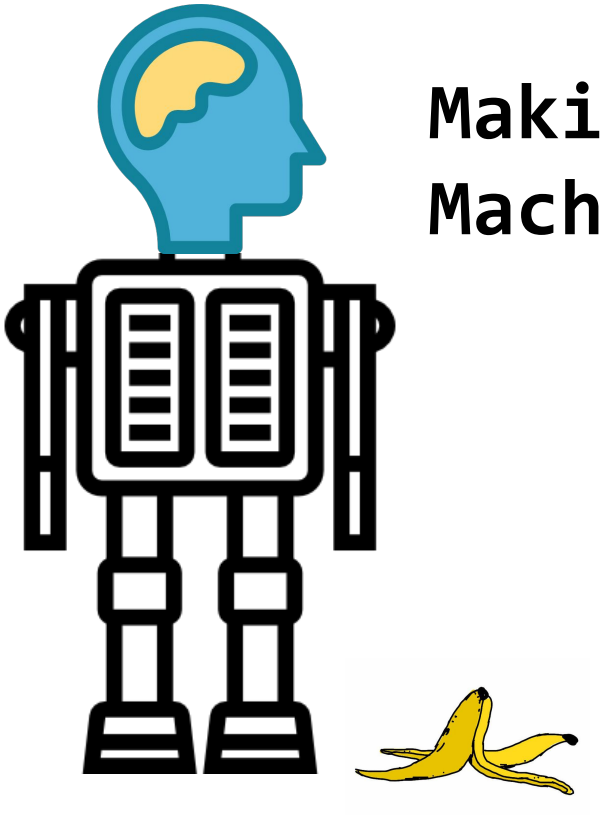


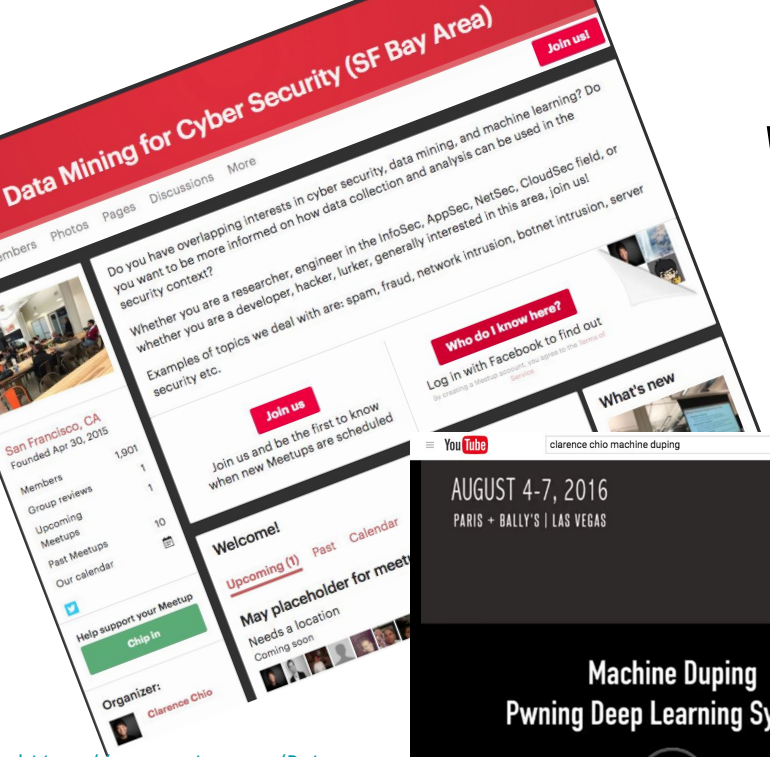
Making & Breaking Machine Learning Systems

(for infosec)

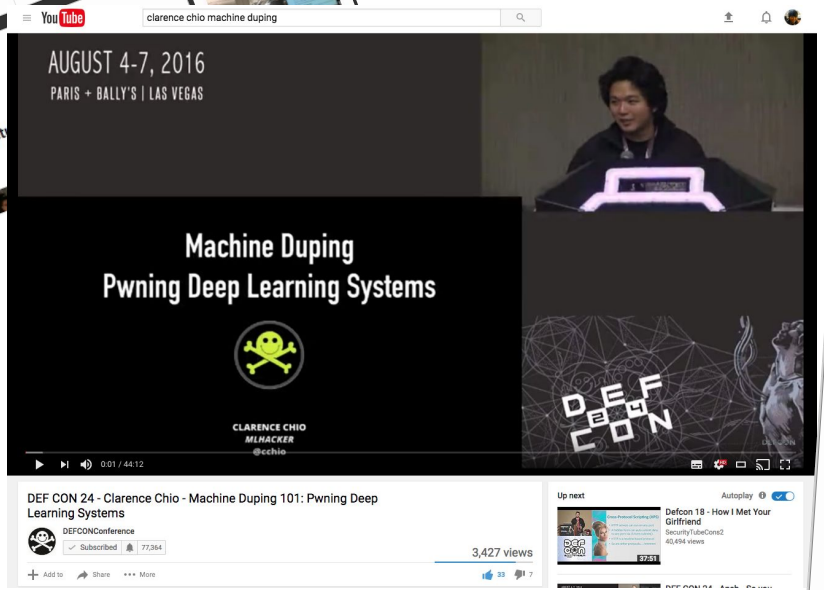
who are you?

who are we?

