Building IMU-based Gesture Recognition!!!

Jennifer Wang
jewang.net

Hello world!

Jennifer Wang

Software Engineer & Hardware Prototyper

I <3 sensors

- Phones
- Wearables
- Robots
- Telescopes

jewang.net / jen@jewang.net

Preamble

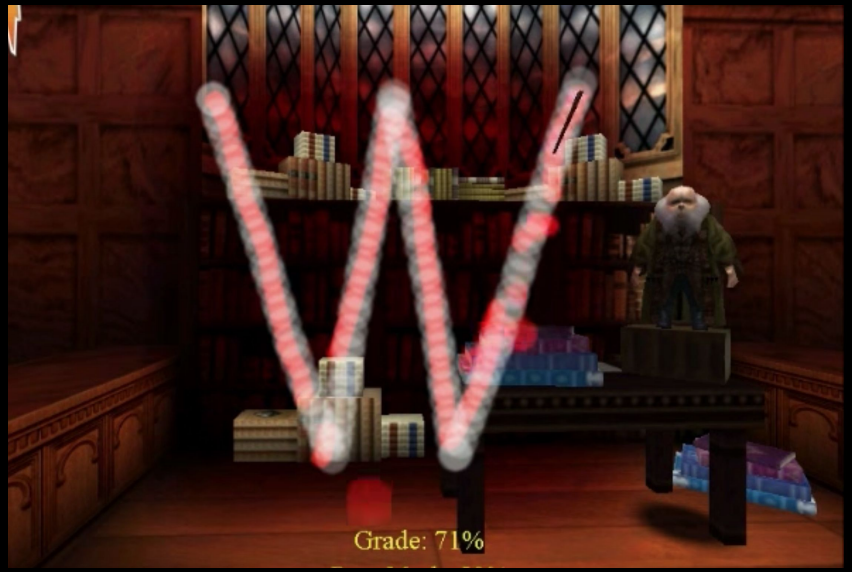
Thank you to these lovely people:

- Tim ‘mithro’ Ansell
- Kat Fang
- Cynthia Gan
- Samy Kamkar
- Sophi Kravitz
- Jinna Lei
- Jen Tong
- Tony Valderrama
- Ruth Grace Wong

Please go vote!


Harry Potter
AND THE SORCERER'S STONE
EVERYONE
E
CONTENT RATED BY
ESRB
EA GAMES


That door in front of you is magically locked. Cast Alohomora to open it.



Wingardium (Leviosa)
Levitation

Flippendo
Bread and butter

Final Product

https://github.com/jewang/gesture-demo

Explorations

localhost:8890/notebooks/dev/gesture-demo/Explorations.ipynb

jupyter Explorations Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Read & trim data

In [25]:

```
FOLDER_NAMES = ["wingardium", "flippendo", "negative", "test"]
GESTURES = ["wingardium", "flippendo", "negative_trim"]

# Manually determined trims (copied from Google Sheets)
TRIMS = pd.read_csv('trim_ms.csv', header=0, index_col=0, squeeze=True)
TRIMS = pd.to_numeric(TRIMS, errors='coerce').to_dict()

SEGMENT_LEN_MS = 1500
SAMPLE_FREQ_HZ = 100
SEGMENT_LEN = int(SEGMENT_LEN_MS / SAMPLE_FREQ_HZ * 10) - 5
```

In [26]:

```
ef read_data(folder_name):
    traces = []
    trace_names = []
    for f in sorted(glob.glob(folder_name + "/*.csv")):
        print(f)
        filename = f[f.find('/') + 1:-4]
        trace = pd.read_csv(f, header=0, index_col="time_ms")
        print(trace.shape)
        trace = trace.apply(pd.to_numeric, errors='coerce')
        trace["accel"] = np.linalg.norm((trace["accel_ms2_x"], trace["accel_ms2_y"], trace["
        trace["gyro"] = np.linalg.norm((trace['gyro_degs_x'], trace[u'gyro_degs_y'], trace[u

        trace_names.append(filename)
        traces.append(trace)
    return trace_names, traces

races = {}
race_names = {}
or folder_name in FOLDER_NAMES:
    trace_names[folder_name], traces[folder_name] = read_data(folder_name)
```

```
(5987, 24)
negative/negative001.csv
(5986, 24)
negative/negative002.csv
(5972, 24)
negative/negative003.csv
(5971, 24)
negative/negative004.csv
(5982, 24)
```

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

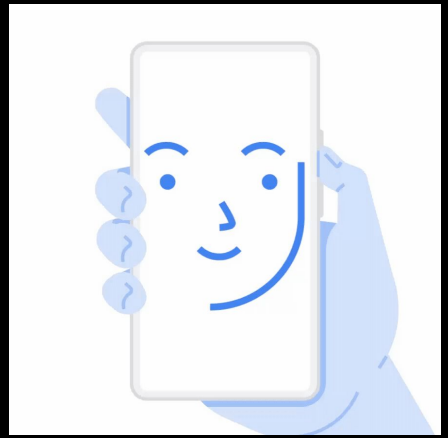
IMUs + Machine Learning = Lots of fun!

Real firebending (Allen Pan)

Smartwatch gestures/input/stealing ATM Pins

Music Control Gloves (Mimu Gloves used by Imogen Heap)

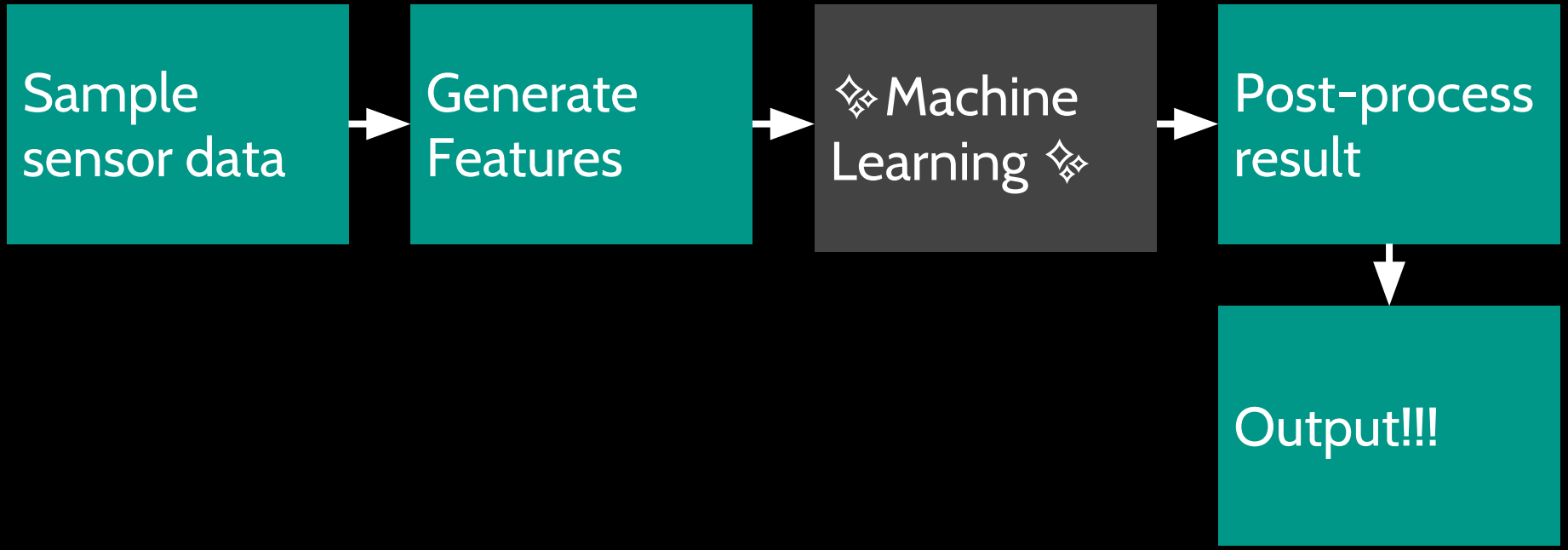
Gait-based auth



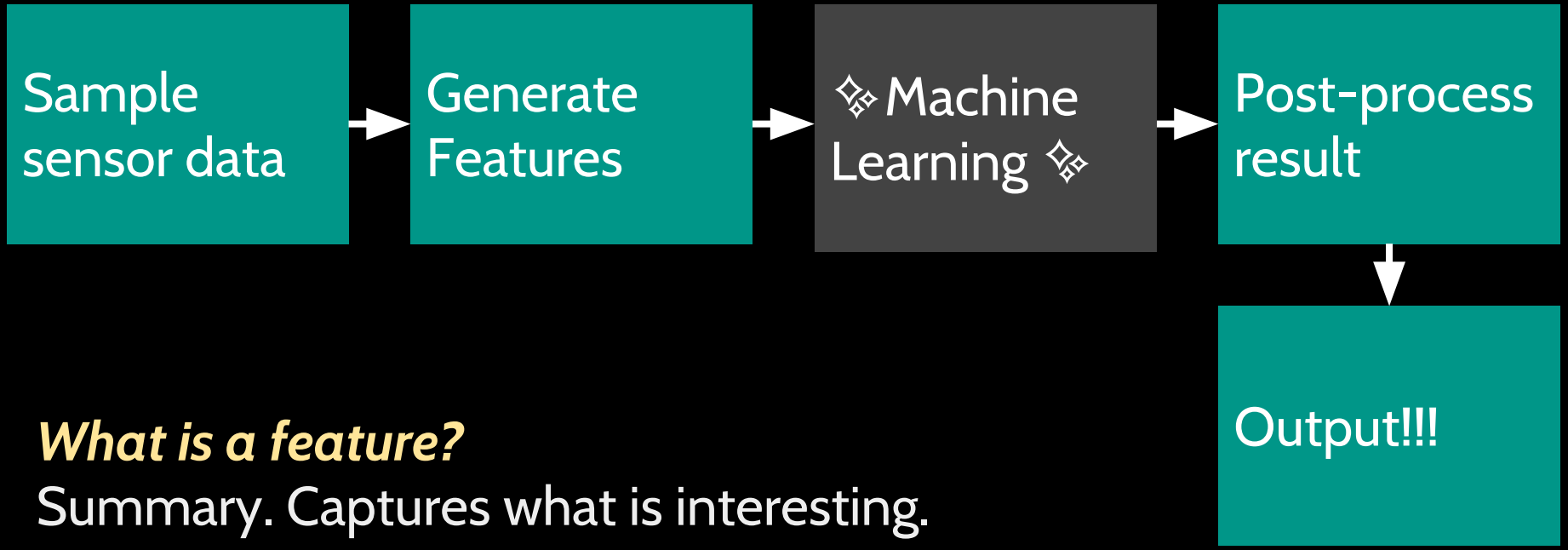
Android Camera Control Gestures

Fitness / Sleep Tracking "actigraphy" (FitBit)

What will the final product look like?



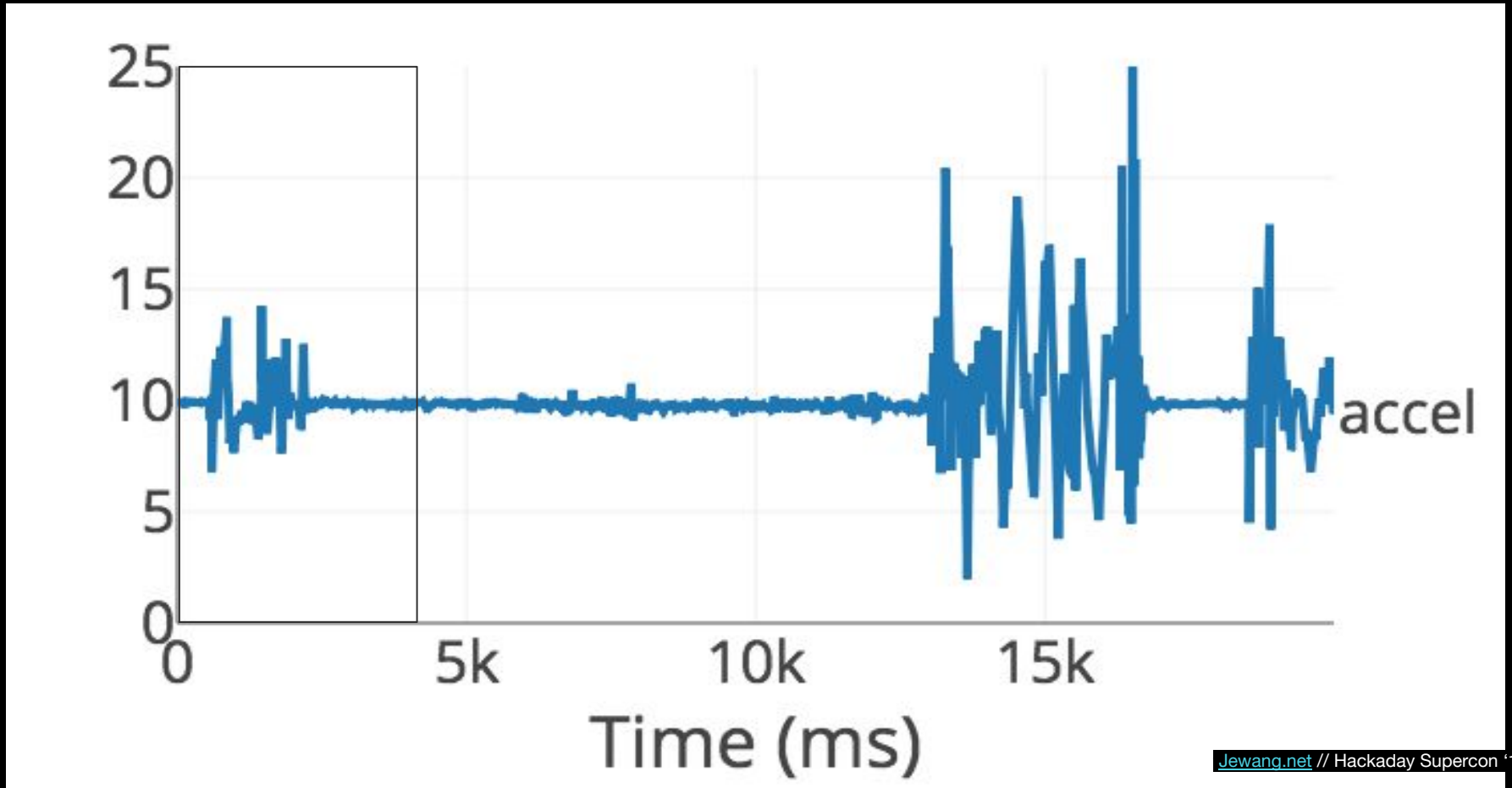
What will the final product look like?



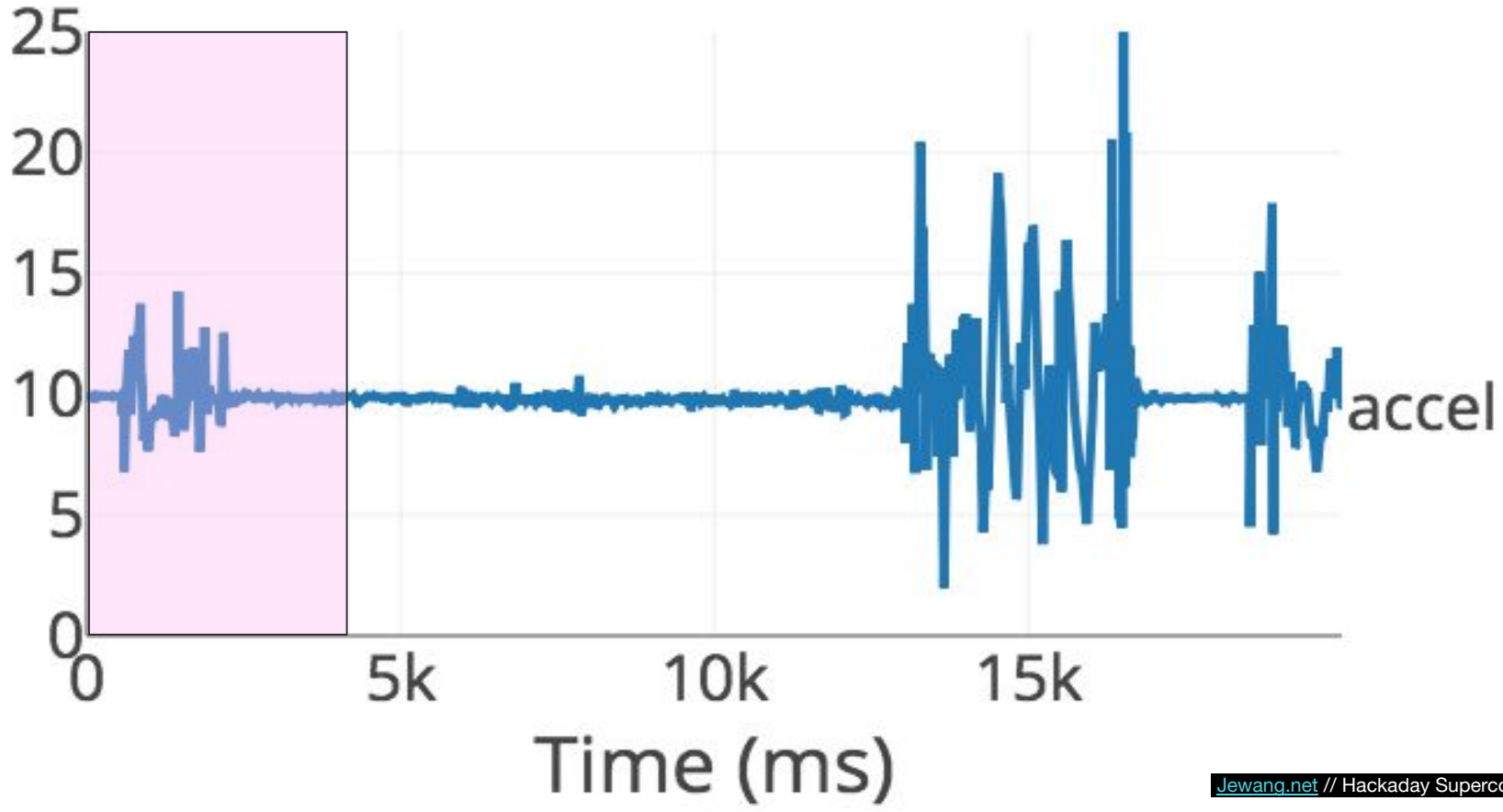
What is a feature?

