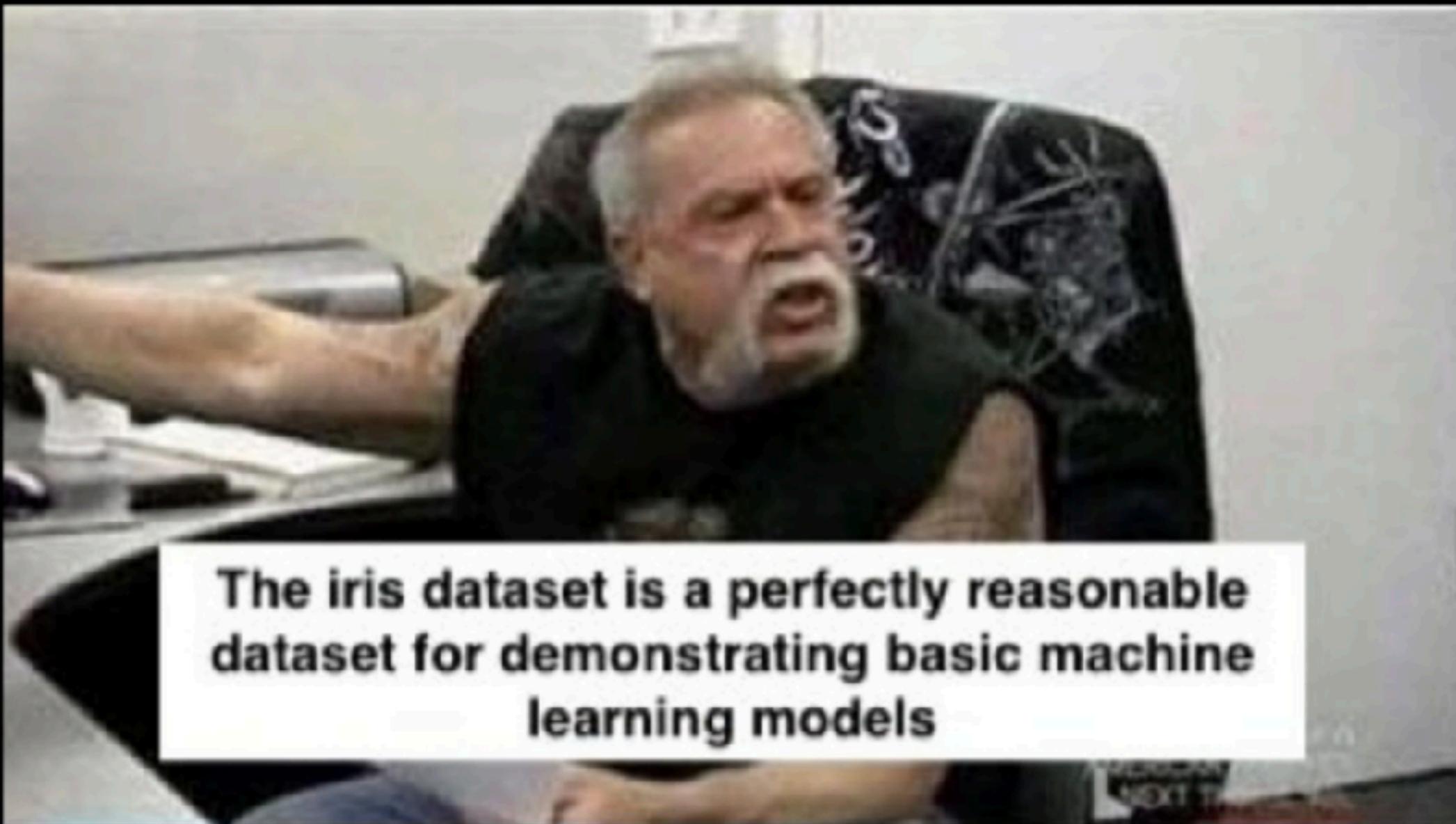


Data Hierarchy of Needs









**The iris dataset is a perfectly reasonable
dataset for demonstrating basic machine
learning models**



**It lulls beginners into falsely
thinking working with data is
easy**

ALL NEW
VAN CHOPPER
17 THOUSAND WIRE



**EVERYONE NEEDS TO START
SOMEWHERE**



IT'S THE WRONG
PLACE FOR
BEGINNERS TO
START

AMERICAN DRAFT
NEXT. THERE'S NO



**YOU DON'T
REMEMBER
WHAT IT'S LIKE
TO BE A
BEGINNER**







PONDOK K

NPG