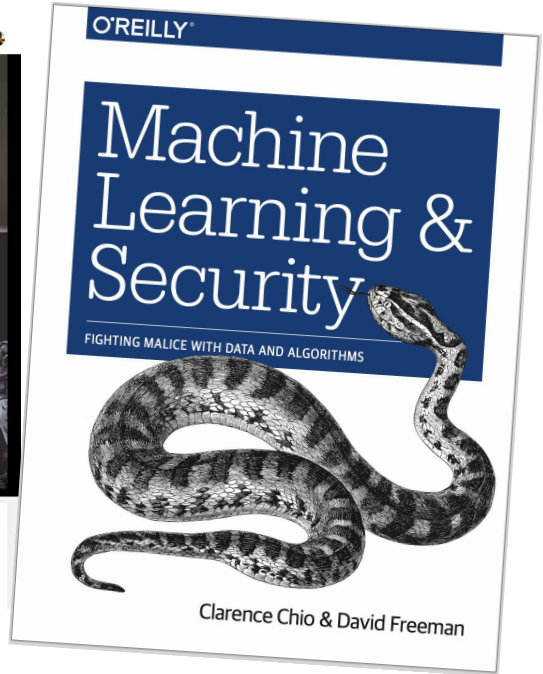
clarence chio (@cchio)



https://www.meetup.com/Data-Mining-for-Cyber-Security/

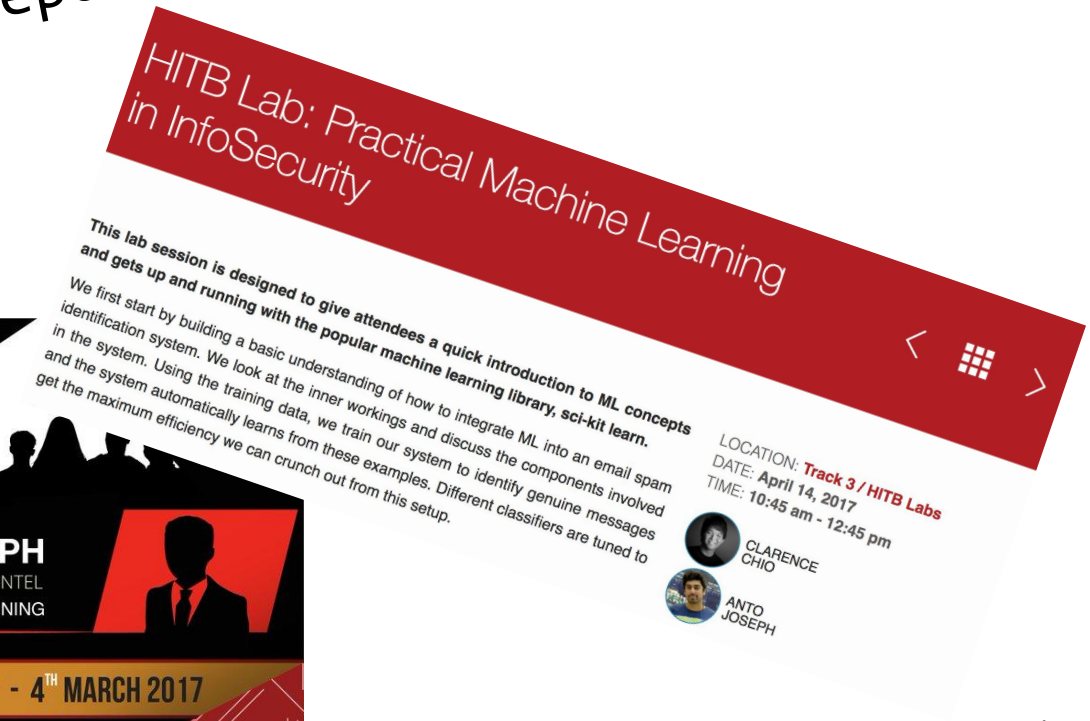


https://www.youtube.com/watch?v=JAGDpJFFM2A



who are we?

anto joseph (@antojosep007)



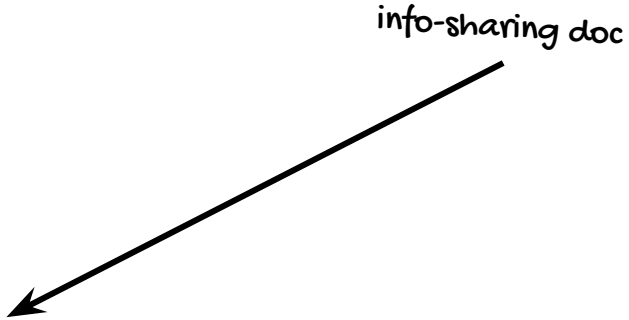
Agenda - Day one

- Setting up a development environment
- Building a spam classifier (HANDS-ON)
- Machine learning crash course (PART ONE)
- Using machine learning in infosec
- Machine learning crash course (PART TWO)
- Classifying malware (HANDS-ON)
- Machine learning WAF implementations (HANDS-ON)

Agenda - Day two

- Deep learning crash course
- TensorFlow for everyone
- Building neural networks (HANDS-ON)
- LSTMs (HANDS-ON)
- Machine learning @ scale & in production (HANDS-ON)
- How to break machine learning
  (this is why we can’t have nice things)
  - Adversarial deep learning (HANDS-ON)
- FINAL CHALLENGE