Summary. Captures what is interesting.

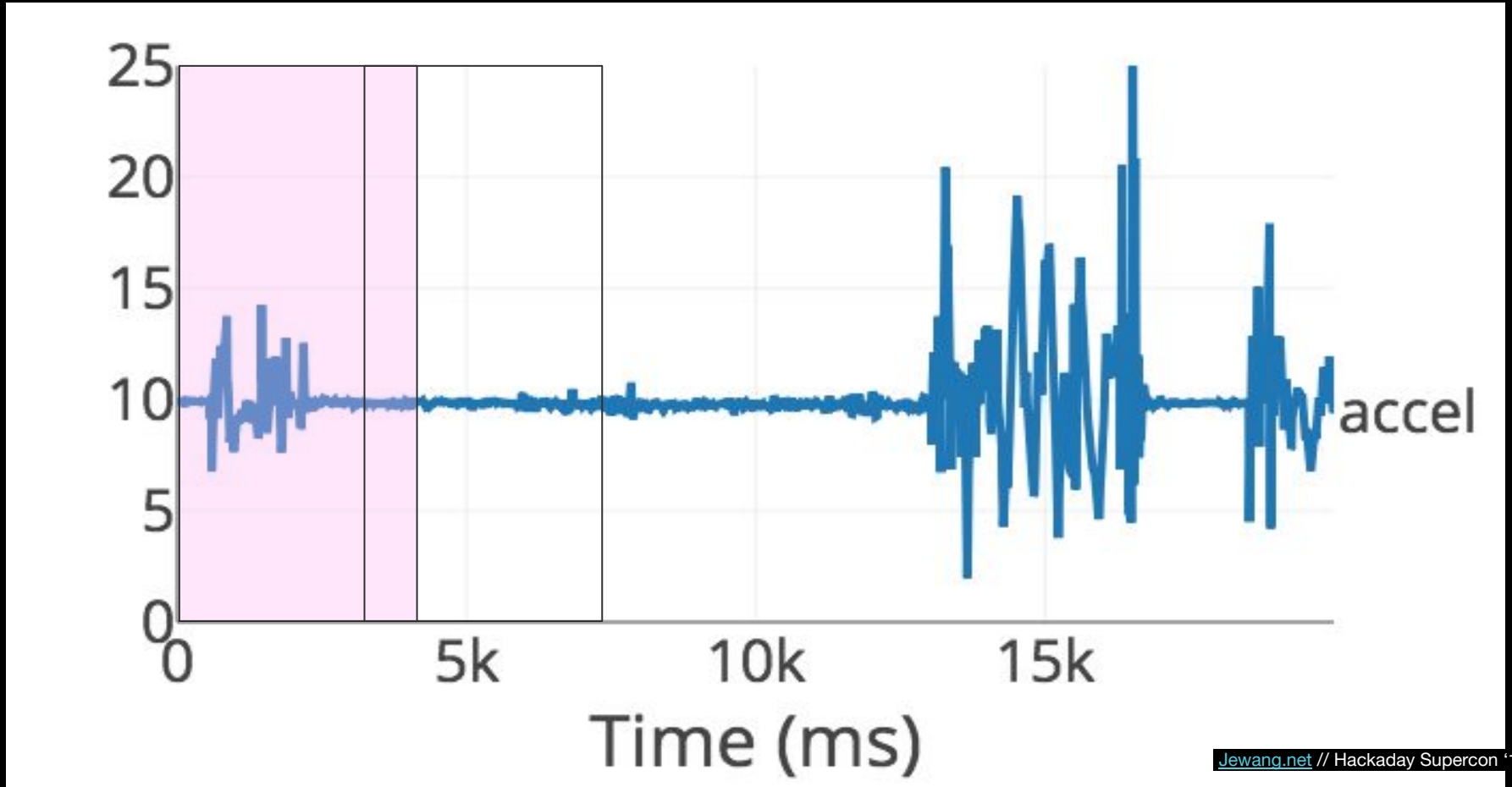
Sensor Data



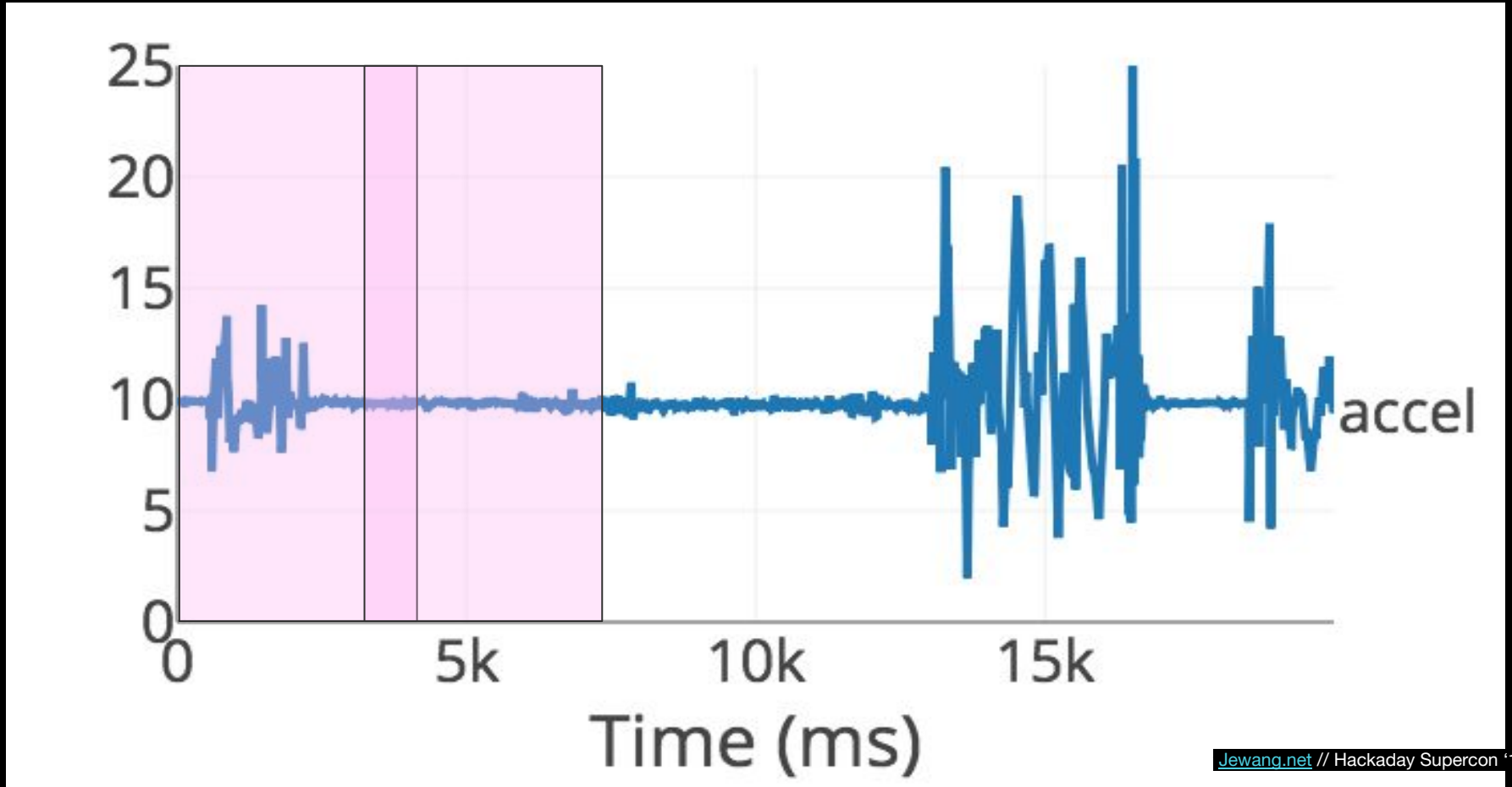
Sensor Data



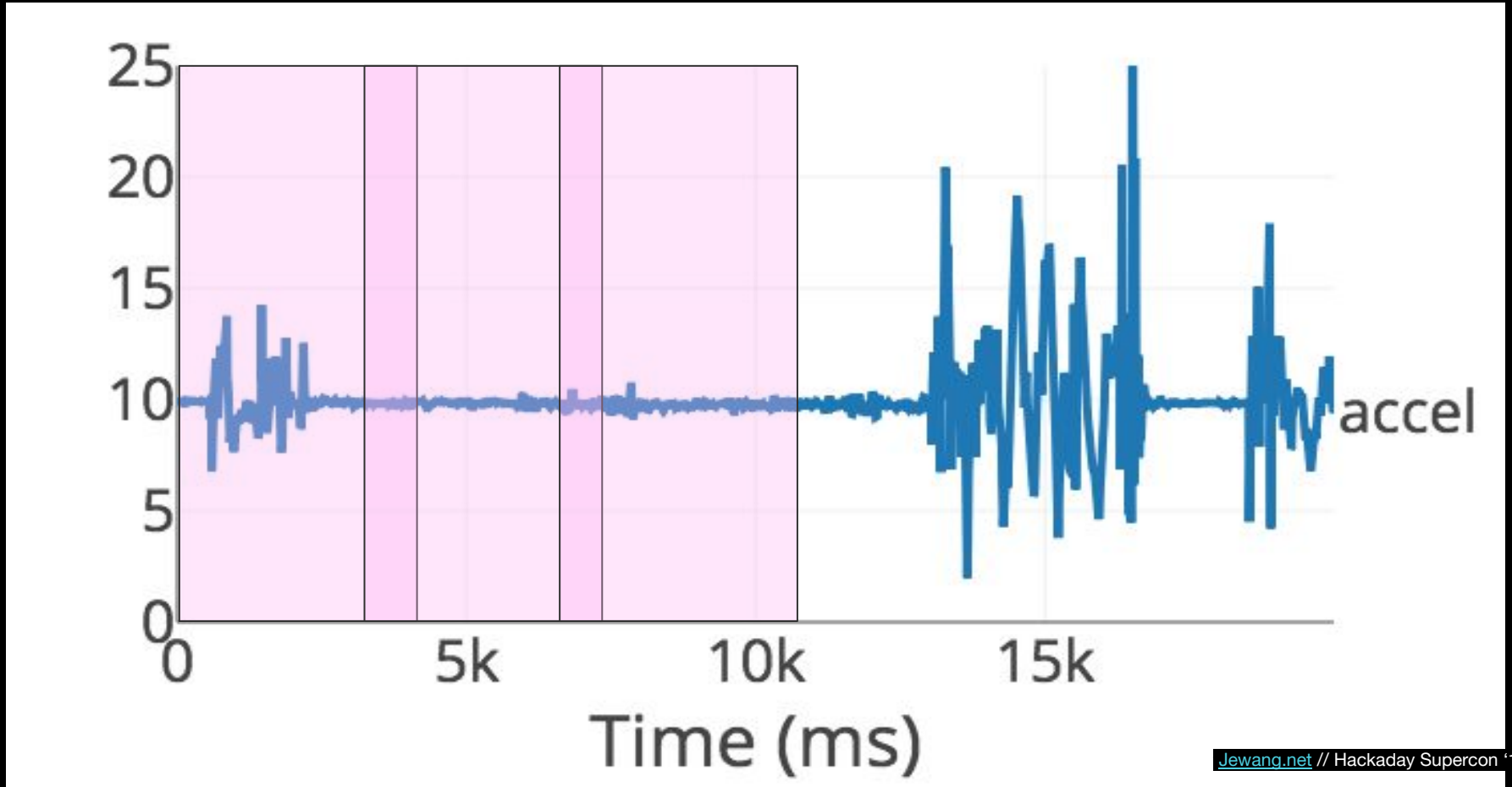
Sensor Data



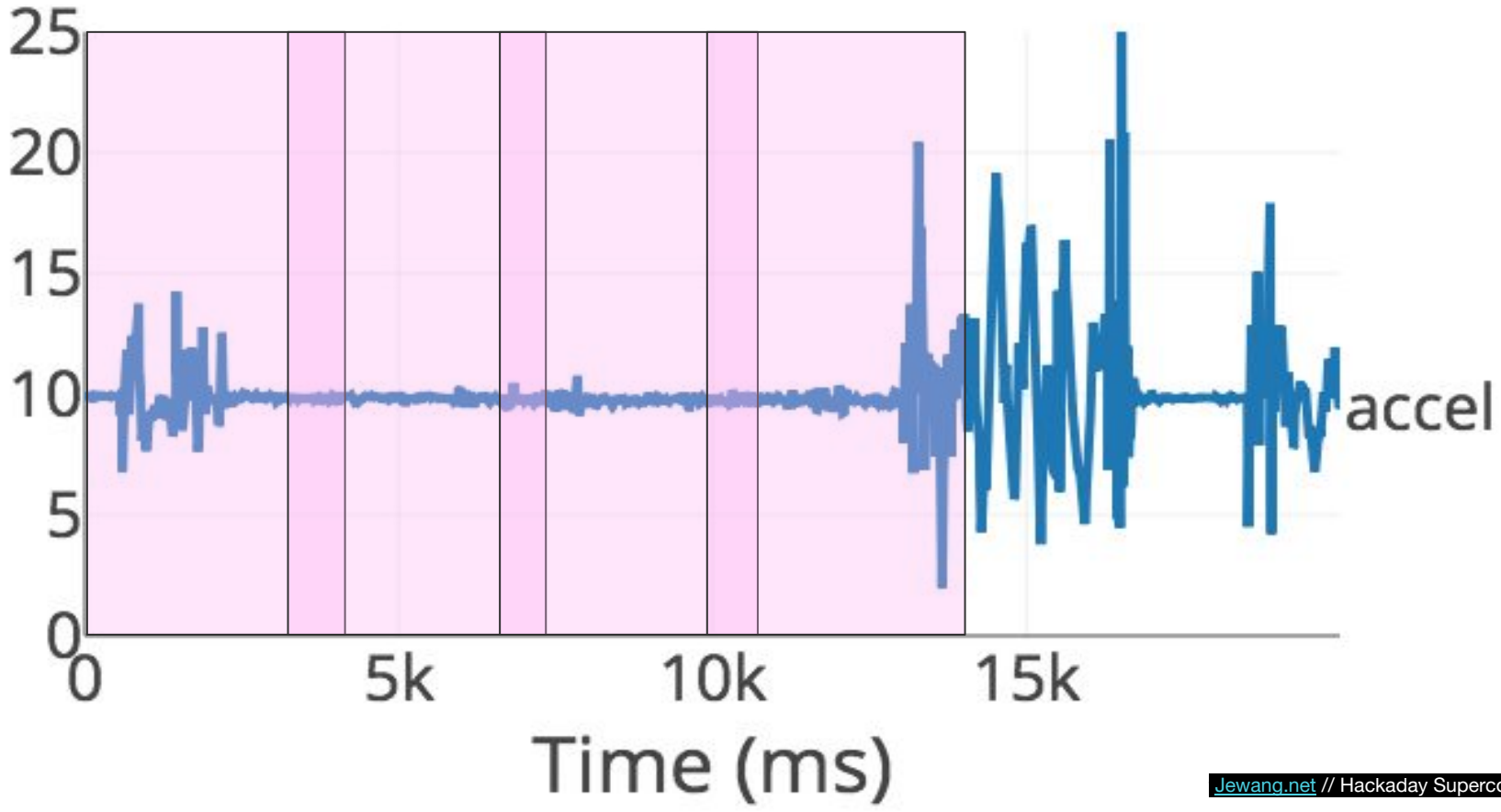
Sensor Data



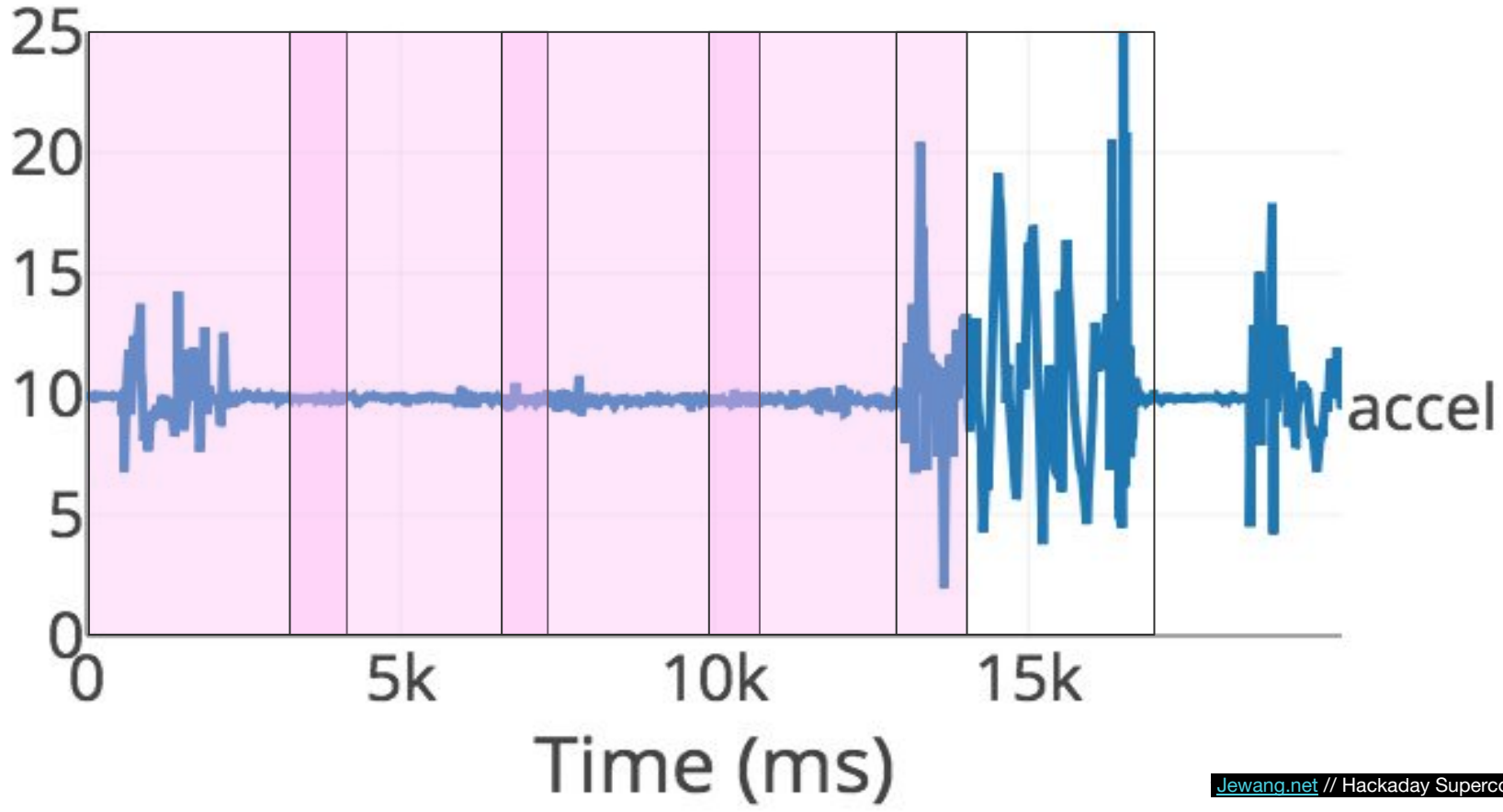
Sensor Data



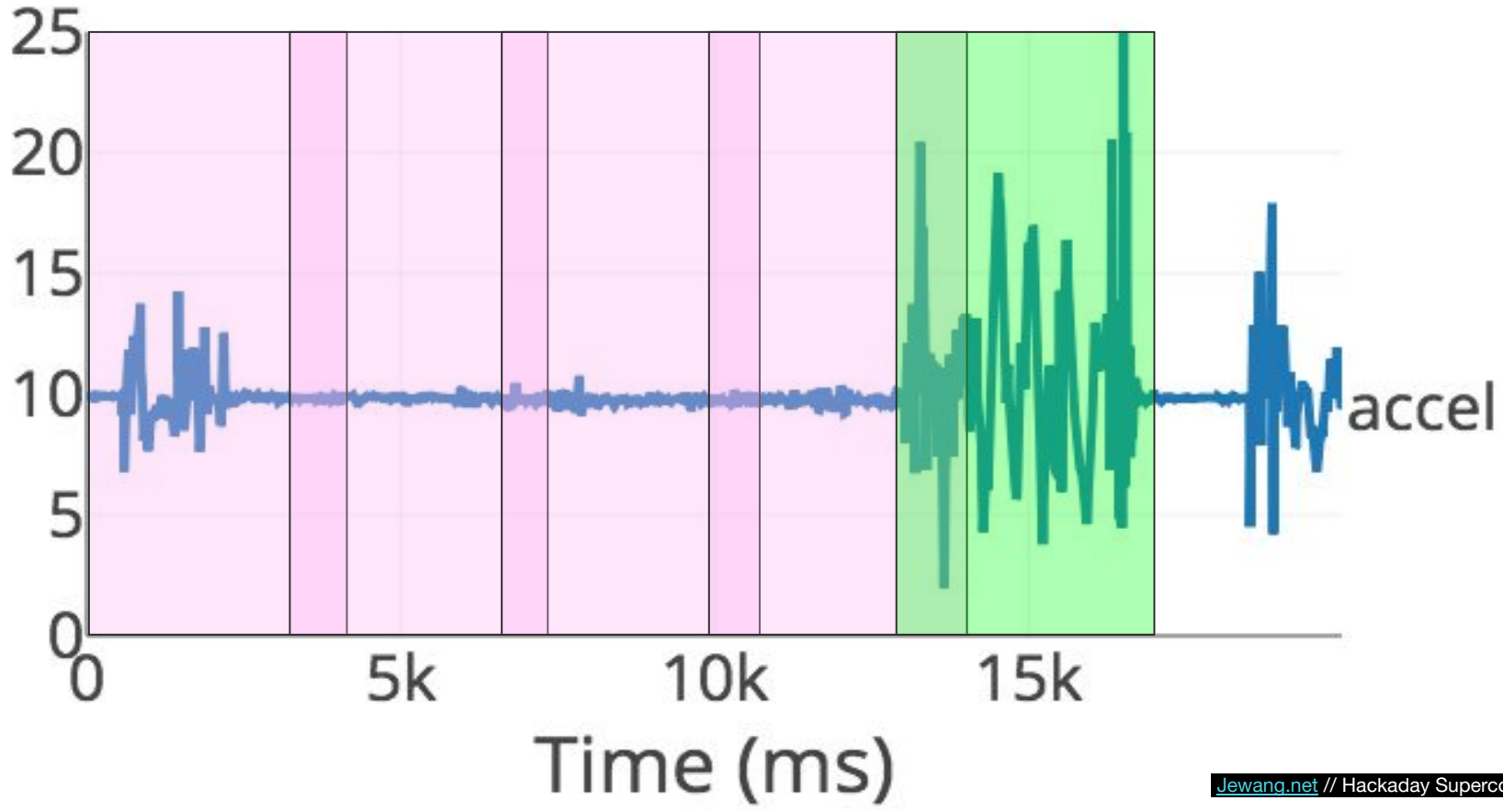
Sensor Data



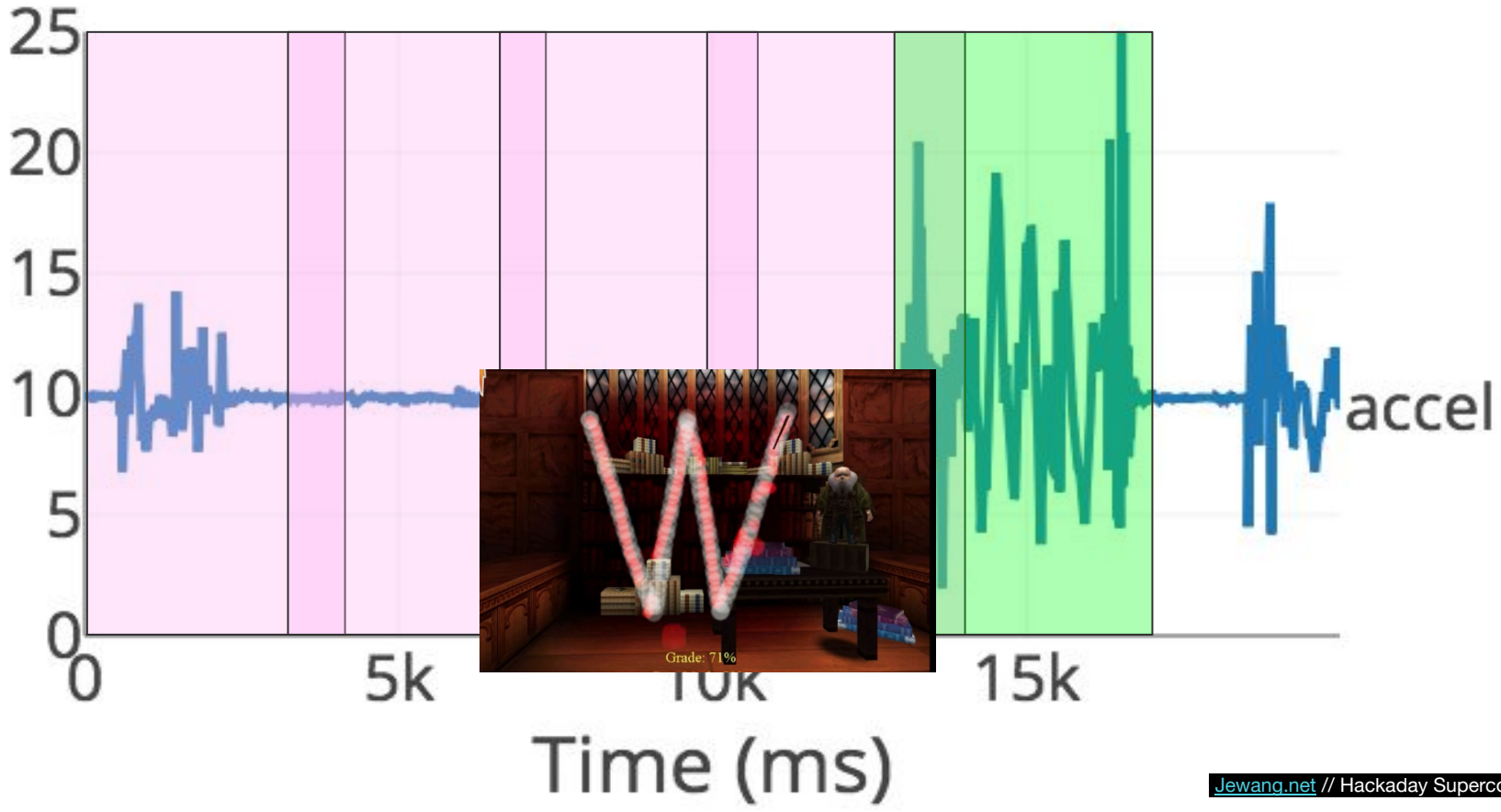
Sensor Data



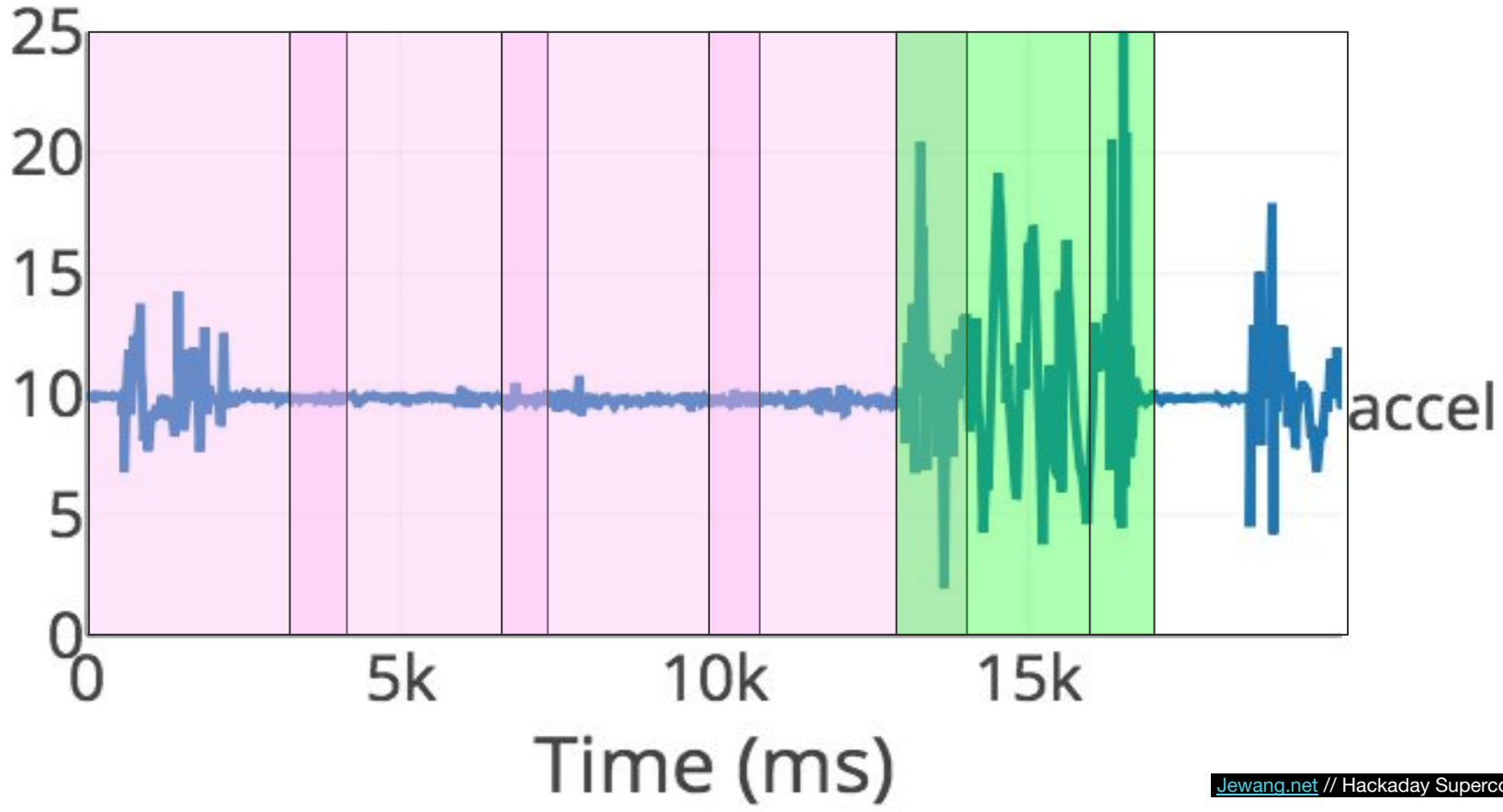
Sensor Data



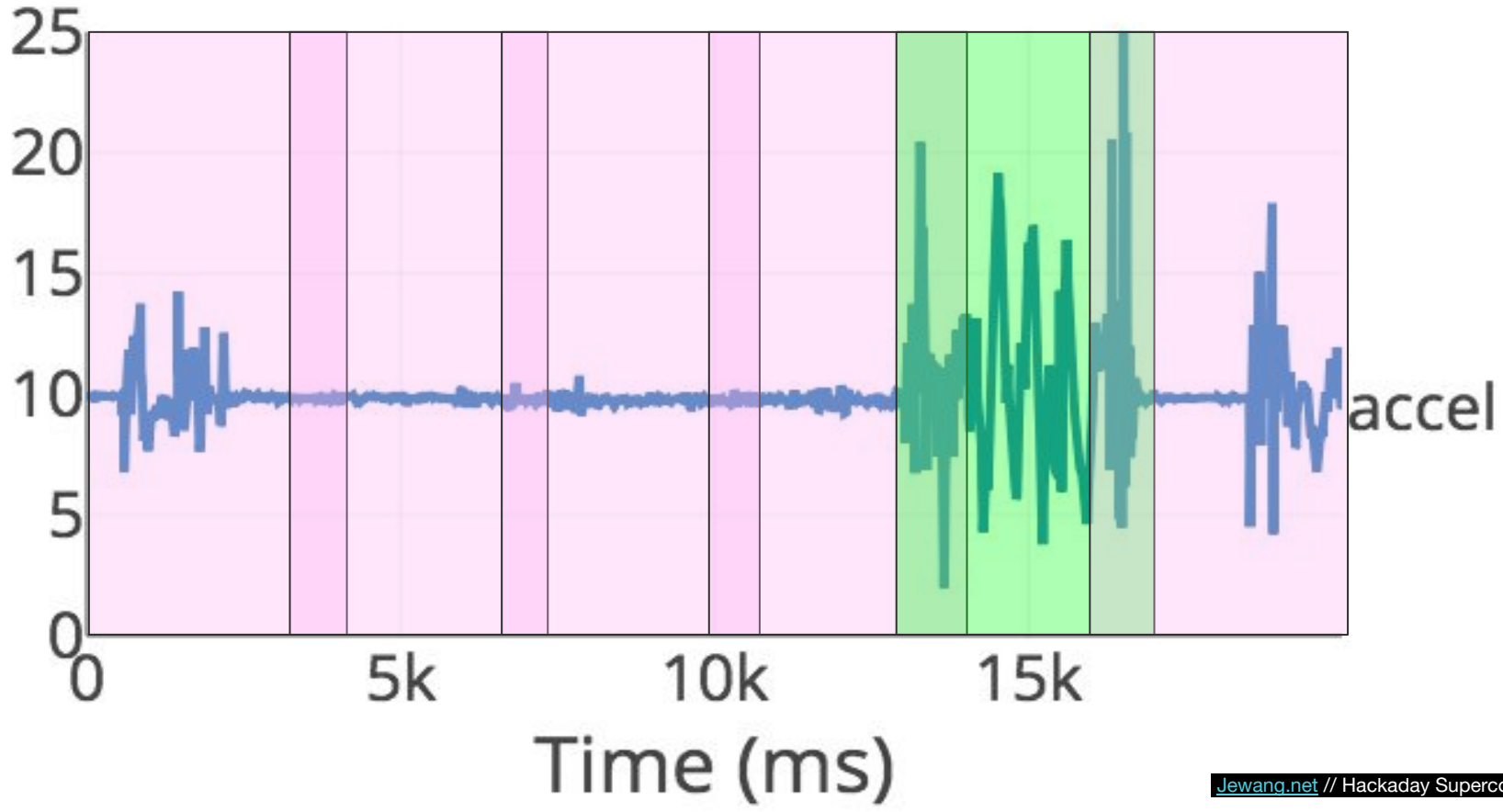
Sensor Data



Sensor Data



Sensor Data



How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

Define & scope the Problem

What are your resources? Time? $?	
Who/when/where will it be used?	
Power, Latency, Memory Req's?	
When is your model good enough?	

Define & scope the Problem

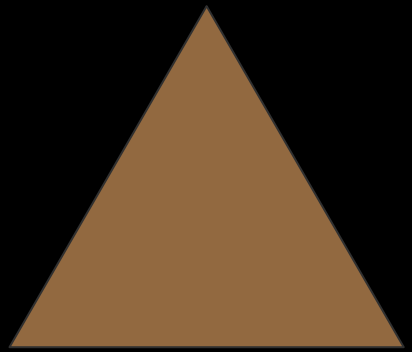
What are your resources? Time? $?	Me. 1 Month! <$200?
Who/when/where will it be used?	
Power, Latency, Memory Req's?	
When is your model good enough?	

Define & scope the Problem

What are your resources? Time? $?	Me. 1 Month! <$200?
Who/when/where will it be used?	Me. Halloween evening! Outdoors, in my neighborhood.
Power, Latency, Memory Req's?	
When is your model good enough?	

Machine Learning on an Embedded System: Pick 2

Accuracy

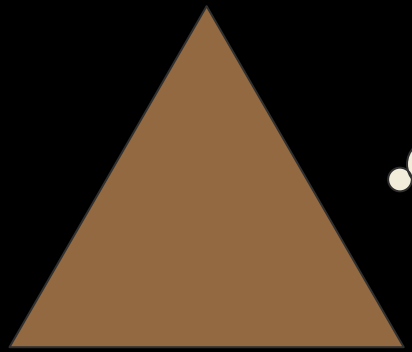


Memory-efficient
(Model Size)

Real-time
Fast Detection
(Latency)

Machine Learning on an Embedded System: Pick 2

Accuracy



Ship as fast as possible

Memory-efficient (Model Size)

Real-time Fast Detection (Latency)

Define & scope the Problem

What are your resources? Time? $?	Me. 1 Month! <$200?
Who/when/where will it be used?	Me. Halloween evening! Outdoors, in my neighborhood.
Power, Latency, Memory Req's?	
When is your model good enough?	

Define & scope the Problem

What are your resources? Time? $?	Me. 1 Month! <$200?
Who/when/where will it be used?	Me. Halloween evening! Outdoors, in my neighborhood.