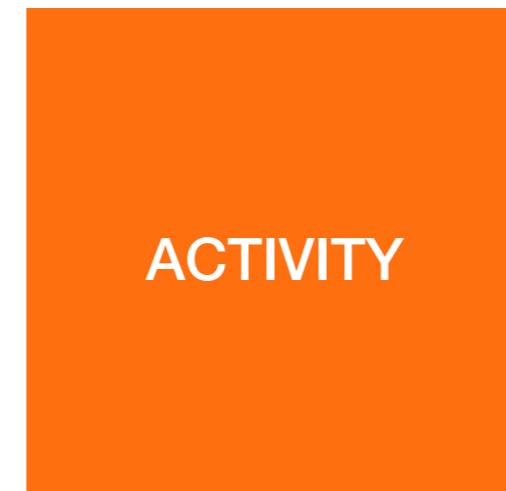
KEEPEACKAGING.COM

W

1

Extend

live coding
➤ **model.ipynb**



ACTIVITY

activities/mapper.ipynb

Q&A

Jupyter to Hydrogen



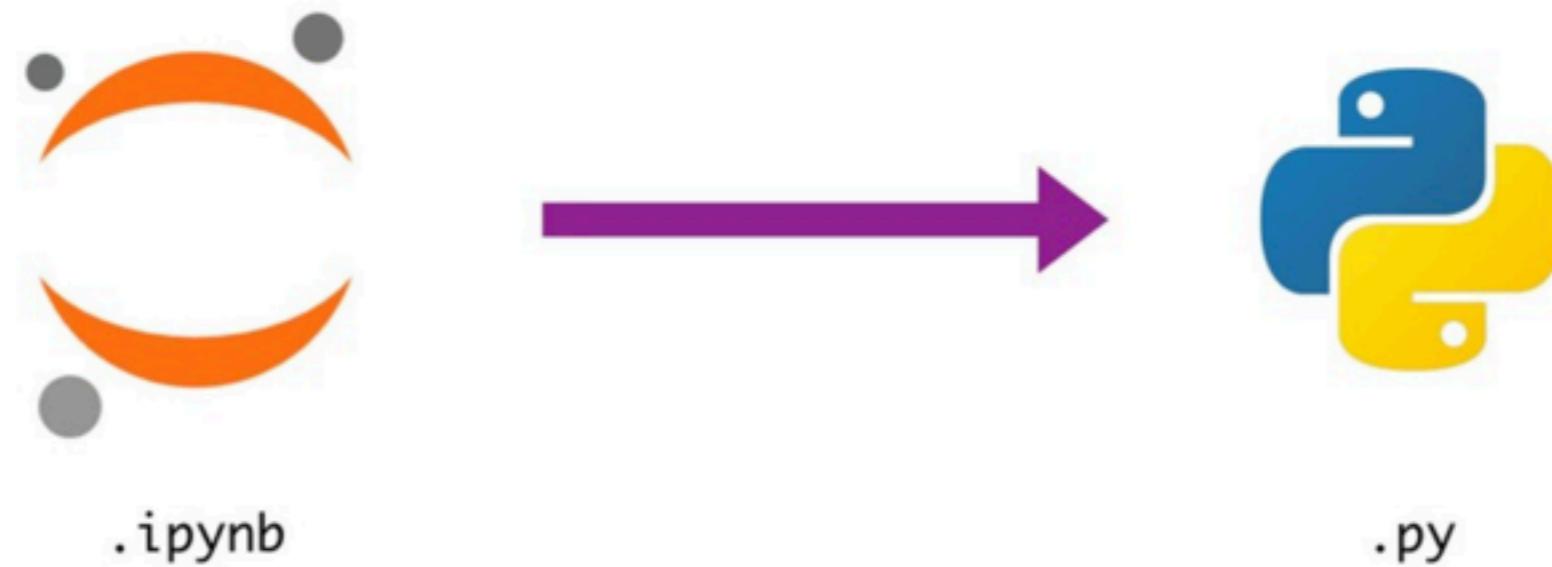
Jupyter to Hydrogen



- exploratory analysis
- visualizing ideas
- prototyping

- messy
- bad at versioning
- not ideal for production

Jupyter to Hydrogen



```
$ jupyter nbconvert --to script model.ipynb
```

Part I: Extend

Need data to talk about data, and a model to talk about models...