Setting up your environment:



goo.gl/9Bv5NS

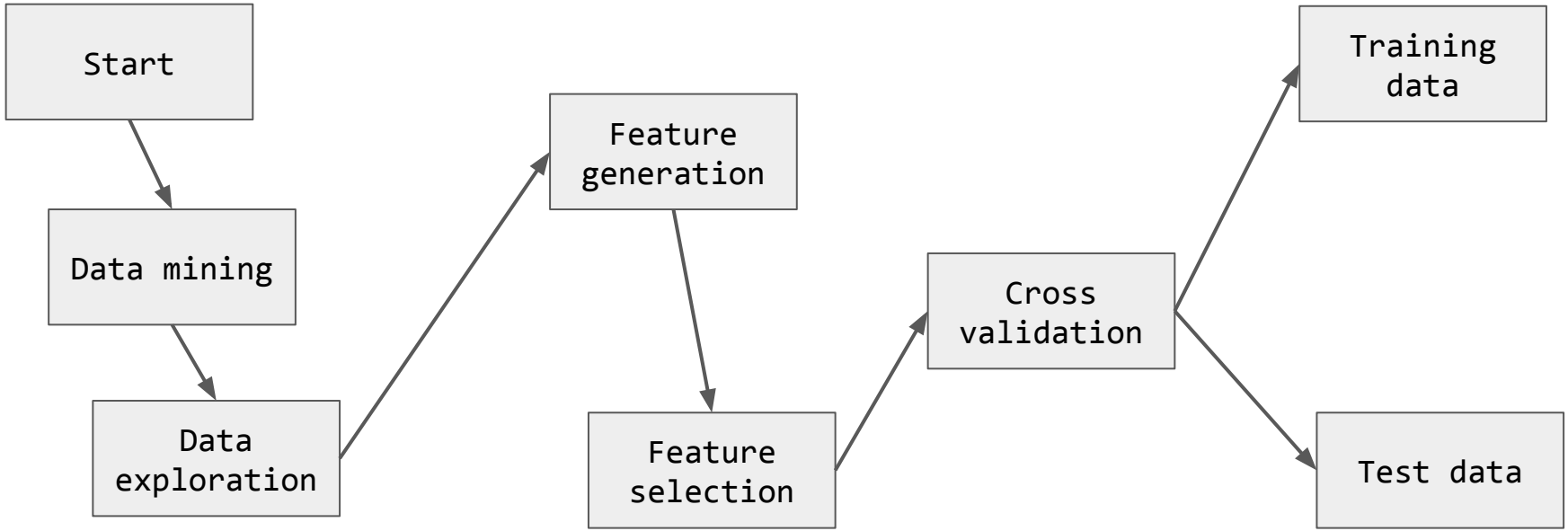
Let f be a function representing this training session. Let x be the amount of math involved.

$$\lim_{x \to 0} f(x)$$

(supervised)

Machine learning from 10,000ft

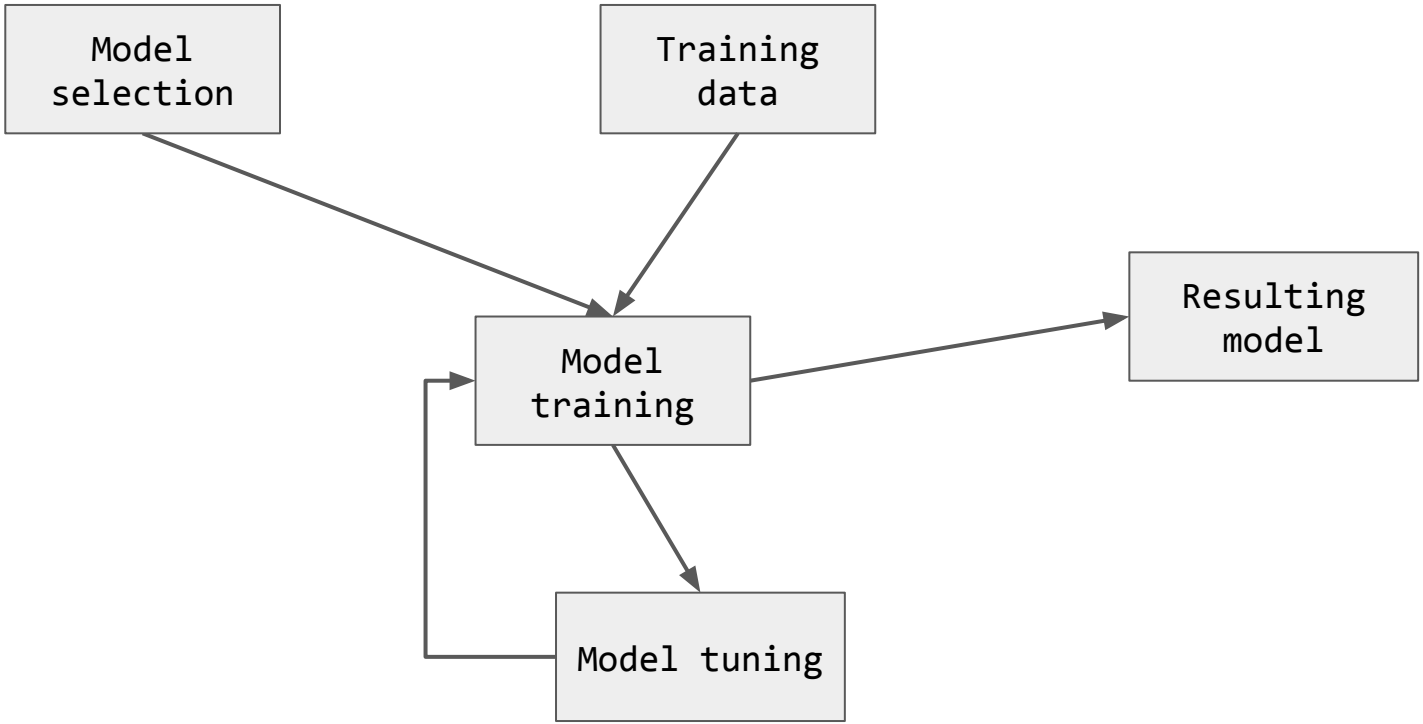
data engineering phase



(supervised)