Power, Latency, Memory Req's?	Don't worry about it for now. Ship as fast as possible.
When is your model good enough?	

Define & scope the Problem

What are your resources? Time? $?	Me. 1 Month! <$200?
Who/when/where will it be used?	Me. Halloween evening! Outdoors, in my neighborhood.
Power, Latency, Memory Req's?	Don't worry about it for now. Ship as fast as possible.
When is your model good enough?	When Halloween happens When kids are happy :)

Define & scope the hardware

What sensors?	
What is the basic architecture?	

Common sensors for gesture recognition

- **Camera?**

Common sensors for gesture recognition

- **Camera?** → No, it's dark outside

Common sensors for gesture recognition

- **Camera?** → No, it's dark outside
- **IR/Wireless Beacons?**

Common sensors for gesture recognition

- **Camera?** → No, it's dark outside
- **IR/Wireless Beacons?** → No, I can't mount these all over my neighborhood

Common sensors for gesture recognition

- **Camera?** → No, it's dark outside
- **IR/Wireless Beacons?** → No, I can't mount these all over my neighborhood
- **Sonar / Audio?**

Common sensors for gesture recognition

- **Camera?** → No, it's dark outside
- **IR/Wireless Beacons?** → No, I can't mount these all over my neighborhood
- **Sonar / Audio?** → No, the environment will be noisy because I'll be outside! Walking around!

Common sensors for gesture recognition

- **Camera?** → No, it's dark outside
- **IR/Wireless Beacons?** → No, I can't mount these all over my neighborhood
- **Sonar / Audio?** → No, the environment will be noisy because I'll be outside! Walking around!
- **Magic E/M Sensing?**

Common sensors for gesture recognition

- **Camera?** → No, it's dark outside
- **IR/Wireless Beacons?** → No, I can't mount these all over my neighborhood
- **Sonar / Audio?** → No, the environment will be noisy because I'll be outside! Walking around!
- **Magic E/M Sensing?** → No, I am not an academic :)

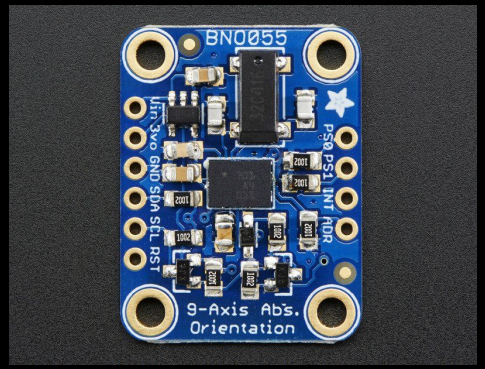
Common sensors for gesture recognition