The Data

```
In [1]: import pandas as pd  
  
df = pd.read_csv('data/hockey.csv')
```

```
In [2]: df.shape
```

```
Out[2]: (657, 13)
```

```
In [3]: df[df['name'] == 'Auston Matthews'].head(3)
```

```
Out[3]:
```

1	2	matthau01	Auston Matthews	C	2018- 10-03	TOR	Home	MTL	W	2	0		
15	16	matthau01	Auston Matthews	C	2018- 10-06	TOR	Home	OTT	L	1	1		

```
$ jupyter nbconvert --to script model.ipynb
```

```
model.py
```

```
1 #!/usr/bin/env python-
2 # coding: utf-8-
3
4 # # Part I: Extend-
5
6 # Need data to talk about data, and a model to talk about models...-
7
8 # ### The Data-
9
10 # In[1]:-
11
12
13 import pandas as pd-
14
15 df = pd.read_csv('data/hockey.csv')-
16
17
18 # In[2]:-
```

Hydrogen



slack 12/773 build passing

Hydrogen is an interactive coding environment that supports Python, R, JavaScript and other Jupyter kernels.



The screenshot shows the Hydrogen Atom package integrated into the Atom text editor. The main window displays a Python script named 'demo.py' with the following code:

```
demo.py — /Users/lukasgeiger/Desktop
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
```

At the bottom of the code editor, there is a status bar showing 'demo.py 33:1 Python 3: idle'. To the right of the code editor, there is a preview panel displaying a table from the script:

	A	B	C	D	E	F	X
0	1.0	2013-01-02	1.0	3	test	foo	
1	1.0	2013-01-02	1.0	3	train	foo	
2	1.0	2013-01-02	1.0	3	test	foo	
3	1.0	2013-01-02	1.0	3	train	foo	

The status bar at the bottom right indicates 'LF UTF-8 Python'.

<https://atom.io/packages/hydrogen>

live coding

> model_slim.py

Q&A





Python Fire

python 2.7 | 3.4 | 3.5 | 3.6

Python Fire is a library for automatically generating command line interfaces (CLIs) from absolutely any Python object.

- Python Fire is a simple way to create a CLI in Python. [\[1\]](#)
- Python Fire is a helpful tool for developing and debugging Python code. [\[2\]](#)
- Python Fire helps with exploring existing code or turning other people's code into a CLI. [\[3\]](#)
- Python Fire makes transitioning between Bash and Python easier. [\[4\]](#)
- Python Fire makes using a Python REPL easier by setting up the REPL with the modules and variables you'll need already imported and created. [\[5\]](#)

Installation

To install Python Fire with pip, run: `pip install fire`

To install Python Fire with conda, run: `conda install fire -c conda-forge`

To install Python Fire from source, first clone the repository and then run: `python setup.py install`

live coding

> predict_scaffold.py

live coding

> predict.py



python predict.py ovechal01



ACTIVITY

activities/fire_starter.py

Q&A

2

Save

live coding

> predict_to_database.py



python predict_to_database.py ovechal01



Monitor



THIS IS YOUR MACHINE LEARNING SYSTEM?

YUP! YOU POUR THE DATA INTO THIS BIG
PILE OF LINEAR ALGEBRA, THEN COLLECT
THE ANSWERS ON THE OTHER SIDE.

WHAT IF THE ANSWERS ARE WRONG?

JUST STIR THE PILE UNTIL
THEY START LOOKING RIGHT.





FREE EDITION

5,000 events per month.
Includes all Standard features.