Machine learning from 10,000ft

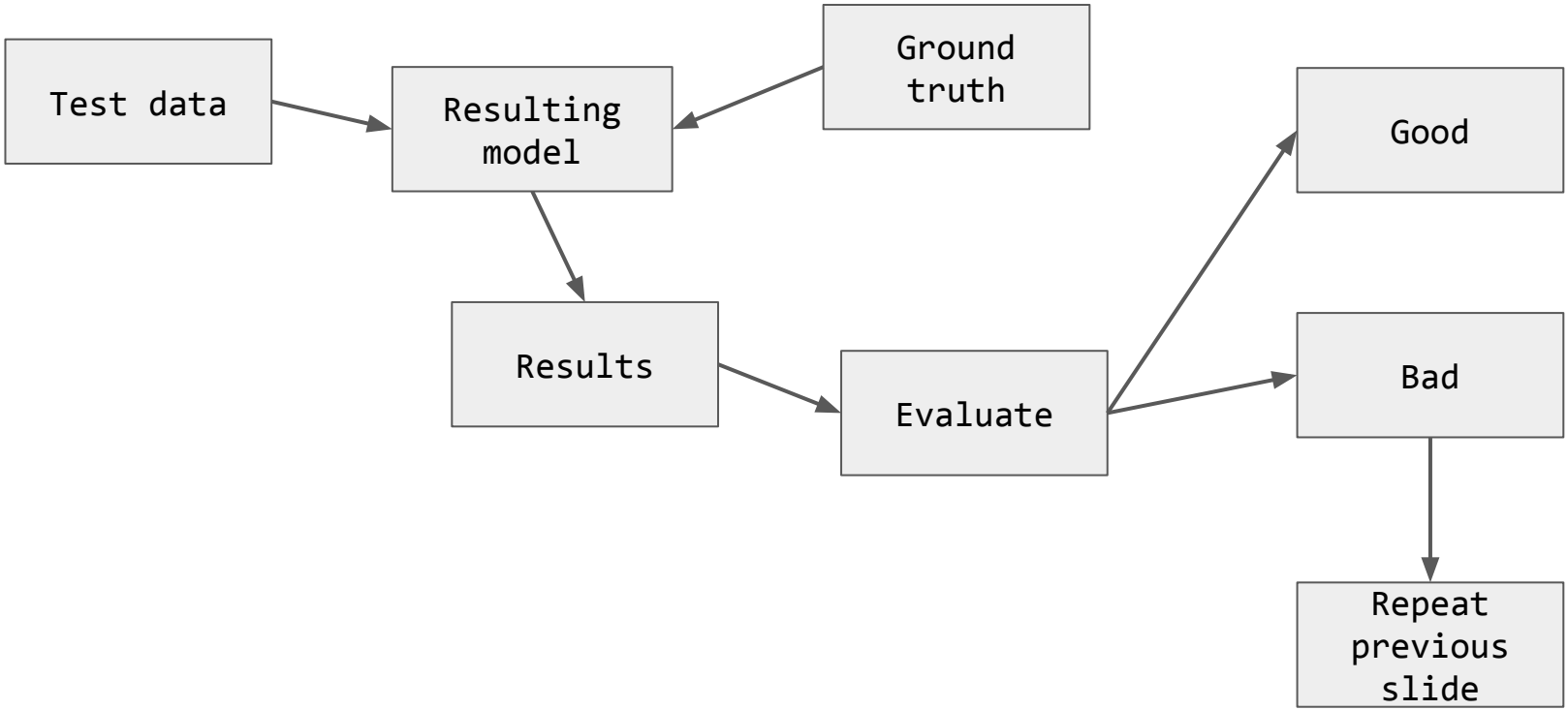
model training phase



(supervised)

Machine learning from 10,000ft

model validation phase



Python toolKits

- Scikit-Learn - Python library that implements a range of machine **learning algos and helpers**

- TensorFlow - library for numerical computation using data flow graphs / deep learning

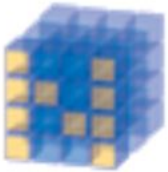
Scikit-Learn

- easy-to-use, general-purpose toolbox for machine learning in Python.
- supervised and unsupervised machine learning techniques.
- Utilities for common tasks such as model selection, feature extraction, and feature selection
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Tensorflow

- Open source
- By Google
- used for both research and production
- Used widely for Deep learning
- Multiple GPU Support

Data science libs



NumPy

Base N-dimensional array package

SciPy library

Fundamental library for scientific computing

Matplotlib

Comprehensive 2D Plotting

IP[y]:
IPython

IPython

Enhanced Interactive Console

Sympy

Symbolic mathematics

pandas

Data structures & analysis

Basic terms

Classifier

"An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm, that maps input data to a category."

Model

Linear regression algorithm is a technique to fit points to a line $y = m x+c$. Now after fitting, you get for example, $y = 10 x + 4$. This a model. A model is something to which when you give an input, gives an output. In ML, any 'object' created after training from an ML algorithm is a model.

Linear Regression

Fitting a linear relationship b/w two quantitative variables

Problems in machine learning

Overfitting

when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data.