- **Camera?** → No, it's dark outside
- **IR/Wireless Beacons?** → No, I can't mount these all over my neighborhood
- **Sonar / Audio?** → No, the environment will be noisy because I'll be outside! Walking around!
- **Magic E/M Sensing?** → No, I am not an academic :)
- **Inertial Measurement Unit (IMU)?**

Common sensors for gesture recognition

- **Camera?** → No, it's dark outside
- **IR/Wireless Beacons?** → No, I can't mount these all over my neighborhood
- **Sonar / Audio?** → No, the environment will be noisy because I'll be outside! Walking around!
- **Magic E/M Sensing?** → No, I am not an academic :)
- **Inertial Measurement Unit (IMU)?** → Yes!

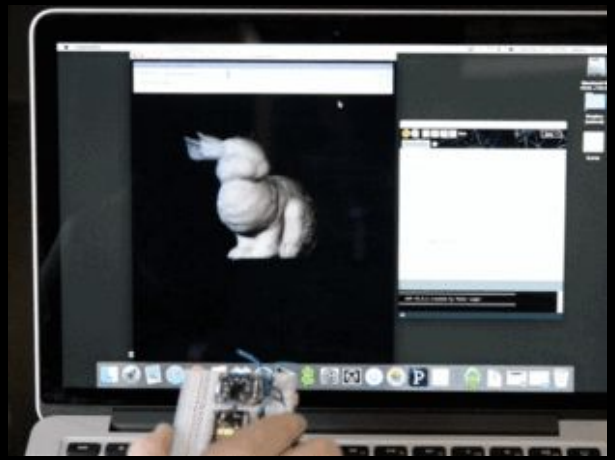
What is an inertial measurement unit (IMU)?

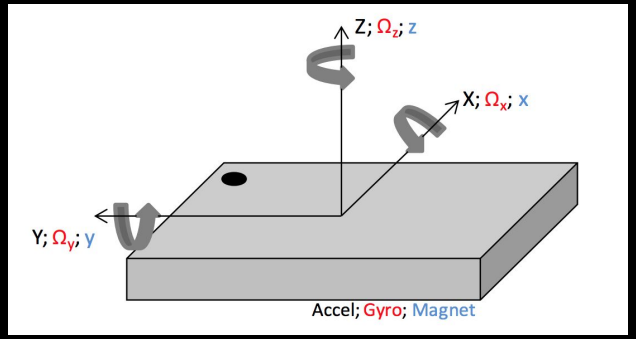


✧ Orientation! ✧

Accelerometer + Gyroscope + Magnetometer

3 Axis

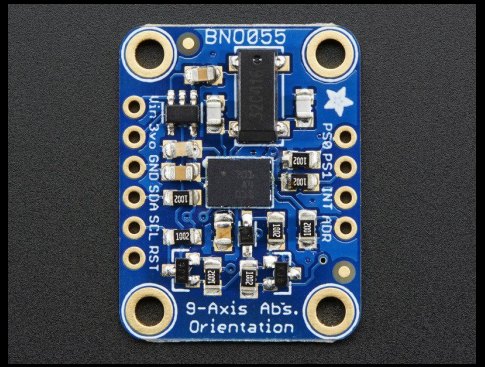




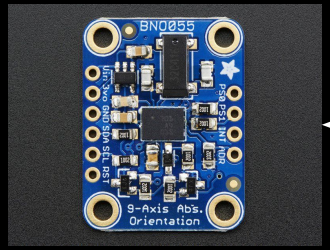
Define & scope the hardware

What sensors?	
What is the basic architecture?	

Define & scope the hardware

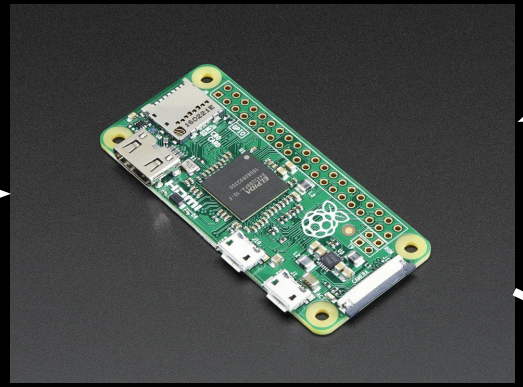
What sensors?	IMU (Bosch BNO055) 
What is the basic architecture?	

Parts!



BNO055
IMU

UART



Raspi Zero W

USB Power

Phone battery

USB Audio

Speaker

Glue this to a cosplay wand.

Parts

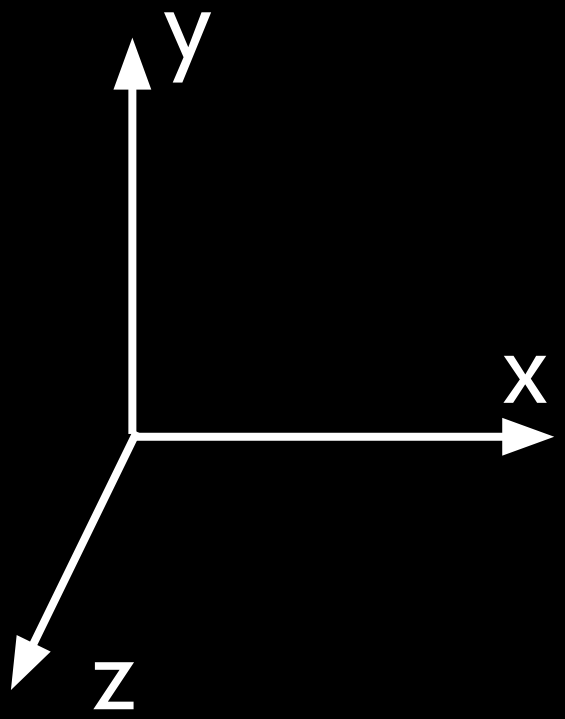
IMU

Raspi Zero W

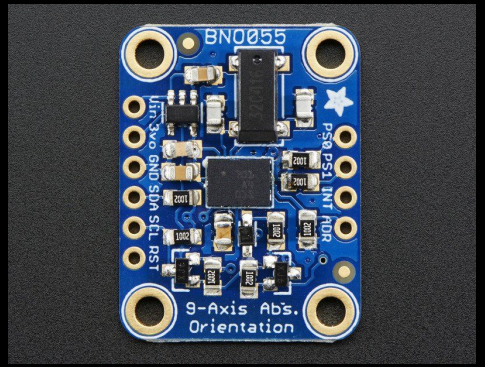
Wand

Battery

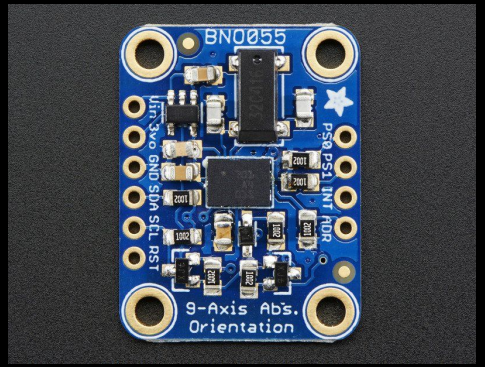
Speaker goes up your sleeve ;)


y
x
z

Define & scope the hardware

What sensors?	IMU (Bosch BNO055) 
What is the basic architecture?	

Define & scope the hardware

What sensors?	IMU (Bosch BNO055) 
What is the basic architecture?	Raspi Zero W glued to a wand. Embedded Linux.

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

What algorithm should I use?

	Deep learning!	Signal processing	Traditional Machine Learning
Domain knowledge	Not much	Lots!	Medium
Amount of data	Lots!	Not as much	Medium
Easy to debug	Harder	Easier	Medium

What algorithm should I use?