[Try it free](#)

⌚ Waiting for data from your Python app...

This page will update once we receive an error from your app.

Don't want to start with Python? [Choose a different SDK.](#)

Installation

To send errors to Rollbar from your Python application you should use the [pyrollbar](#) SDK. Install pyrollbar with pip:

```
pip install rollbar
```

COPY

Send a Message

You'll need your project's server-side access token to initialize the pyrollbar SDK. Sending a message to the Rollbar server is as simple as:

```
import rollbar  
  
rollbar.init('49e86c1178e346899c2f29e3744420a9')  
rollbar.report_message('Rollbar is configured correctly')
```

COPY

A few seconds after you execute this code, the message should appear on your project's "Items" page.



remove password

Search

Repositories	244
Code	50M+
Commits	508K
Issues	280K
Packages	1
Marketplace	0
Topics	0
Wikis	34K
Users	0

[Advanced search](#) [Cheat sheet](#)

Showing 445,215 available commit results ?

Sort: Recently committed ▾

removed password ...

Verified



c64833c



reharr4 committed to [reharr4/blamazon](#) 2 hours ago

Onmibus updates (#7) ...

Verified



b1b80c9



stevector committed to [stevector/internet-button-playground](#) 4 hours ago ✓

Merge pull request #534 in EDD/edd-django from ~WCMORRELL/edd:feature... ...

8aa8978



wmorrell committed to [JBEI/edd](#) 7 hours ago

File restructuring ...

caitlin authored and PilotBob committed to [CassandraWerewolf/CassandraWeb](#) 7 hours ago



8735106



removed password

removed password from bamazonCustomer.js

master

 reharr4 committed 2 hours ago Verified

1 parent 7e38084 commit c64833c915cc0f604cb0eeab

 Showing 1 changed file with 2 additions and 2 deletions.

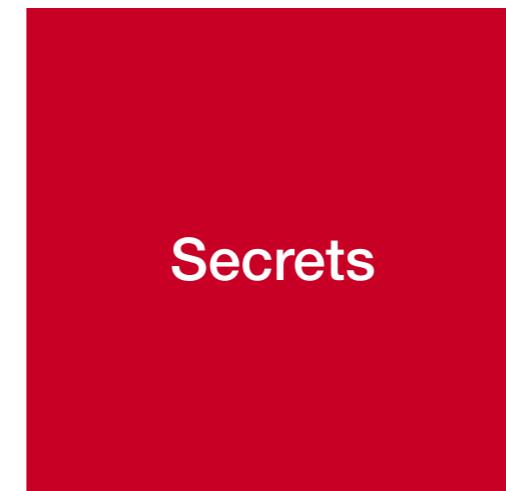
▼ 4  bamazonCustomer.js 

```
@@ -9,7 +9,7 @@ var connection = mysql.createConnection({  
 9   9     // username  
10  10    user: "root",  
11  11    // password  
12  -    password: "L!sb0n13",  
12  +    password: "",  
13  13    database: "bamazon_db"  
14  14});  
15  15
```

```
@@ -94,4 +94,4 @@ connection.connect(function (err) {  
94  94    //      })  
95  95    //      })  
96  96    // }  
97  -    connection.end(); @@  
97  +    connection.end();
```

PSSST...!





Secrets



python-dotenv | build passing coverage 90% pypi package 0.10.3 Say Thanks !

Reads the key,value pair from `.env` file and adds them to environment variable. It is great for managing app settings during development and in production using [12-factor](#) principles.

| Do one thing, do it well!

- [Usages](#)
- [Installation](#)
- [Command-line interface](#)
- [iPython Support](#)
- [Setting config on remote servers](#)
- [Related Projects](#)
- [Contributing](#)
- [Changelog](#)

Installation

```
pip install -U python-dotenv
```

live coding

> **model_rollbar.py**



ACTIVITY

activities/secret_sandbox.py

3

Schedule



Airflow

Apache Airflow

[pypi package 1.10.3](#) [build passing](#) [coverage 79%](#) [docs passing](#) [license Apache 2](#) [python 2.7 | 3.5](#) [Follow](#) [2.4k](#) [Slack](#) [join chat](#)

Apache Airflow (or simply Airflow) is a platform to programmatically author, schedule, and monitor workflows.