Underfitting

not a suitable model and will be obvious as it will have poor performance on the training data

a model that can neither model the training data nor generalize to new data

Confusion matrix

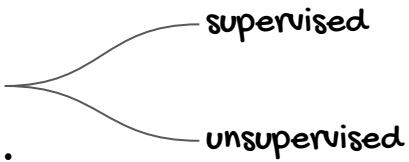
n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

MACHINE LEARNING 101

Types of machine learning use cases:

- Regression
- Classification
- Anomaly detection
- Recommendation

won't cover here, but check out this talk

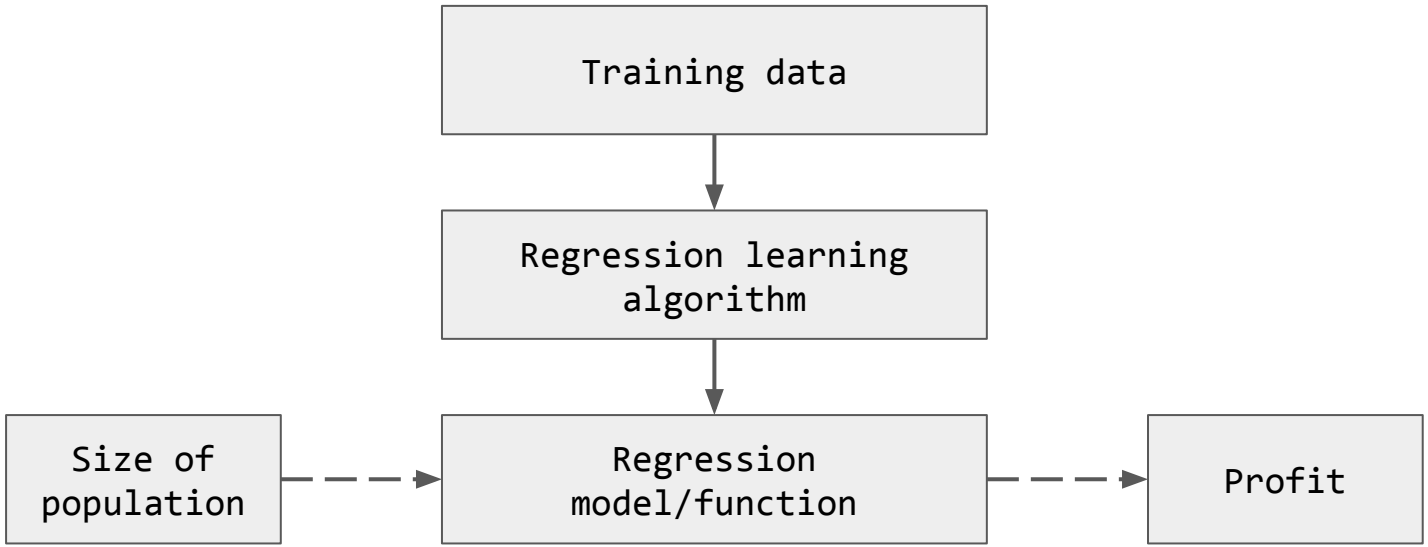


This covers **EVERYTHING**. (almost)