	Deep learning!	Signal processing	***Traditional Machine Learning***
Domain knowledge	Not much	Lots!	Medium
Amount of data	Lots!	Not as much	Medium
Easy to debug	Harder	Easier	Medium

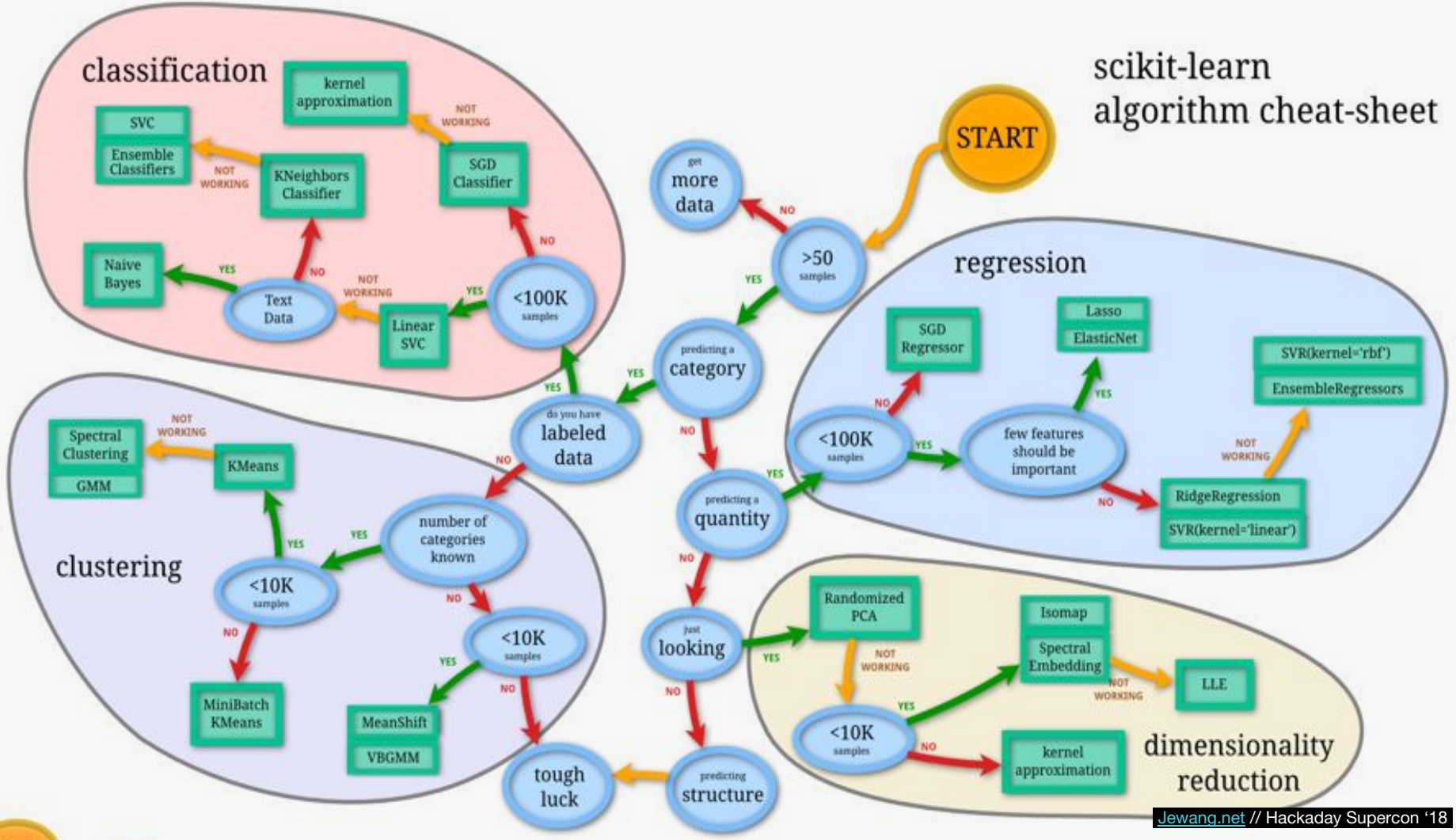
What algorithm should I use?

	Deep learning	Processing	***Traditional Machine Learning***
Domain knowledge			Medium
Amount of data			Medium
Easy to debug	Hard	Easier	Medium

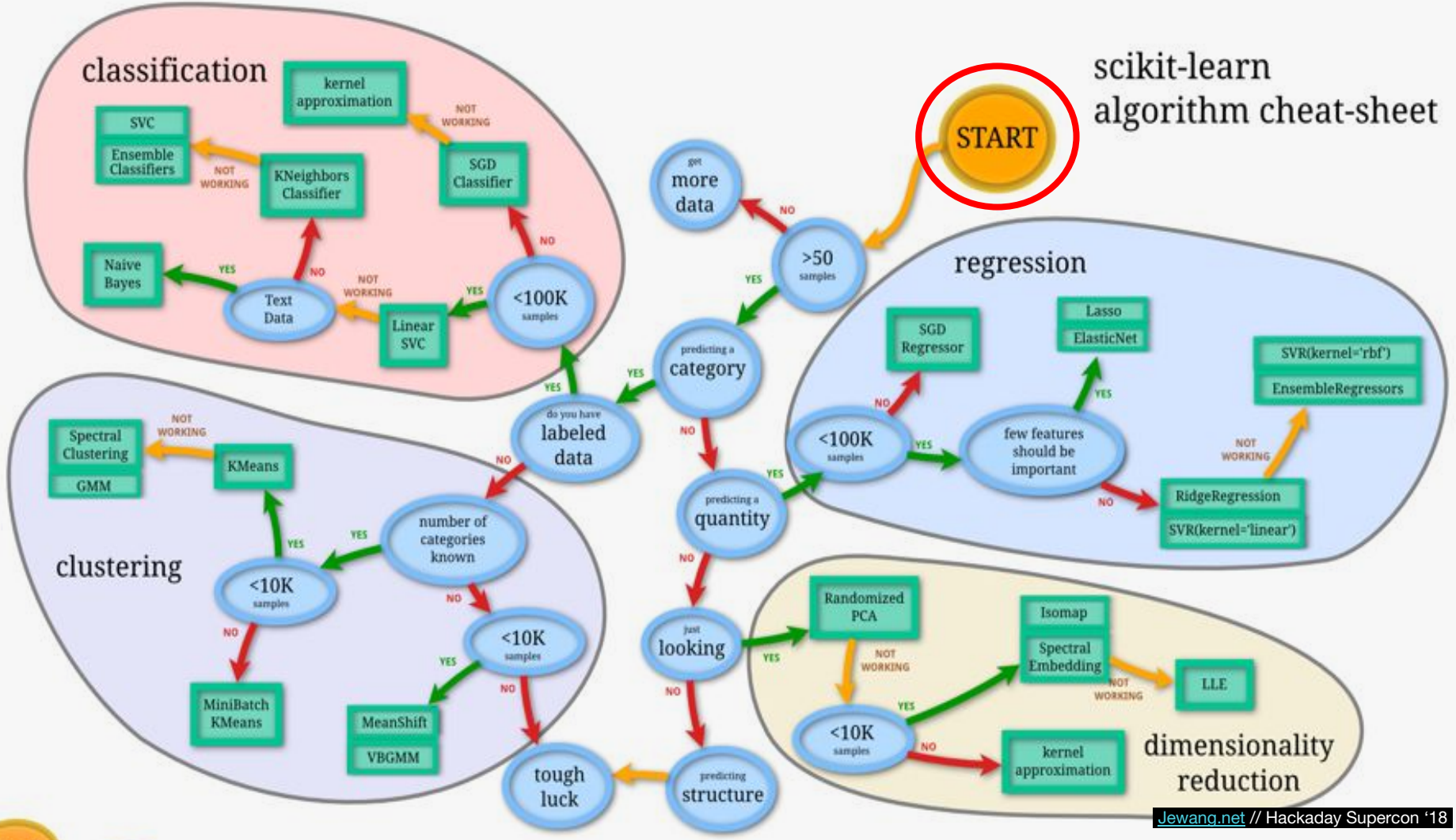
1. Ship your MVP
2. Get more users
3. More users = more data
4. Switch to deep learning

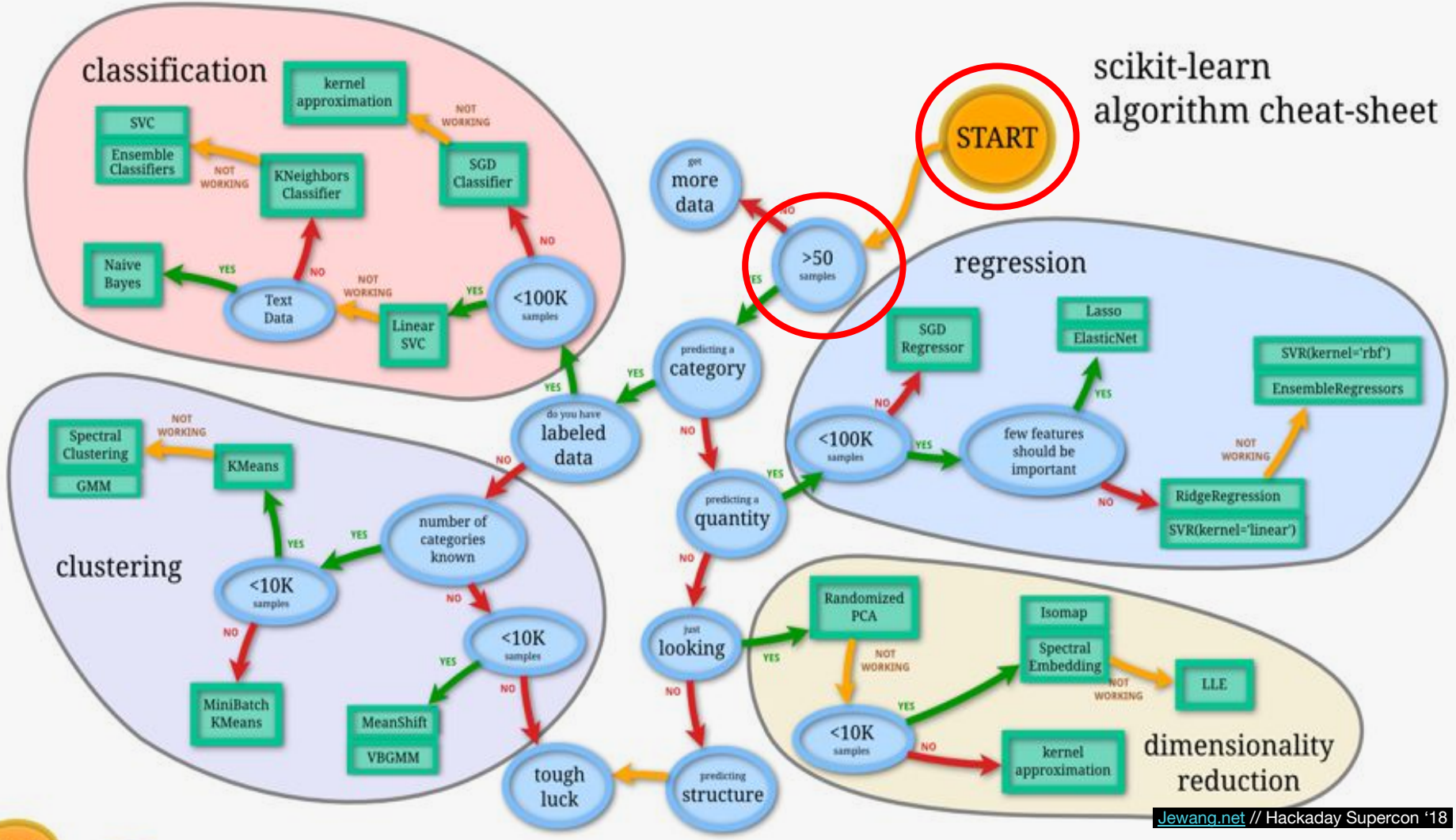
Software Used

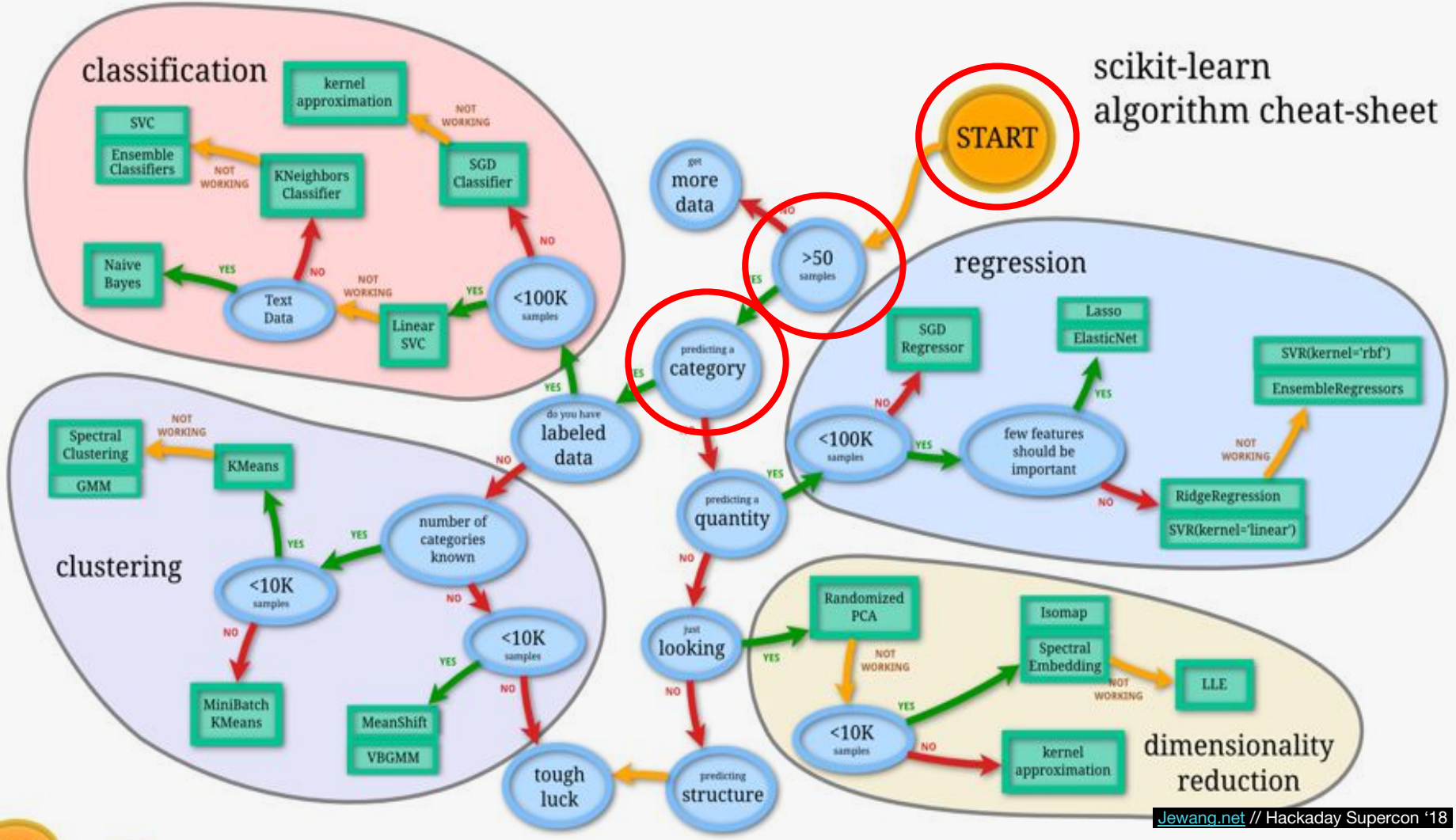
- **Language:** Python3
- **Numerical Libraries:** Pandas, Numpy
- **Data Notebook:** Jupyter Notebook
- **Data Visualization:** Plot.ly
- **Machine learning library:** scikit-learn