When workflows are defined as code, they become more maintainable, versionable, testable, and collaborative.

Use Airflow to author workflows as directed acyclic graphs (DAGs) of tasks. The Airflow scheduler executes your tasks on an array of workers while following the specified dependencies. Rich command line utilities make performing complex surgeries on DAGs a snap. The rich user interface makes it easy to visualize pipelines running in production, monitor progress, and troubleshoot issues when needed.

Getting started

Please visit the Airflow Platform documentation (latest **stable** release) for help with [installing Airflow](#), getting a [quick start](#), or a more complete [tutorial](#).

Documentation of GitHub master (latest development branch): [ReadTheDocs Documentation](#)

For further information, please visit the [Airflow Wiki](#).

Tutorial

This tutorial walks you through some of the fundamental Airflow concepts, objects, and their usage while writing your first pipeline.

Example Pipeline definition

Here is an example of a basic pipeline definition. Do not worry if this looks complicated, a line by line explanation follows below.

```
"""
Code that goes along with the Airflow tutorial located at:
https://github.com/apache/airflow/blob/master/airflow/example_dags/tutorial.py
"""

from airflow import DAG
from airflow.operators.bash_operator import BashOperator
from datetime import datetime, timedelta

default_args = {
    'owner': 'airflow',
    'depends_on_past': False,
    'start_date': datetime(2015, 6, 1),
    'email': ['airflow@example.com'],
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
    # 'queue': 'bash_queue',
    # 'pool': 'backfill',
    # 'priority_weight': 10,
    # 'end_date': datetime(2016, 1, 1),
}

dag = DAG('tutorial', default_args=default_args, schedule_interval=timedelta(days=1))
```

Tutorial

This tutorial walks you through some of the fundamental Airflow concepts, objects, and their usage while writing your first pipeline.

Example Pipeline definition

Here is an example of a basic pipeline definition. Do not worry if it looks complicated, a line by line explanation follows below.

```
"""
Code that goes along with the Airflow tutorial located at:
https://github.com/apache/airflow/blob/master/airflow/example_dags/tutorial.py
"""

from airflow import DAG
from airflow.operators.bash_operator import BashOperator
from datetime import datetime, timedelta

default_args = {
    'owner': 'airflow',
    'depends_on_past': False,
    'start_date': datetime(2015, 6, 1),
    'email': ['airflow@example.com'],
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=5),
    # 'queue': 'bash_queue',
    # 'pool': 'backfill',
    # 'priority_weight': 10,
    # 'end_date': datetime(2016, 1, 1),
}

dag = DAG('tutorial', default_args=default_args, schedule_interval=timedelta(days=1))
```

live coding

```
cd air
```

```
# airflow needs a home  
export AIRFLOW_HOME=`pwd`/airflow
```

```
pip install apache-airflow
```

```
# in airflow.cfg set:  
# load_examples = False
```

```
# initialize the database  
airflow initdb
```



live coding

> **airflow/dags/hockey.py**

airflow test hockey fetch 2019-02-15

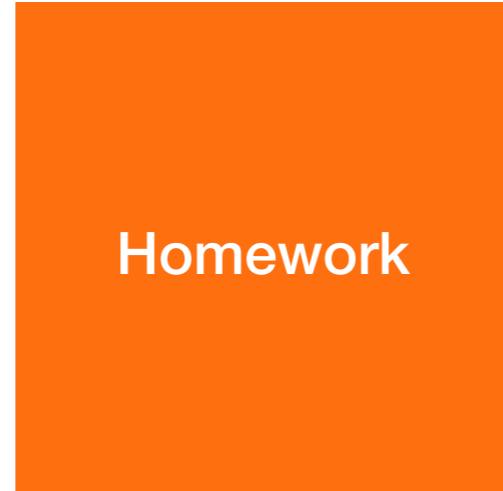


airflow test hockey fetch 2019-02-15

```
# start the web server  
airflow webserver -p 8080
```

```
# start the scheduler  
airflow scheduler
```





Homework

add t3 to save predictions to the db

Q&A

That's all Folks!