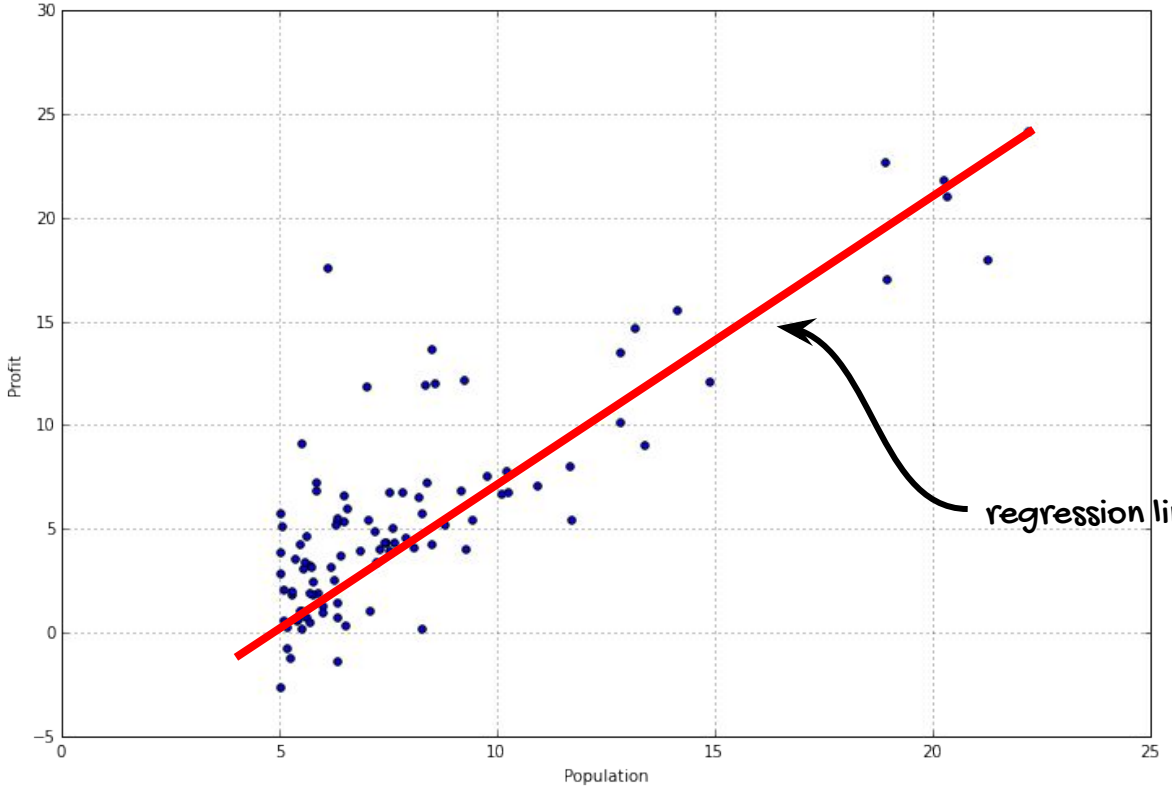
Regression

Regression

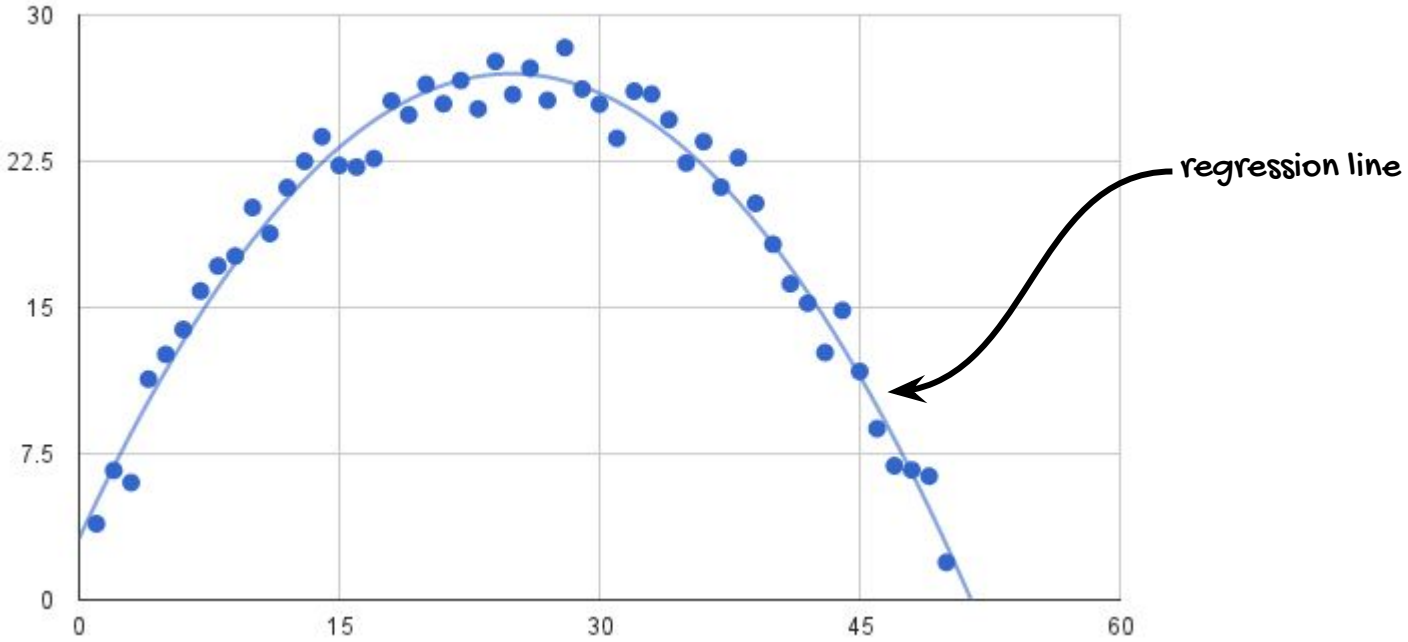
- regression = finding relationships between variables