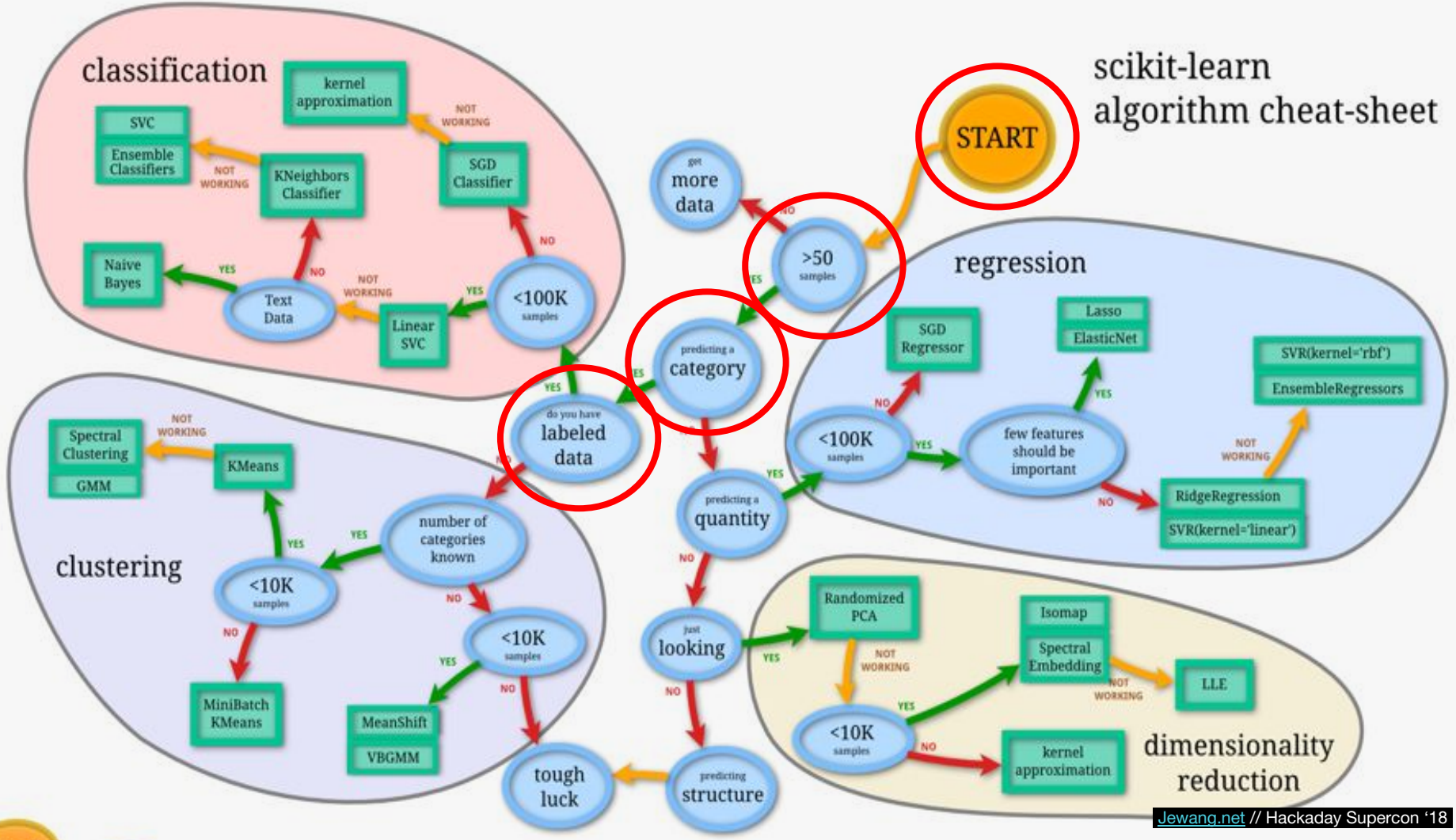


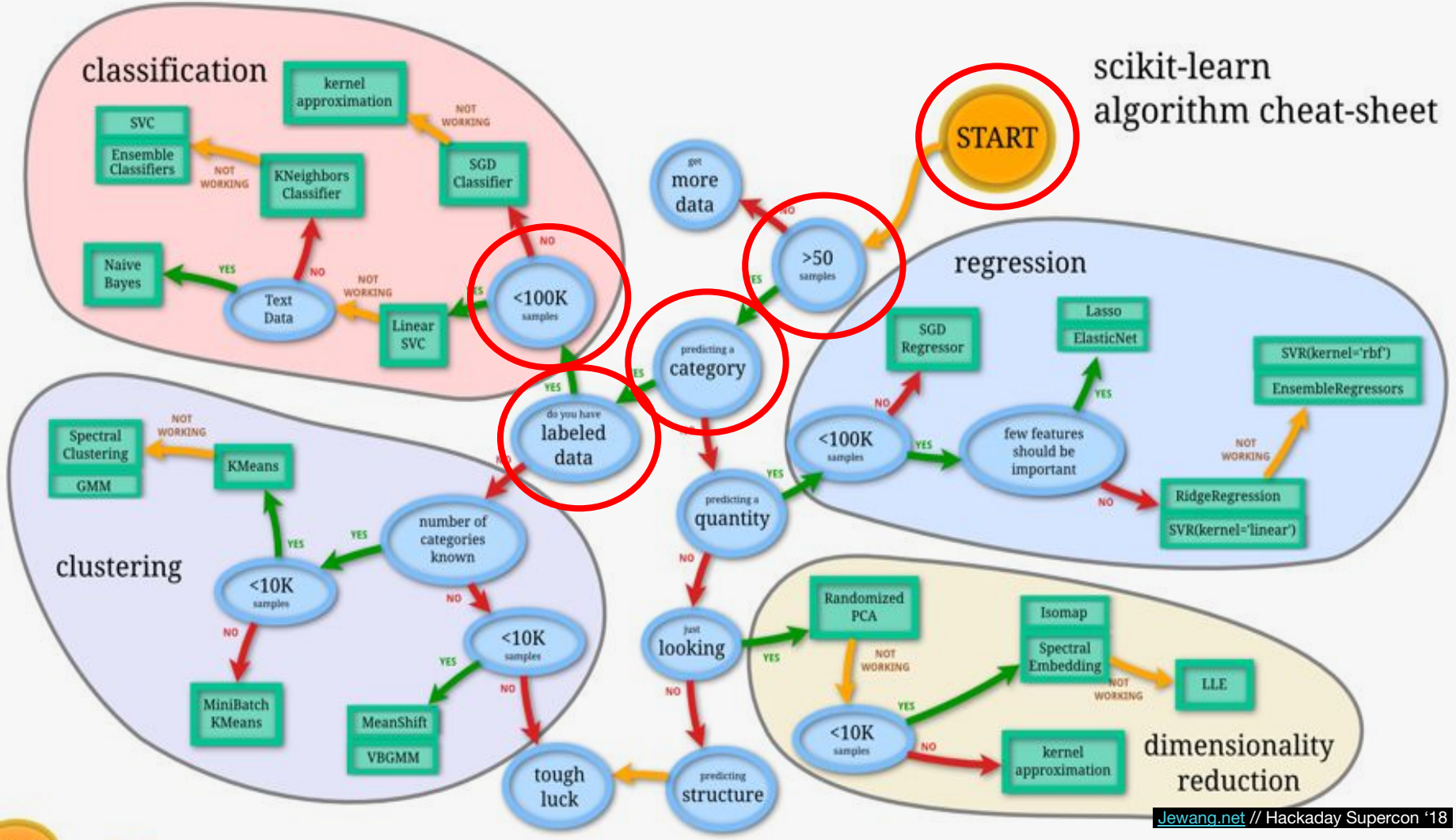
Jewang.net // Hackaday Supercon '18

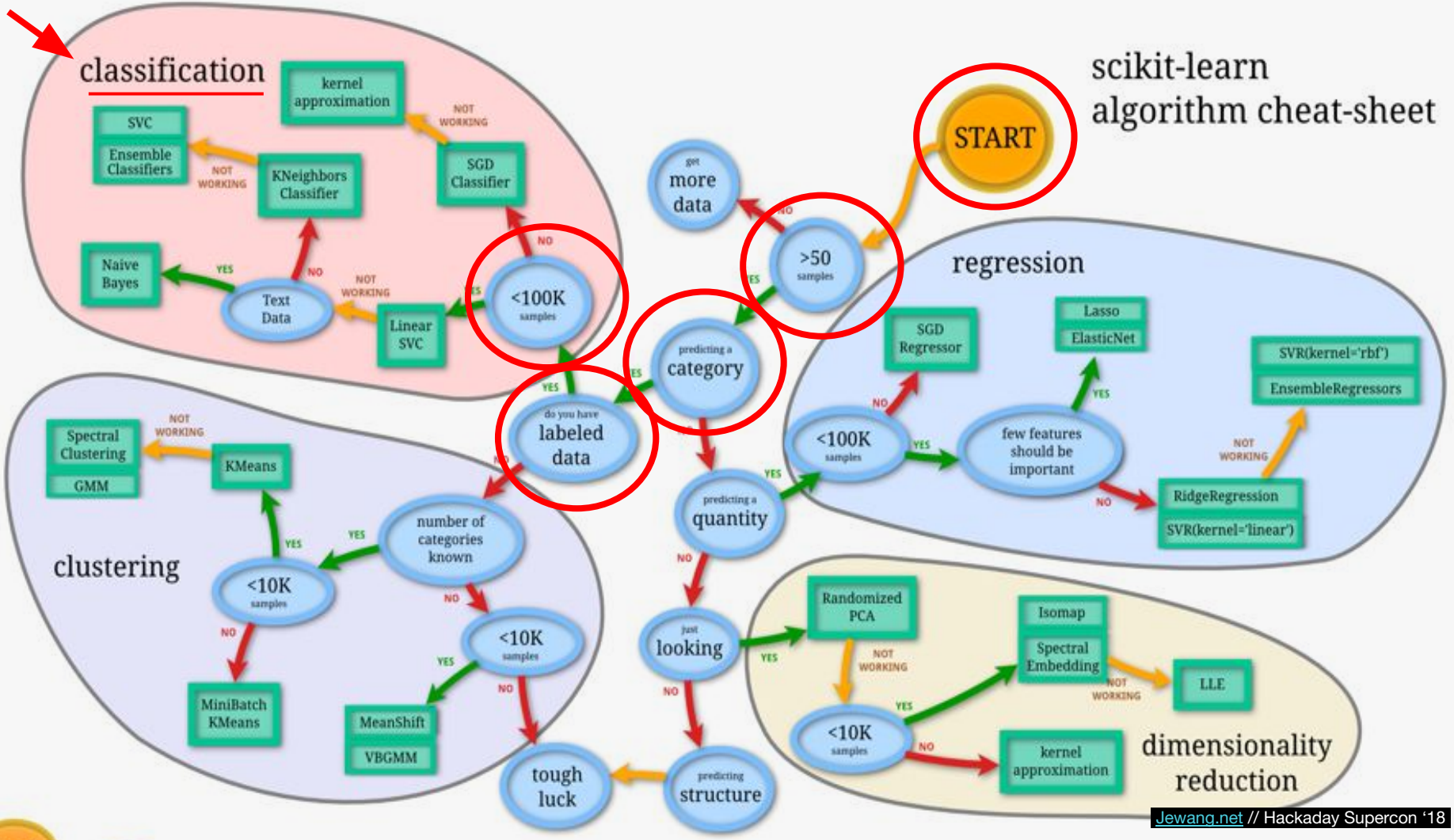

scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
NO
Text Data
NOT WORKING
YES
Linear SVC
<100K samples
get more data
NO
>50 samples
YES
predicting a category
YES
YES
do you have labeled data
NO
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K samples
YES
few features should be important
YES
NOT WORKING
NO
RidgeRegression
SVR(kernel='linear')
Spectral Clustering
GMM
NOT WORKING
KMeans
NO
YES
number of categories known
YES
clustering
<10K samples
NO
<10K samples
NO
YES
MiniBatch KMeans
MeanShift
VBGMM
NO
tough luck
predicting quantity
YES
NO
just looking
YES
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
predicting structure

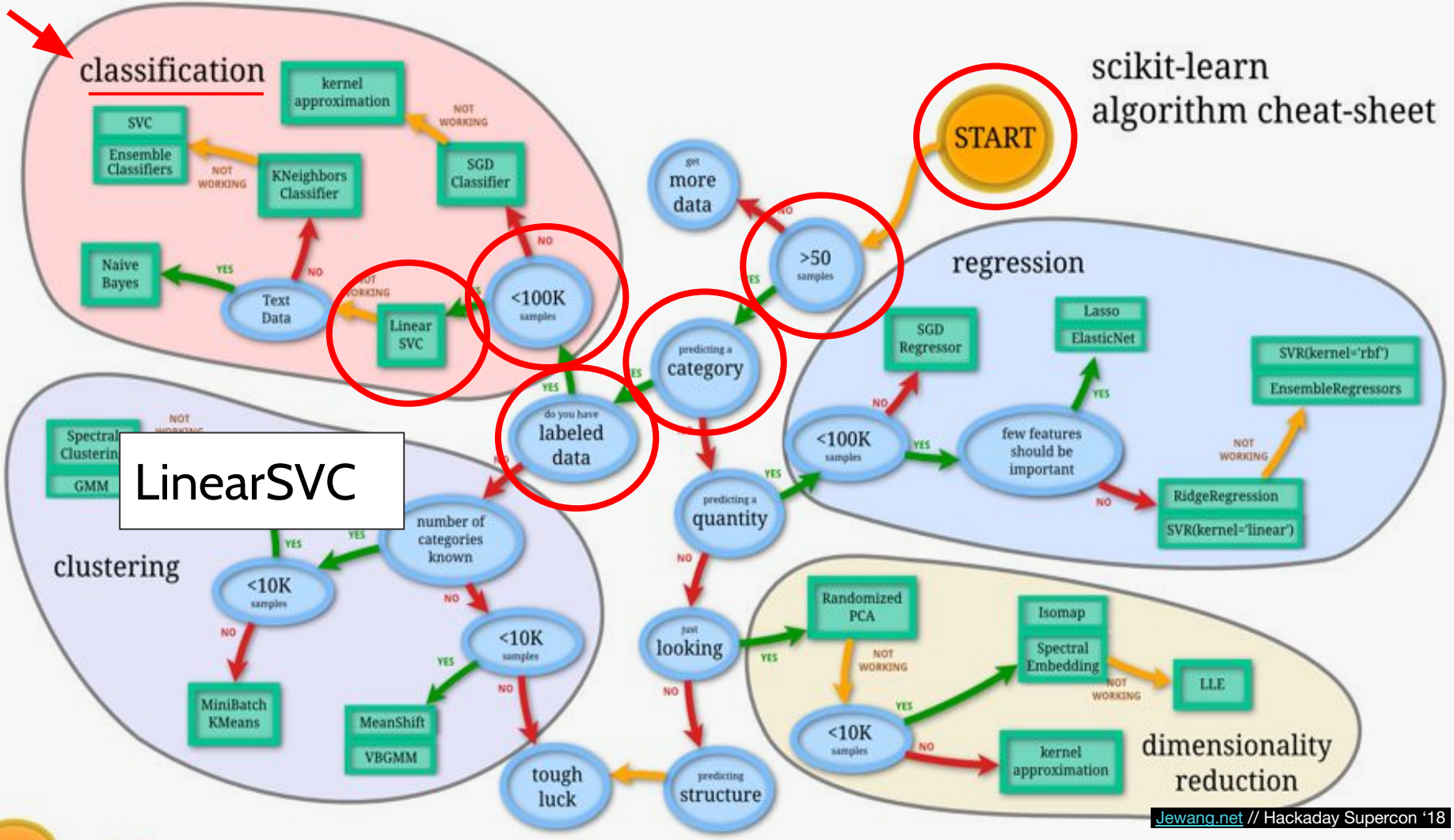

scikit-learn
algorithm cheat-sheet
START
>50
samples
get
more
data
NO
YES
predicting a
category
YES
NO
do you have
labeled
data
YES
NO
predicting a
quantity
YES
NO
just
looking
YES
NO
predicting
structure
tough
luck
classification
kernel
approximation
NOT
WORKING
SVC
Ensemble
Classifiers
NOT
WORKING
KNeighbors
Classifier
SGD
Classifier
NO
Naive
Bayes
YES
NO
Text
Data
NOT
WORKING
Linear
SVC
YES
<100K
samples
regression
SGD
Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
YES
<100K
samples
YES
few features
should be
important
NOT
WORKING
NO
RidgeRegression
SVR(kernel='linear')
clustering
Spectral
Clustering
GMM
NOT
WORKING
KMeans
YES
YES
number of
categories
known
<10K
samples
NO
NO
<10K
samples
YES
NO
MiniBatch
KMeans
MeanShift
VBGMM
dimensionality
reduction
Randomized
PCA
NOT
WORKING
Isomap
Spectral
Embedding
NOT
WORKING
LLE
YES
<10K
samples
NO
kernel
approximation


scikit-learn
algorithm cheat-sheet
START
>50
samples
get
more
data
NO
YES
predicting a
category
do you have
labeled
data
predicting a
quantity
just
looking
predicting
structure
tough
luck
classification
kernel
approximation
NOT
WORKING
SVC
Ensemble
Classifiers
NOT
WORKING
KNeighbors
Classifier
SGD
Classifier
NO
Naive
Bayes
YES
NO
Text
Data
NOT
WORKING
Linear
SVC
YES
<100K
samples
regression
SGD
Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K
samples
YES
few features
should be
important
YES
NOT
WORKING
RidgeRegression
SVR(kernel='linear')
NO
clustering
Spectral
Clustering
GMM
NOT
WORKING
KMeans
YES
number of
categories
known
YES
<10K
samples
NO
NO
<10K
samples
YES
NO
MiniBatch
KMeans
MeanShift
VBGMM
dimensionality
reduction
Randomized
PCA
NOT
WORKING
Isomap
Spectral
Embedding
NOT
WORKING
LLE
YES
<10K
samples
NO
kernel
approximation


scikit-learn
algorithm cheat-sheet
START
>50
samples
NO
get
more
data
YES
predicting a
category
YES
do you have
labeled
data
YES
predicting a
quantity
NO
just
looking
NO
predicting
structure
tough
luck
classification
kernel
approximation
NOT
WORKING
SVC
Ensemble
Classifiers
NOT
WORKING
KNeighbors
Classifier
SGD
Classifier
NO
Naive
Bayes
YES
NO
Text
Data
NOT
WORKING
Linear
SVC
YES
<100K
samples
regression
SGD
Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
YES
<100K
samples
YES
few features
should be
important
NOT
WORKING
NO
RidgeRegression
SVR(kernel='linear')
clustering
Spectral
Clustering
GMM
NOT
WORKING
KMeans
YES
YES
number of
categories
known
<10K
samples
NO
NO
<10K
samples
YES
NO
MiniBatch
KMeans
MeanShift
VBGMM
dimensionality
reduction
Randomized
PCA
NOT
WORKING
YES
Isomap
Spectral
Embedding
NOT
WORKING
LLE
YES
<10K
samples
NO
kernel
approximation


scikit-learn
algorithm cheat-sheet
START
>50
samples
get
more
data
predicting a
category
do you have
labeled
data
predicting a
quantity
just
looking
predicting
structure
tough
luck
classification
kernel
approximation
SVC
Ensemble
Classifiers
KNeighbors
Classifier
SGD
Classifier
Naive
Bayes
Text
Data
Linear
SVC
<100K
samples
NOT
WORKING
YES
NO
clustering
Spectral
Clustering
GMM
KMeans
number of
categories
known
<10K
samples
MiniBatch
KMeans
MeanShift
VBGMM
regression
SGD
Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
<100K
samples
few features
should be
important
RidgeRegression
SVR(kernel='linear')
dimensionality
reduction
Randomized
PCA
Isomap
Spectral
Embedding
LLE
<10K
samples
kernel
approximation


scikit-learn
algorithm cheat-sheet
START
>50
samples
get
more
data
NO
YES
predicting a
category
do you have
labeled
data
predicting a
quantity
just
looking
predicting
structure
tough
luck
classification
kernel
approximation
NOT
WORKING
SVC
Ensemble
Classifiers
KNeighbors
Classifier
SGD
Classifier
Naive
Bayes
Text
Data
Linear
SVC
<100K
samples
regression
SGD
Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
<100K
samples
few features
should be
important
RidgeRegression
SVR(kernel='linear')
clustering
Spectral
Clustering
GMM
KMeans
number of
categories
known
<10K
samples
MiniBatch
KMeans
MeanShift
VBGMM
dimensionality
reduction
Randomized
PCA
Isomap
Spectral
Embedding
LLE
<10K
samples
kernel
approximation


scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
Naive Bayes
YES
NO
Text Data
NOT WORKING
Linear SVC
<100K samples
NO
YES
get more data
NO
>50 samples
YES
predicting a category
YES
do you have labeled data
YES
NO
LinearSVC
Spectral Clustering
GMM
NOT WORKING
clustering
number of categories known
YES
YES
<10K samples
NO
NO
<10K samples
YES
NO
MiniBatch KMeans
MeanShift
VBGMM
tough luck
predicting structure
NO
just looking
YES
predicting a quantity
YES
NO
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K samples
YES
few features should be important
YES
NOT WORKING
NO
RidgeRegression
SVR(kernel='linear')
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

Planning data collection

- How much data do you need?
- Look for pre-existing data
- *More data = More* 💰 / 🕒
  Design collection procedure, manage data collectors, QA...