Linear Regression

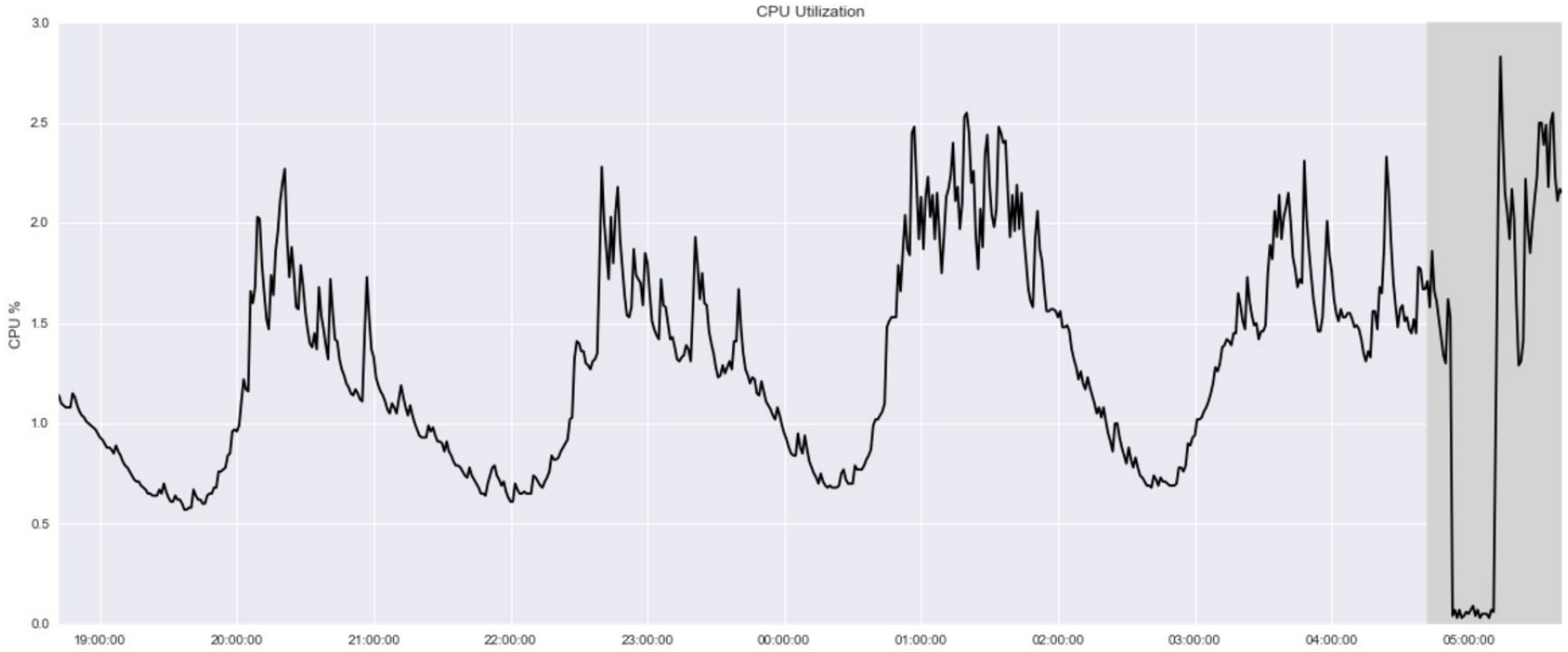


Polynomial Regression



Anomaly Detection

Anomaly detection



Anomaly detection

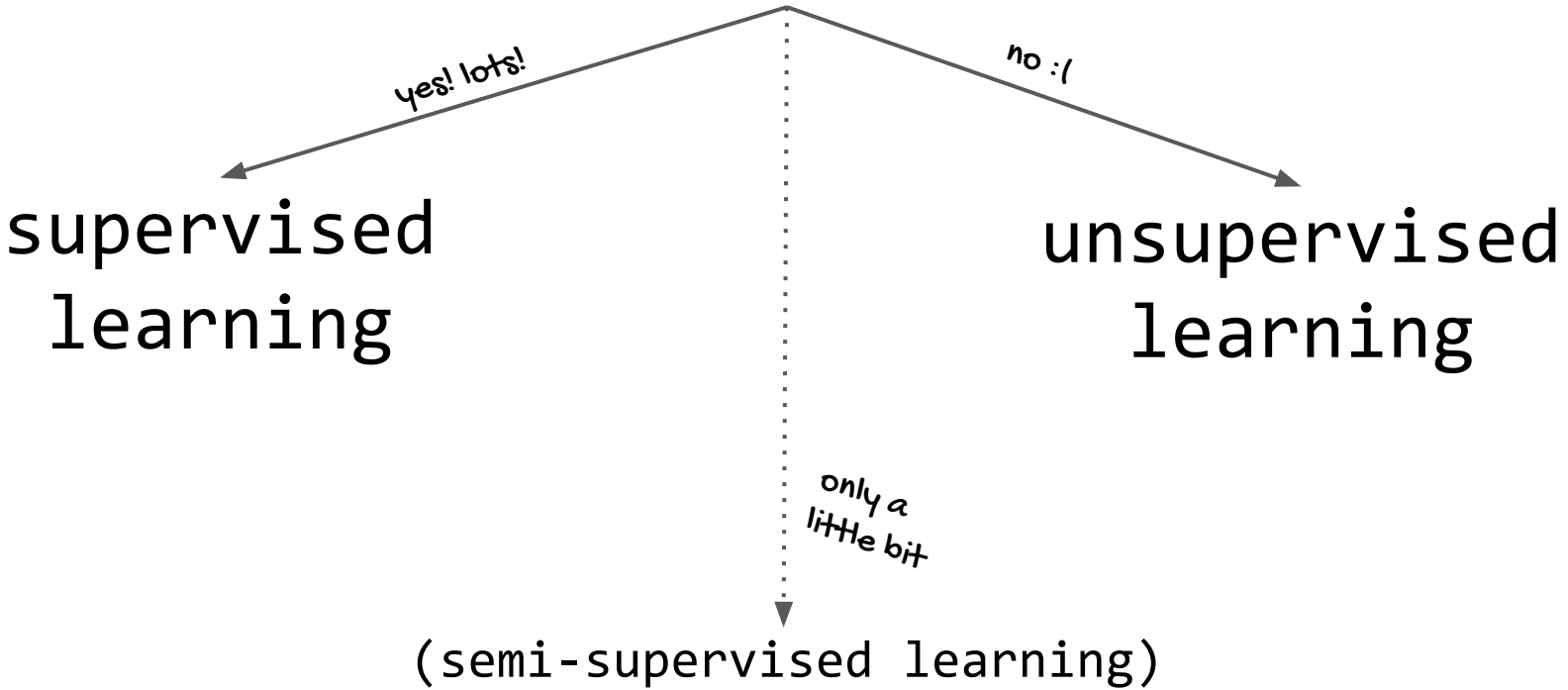
- Outliers vs. novelties
  - *novelties*: unobserved pattern in new observations not included in training data
- Simple statistics/forecasting methods
  - Exponential smoothing, Holt-Winters algorithm
- Machine learning methods
  - Elliptical envelope, density-based, clustering, SVM

Classification

Classification

labeled data - do you have it?