Doing data collection

Time to code!

Lock mechanical design!

Match & record data for all expected use cases!

Who × Posture × When × Where

Record data for all expected use cases!

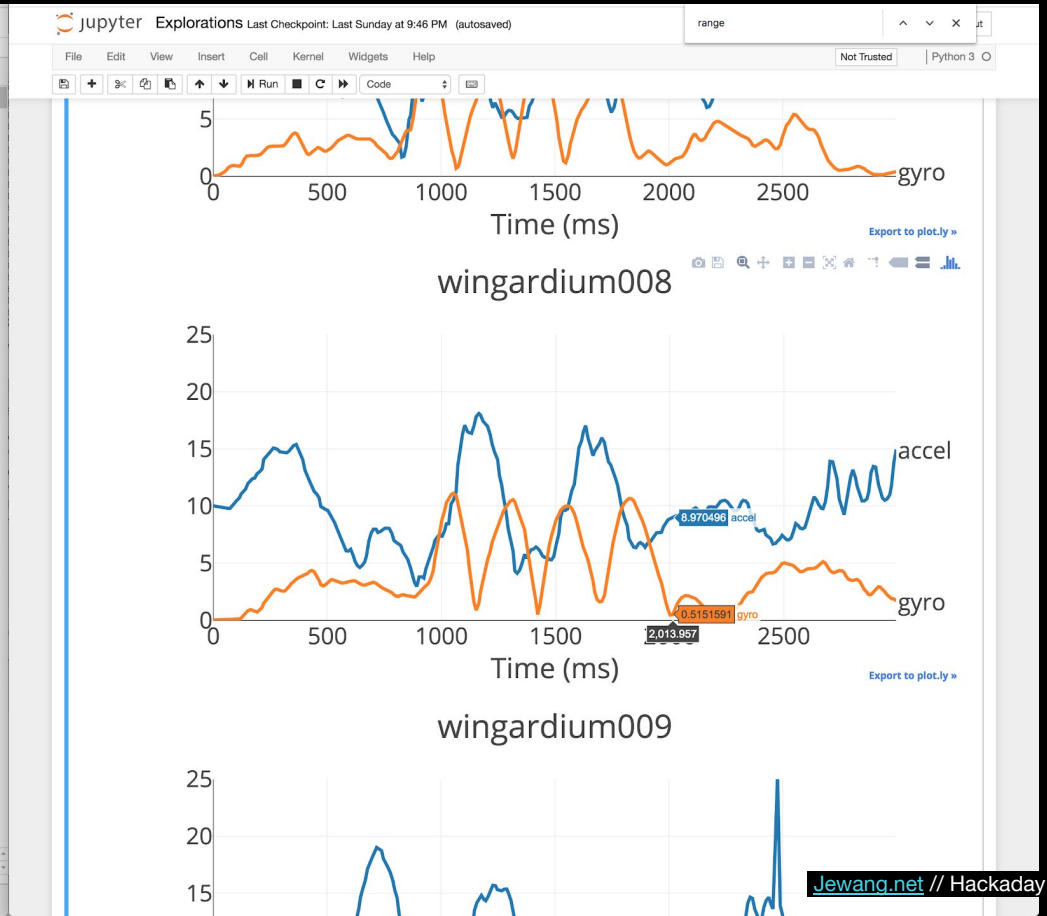
	Wingardium	Flippendo	Negative
Sitting Couch	✓	✓	✓
Sitting Chair	✓	✓	✓
Sitting Floor	✓	✓	✓
Standing	✓	✓	✓
Walking	✓	✓	✓

Record data for all expected use cases!

	Wingardium	Flippendo	**Negative***
Sitting Couch	✓	✓	✓
Sitting Chair	✓	✓	✓
Sitting Floor	✓	✓	✓
Standing	✓	✓	✓
Walking	✓	✓	✓

Data collection is an exercise in diligence

	A	B	C
1	trace		
2	wingardium000	1770	
3	wingardium001	1880	
4	wingardium002	1700	
5	wingardium003	1840	
6	wingardium004	2050	
7	wingardium005	1860	
8	wingardium006	1815	
9	wingardium007	1875	
10	wingardium008		
11	wingardium009		
12	wingardium010		
13	wingardium011		
14	wingardium012		
15	wingardium013		
16	wingardium014		
17	wingardium015		
18	wingardium016		
19	wingardium017		
20	wingardium018		
21	wingardium019		
22	wingardium020		
23	wingardium021		



Final data collected for magic wand

collect_data.py → CSV

286 1.5s traces

98 Wingardium

99 Flippendo

89 Negative

~7 minutes of data

257,000+ data points

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

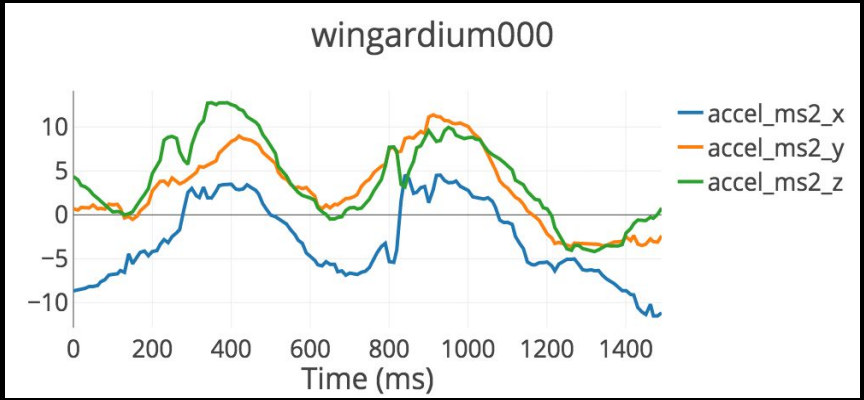
Train model & iterate

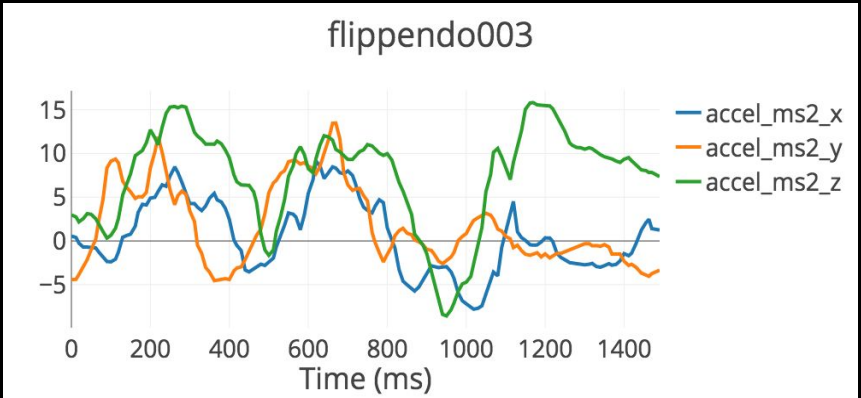
- Explorations.ipynb
- Mostly data cleaning :P

Wingardium

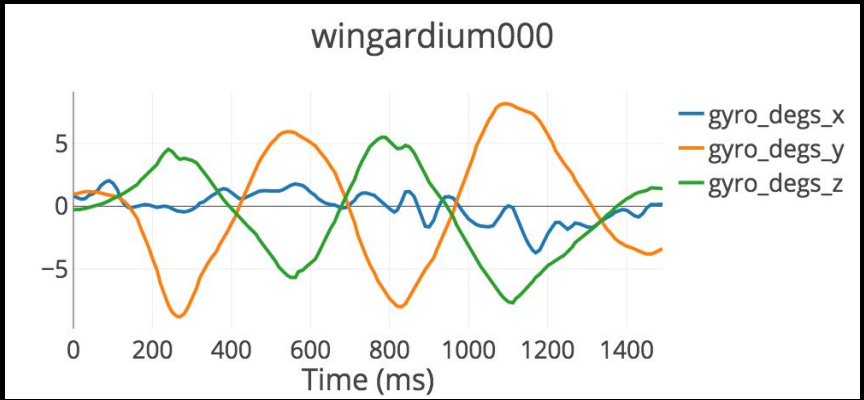
Flippendo

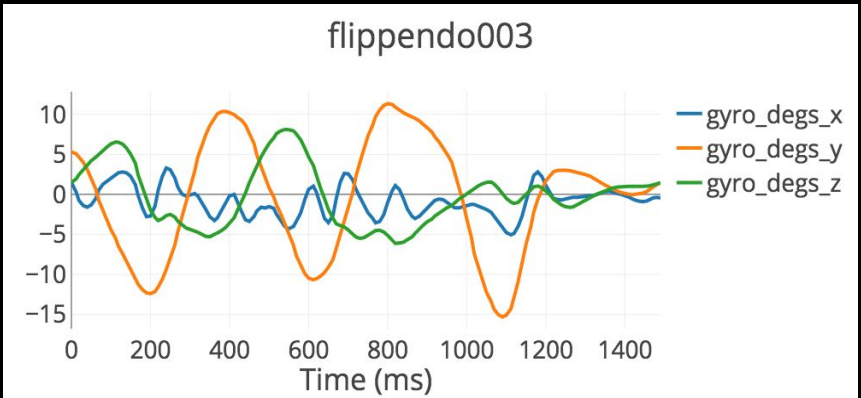
Accel





Gyro





Features

- max_accel
- min_accel
- Range_accel
- mean_accel
- std_accel

- max_gyro
- min_gyro
- range_gyro
- mean_gyro
- std_gyro

Train model & iterate

- Iteration #1 → Didn't work well. 102 traces.

Train model & iterate

- Iteration #1 → Didn't work well. 102 traces.
  - Clean Data! Collect new data!

Train model & iterate

- Iteration #1 → Didn't work well. 102 traces.
  - Clean Data! Collect new data!
- Iteration #2 → Didn't work well. 167 traces.

Train model & iterate

- Iteration #1 → Didn't work well. 102 traces.
  - Clean Data! Collect new data!
- Iteration #2 → Didn't work well. 167 traces.
  - Clean Data! Collect new data!

Train model & iterate

- Iteration #1 → Didn't work well. 102 traces.
  - Clean Data! Collect new data!
- Iteration #2 → Didn't work well. 167 traces.
  - Clean Data! Collect new data!
- Iteration #3: → Worked OK, distinguish gestures. 286 traces.

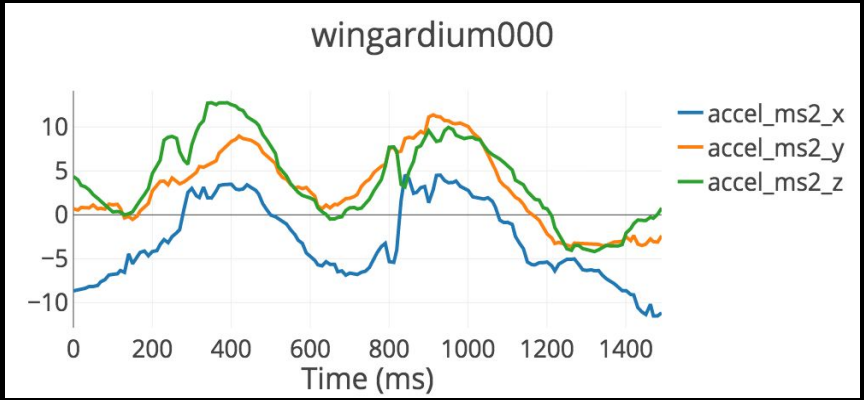
Train model & iterate

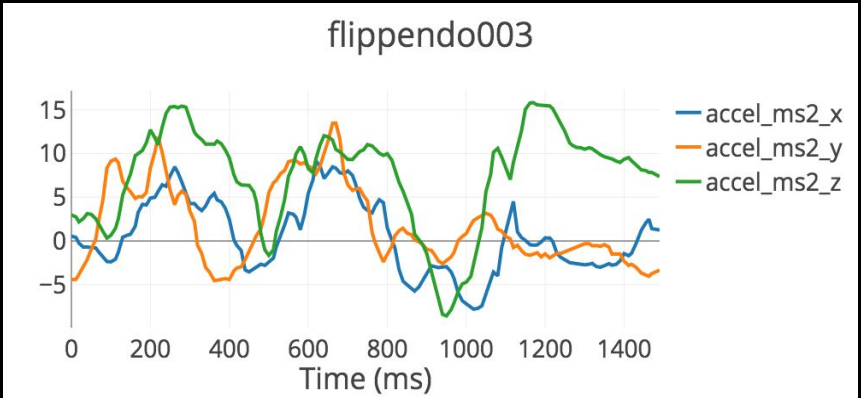
- Iteration #1 → Didn't work well. 102 traces.
  - Clean Data! Collect new data!
- Iteration #2 → Didn't work well. 167 traces.
  - Clean Data! Collect new data!
- Iteration #3: → Worked OK, distinguish gestures. 286 traces.
  - Feature Design

Wingardium

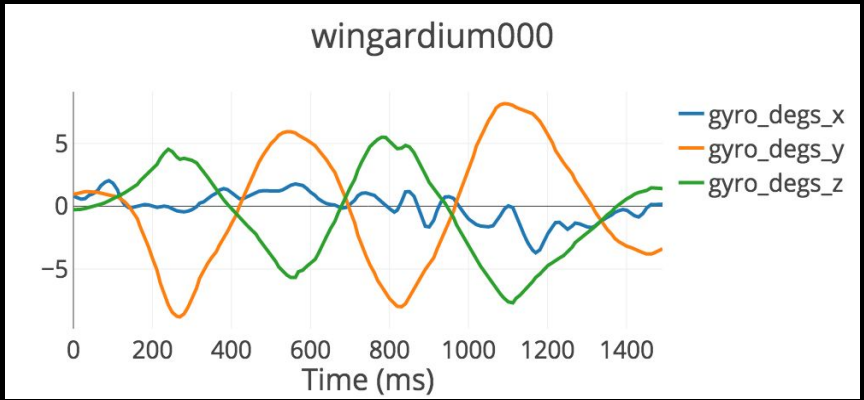
Flippendo

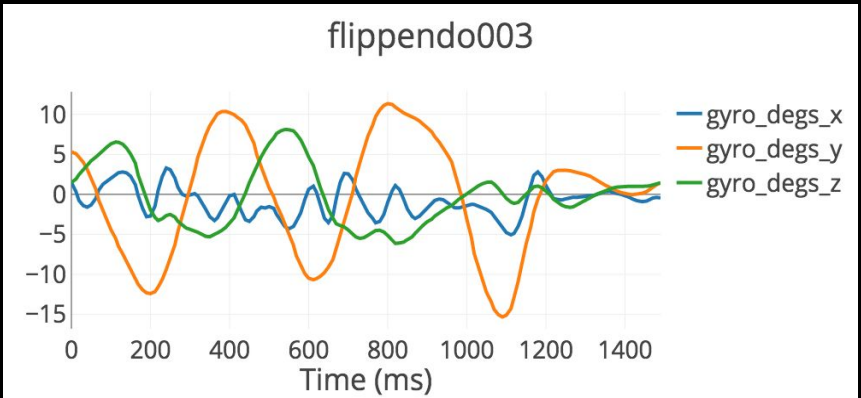
Accel





Gyro

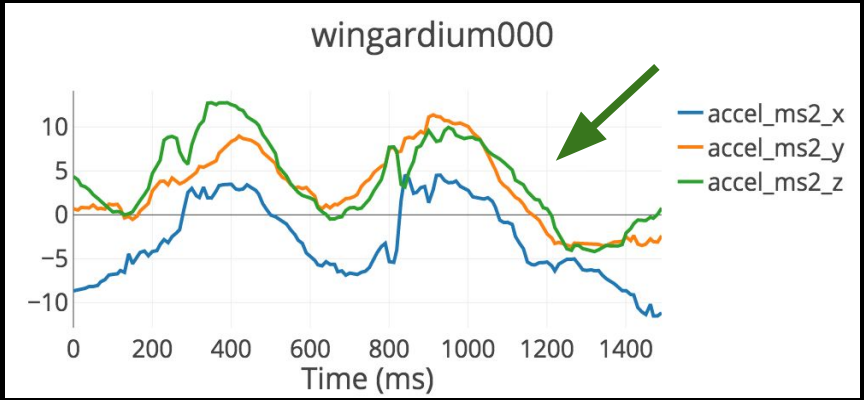


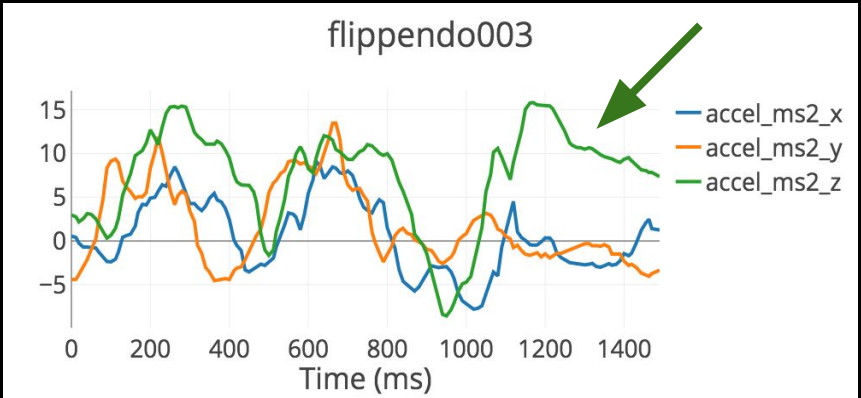


Wingardium

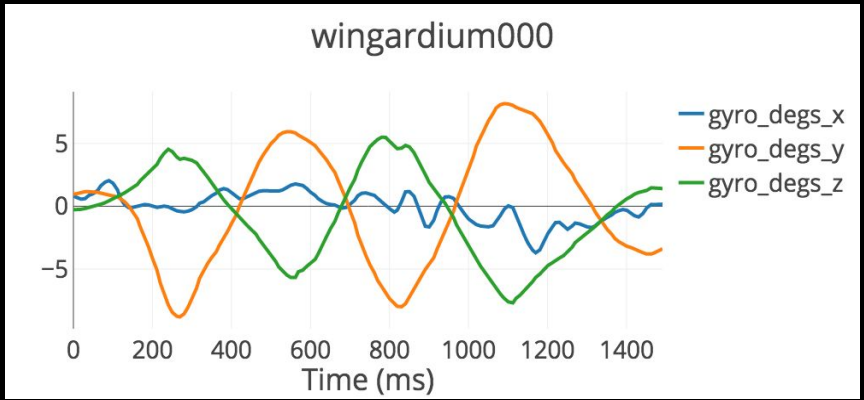
Flippendo

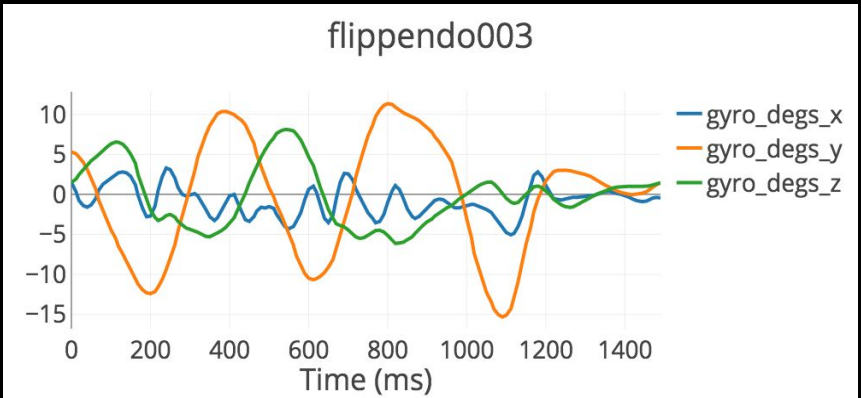
Accel





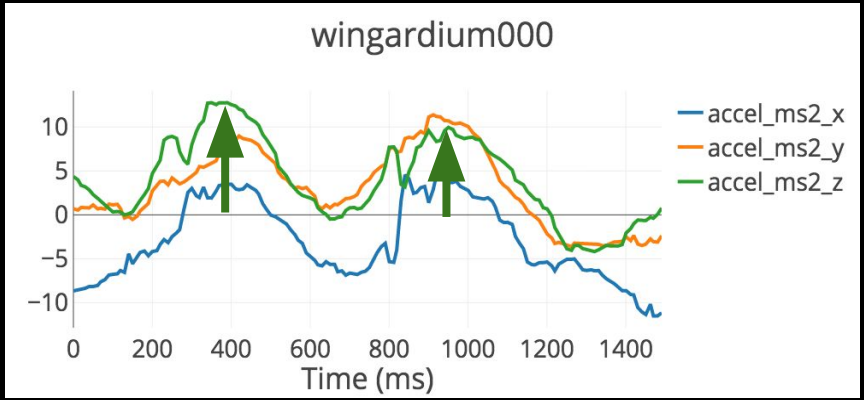
Gyro

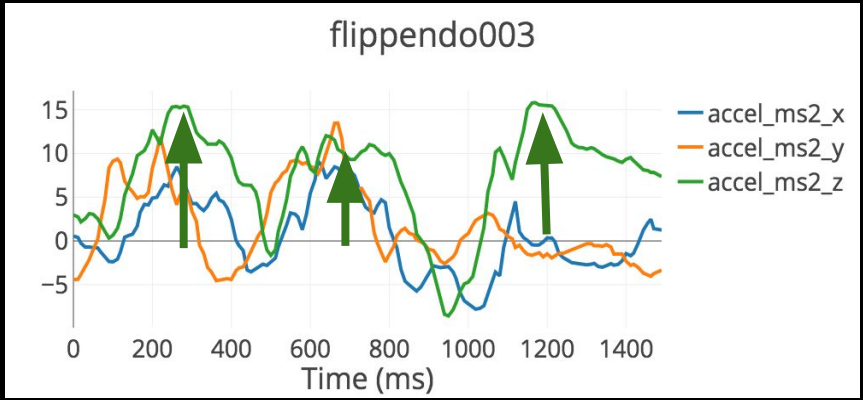




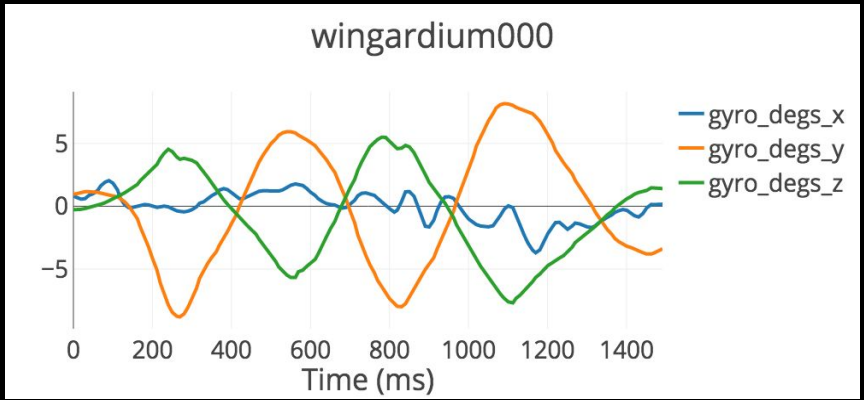
Wingardium has 2 accel z-axis peaks while flippendo has 3!

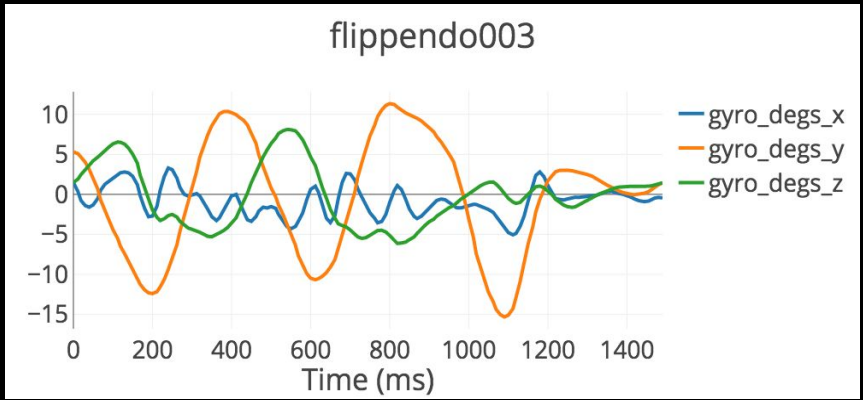
Accel





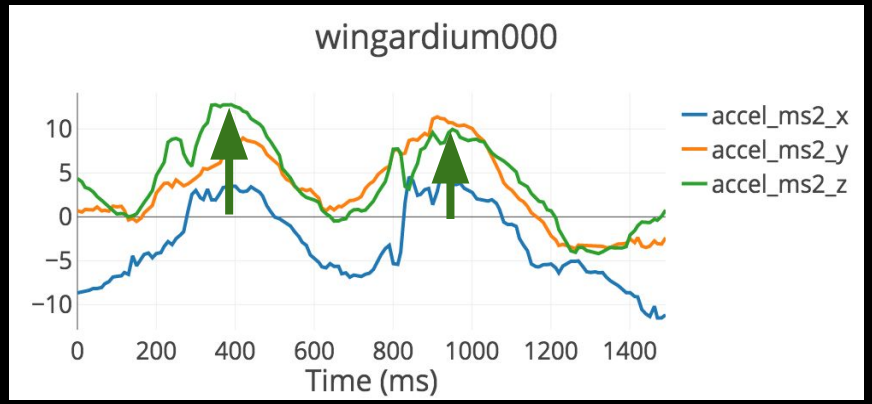
Gyro

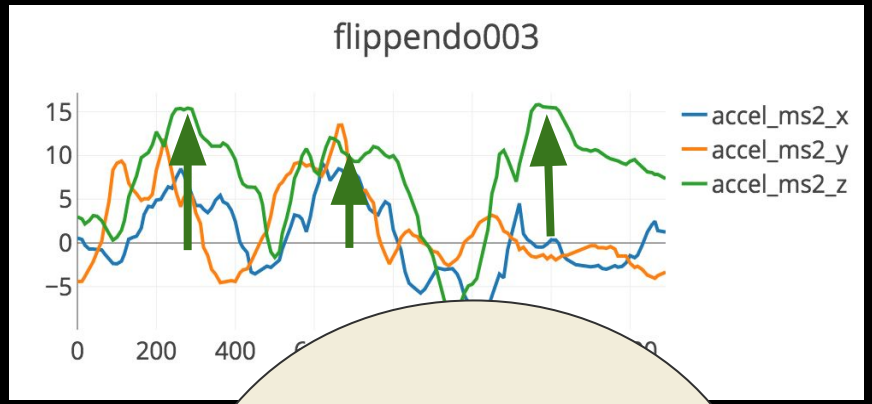




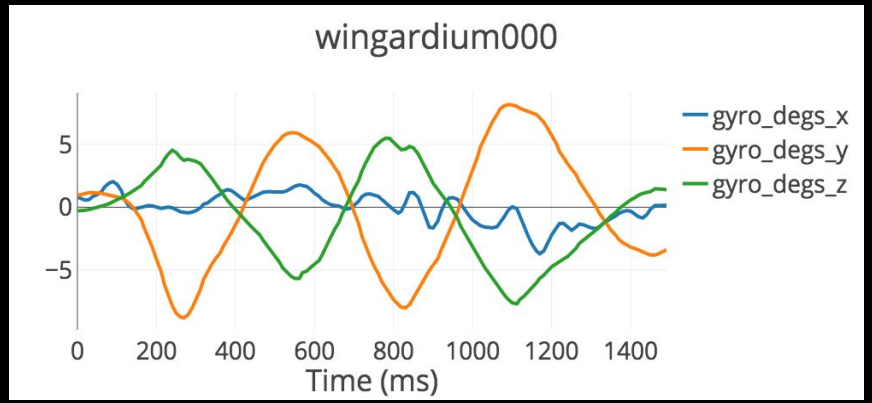
Wingardium has 2 accel z-axis peaks while flippendo has 3!

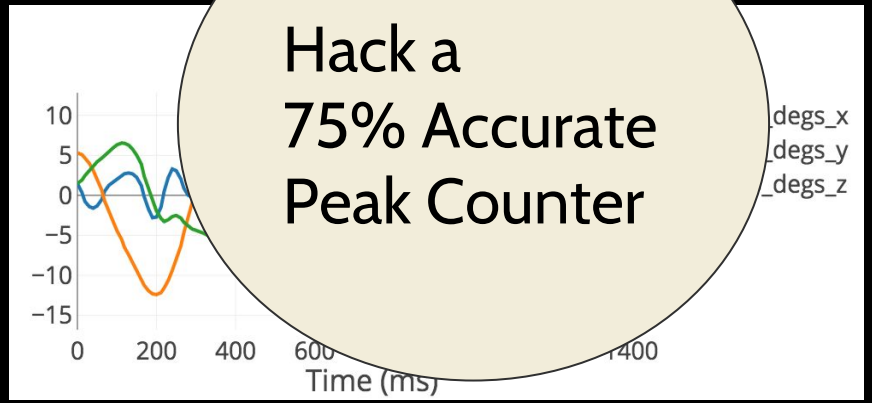
Accel





Gyro





Hack a
75% Accurate
Peak Counter

Features

- max_accel
- min_accel
- Range_accel
- mean_accel
- std_accel

- max_gyro
- min_gyro
- range_gyro
- mean_gyro
- std_gyro

Features

- max_accel
- min_accel
- Range_accel
- mean_accel
- std_accel
- accel_z_peaks
- max_gyro
- min_gyro
- range_gyro
- mean_gyro
- std_gyro

Train model & iterate

- Iteration #1 → Didn't work well. 102 traces.
  - Clean Data! Collect new data!
- Iteration #2 → Didn't work well. 167 traces.
  - Clean Data! Collect new data!
- Iteration #3: → Worked OK, distinguish gestures. 286 traces.
  - Feature Design

Train model & iterate

- Iteration #1 → Didn't work well. 102 traces.
  - Clean Data! Collect new data!
- Iteration #2 → Didn't work well. 167 traces.
  - Clean Data! Collect new data!
- Iteration #3: → Worked OK, distinguish gestures. 286 traces.
  - Feature Design
- Iteration #4 → Good enough. It's Halloween! Ship it!

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

Productionize the model

- gesture_detector.py
- <100 lines of code
- Might need a little / lot of tuning ;)

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

Build more cool stuff!

jewang.net / jen@jewang.net

IMUs + Machine Learning on a large scale...

I <3 sensors

- City walkability
- Better health phenotypes
- Depression treatments
- Census data
- Earthquake detection

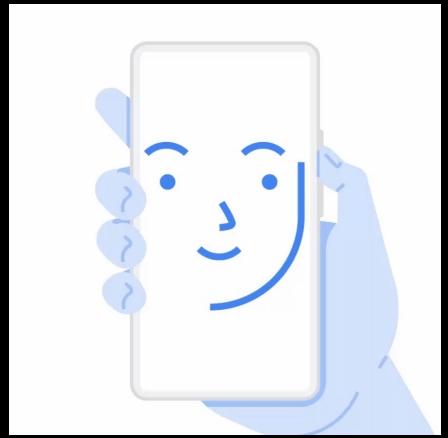
IMUs + Machine Learning = Lots of fun!

Real firebending (Allen Pan)

Smartwatch gestures/input/stealing ATM Pins

Music Control Gloves (Mimu Gloves used by Imogen Heap)

Gait-based auth



Android Camera Control Gestures

Fitness / Sleep Tracking "actigraphy" (FitBit)

How to Gesture Recognition

1. Define & scope the problem
2. Propose machine learning model / algorithm
3. Collect data
4. Train model & iterate
5. Productionize the model

Build more cool stuff!

jewang.net / jen@jewang.net

Appendix

How does a Linear SVC model work?