Classification



Supervised classification

- **Many different algorithms!**
- We will go through five:
  - Logistic regression (it's called regression but is *not* regression)
  - Naive Bayes
  - K-nearest neighbors
  - Support Vector Machines
  - Decision Trees

Logistic Regression classifier

- **SUPERVISED LEARNING**
- CONFUSING - *Logistic regression* is a form of CLASSIFICATION, NOT REGRESSION!
- logistic regression vs. linear regression
-

Naive Bayes classifier

- **SUPERVISED LEARNING**

K-Nearest Neighbors classifier (kNN)

- **SUPERVISED LEARNING**

Support Vector Machines (SVM)

- **SUPERVISED LEARNING**

Decision Tree classifier

- **SUPERVISED LEARNING**

Unsupervised classification

- Mainly refers to **clustering**
- Four types:
  - **Centroid**: K-Means
  - **Distribution**: Gaussian mixture models
  - **Density**: DBSCAN
  - **Connectivity**: Hierarchical clustering

K-Means clustering

- **UNSUPERVISED LEARNING**

Gaussian mixture model clustering

- **UNSUPERVISED LEARNING**

DBSCAN clustering

- **UNSUPERVISED LEARNING**

Hierarchical clustering

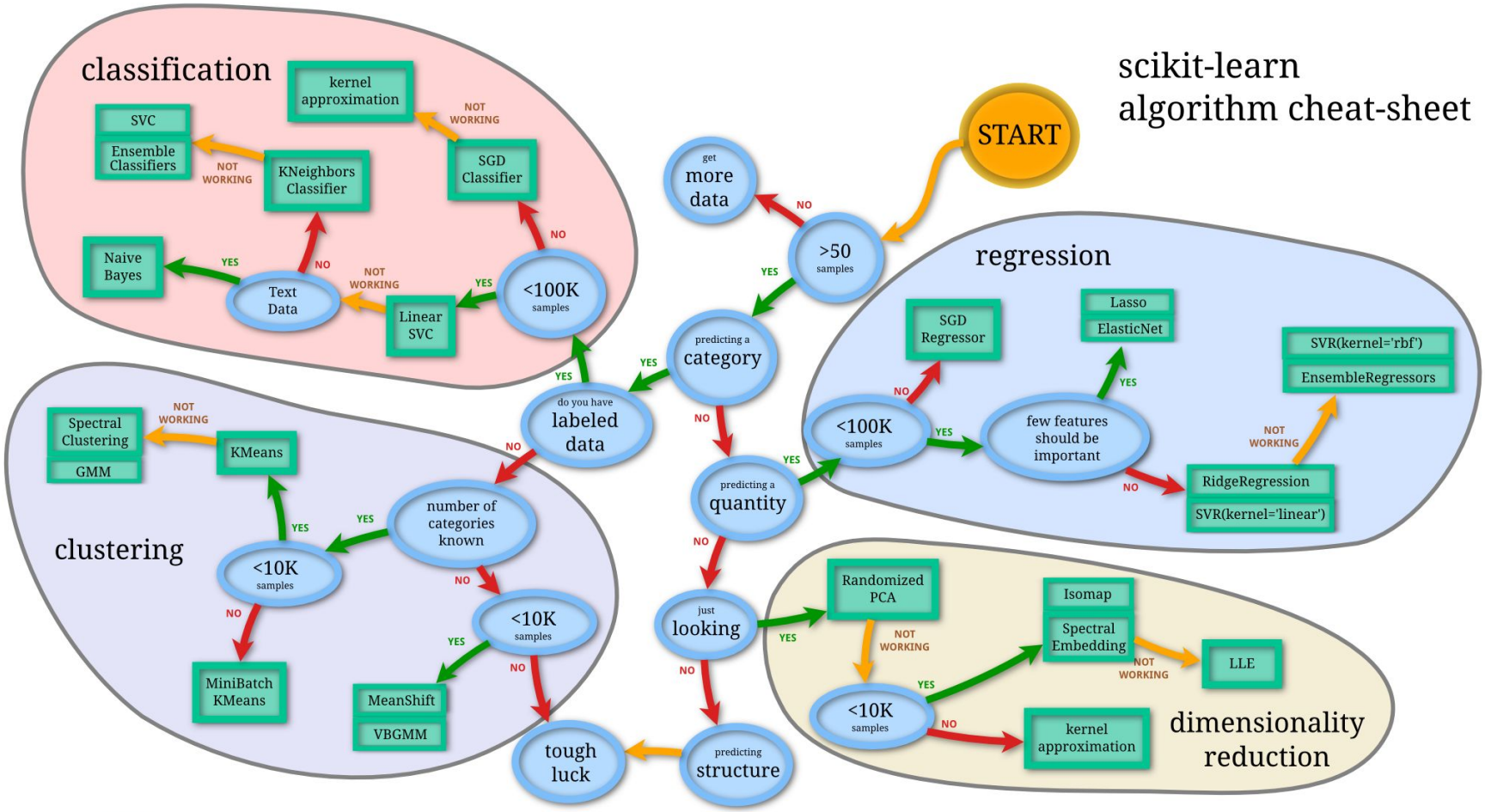
- **UNSUPERVISED LEARNING**

Semi-supervised learning?

- how does this work?

SO MANY ALGORITHMS.

HOW TO PICK.????????


scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
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
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
YES
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')
clustering
Spectral Clustering
NOT WORKING
GMM
KMeans
YES
number of categories known
YES
NO
<10K samples
YES
NO
<10K samples
NO
YES
MiniBatch KMeans
MeanShift
VBGMM
dimensionality reduction
Randomized PCA
YES
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation

Setting up your environment for machine learning

NOTES ABOUT THE ENVIRONMENT

HANDS ON

classifying spam

The dataset: 2007 TREC Public Spam Corpus

http://plg.uwaterloo.ca/~gvcormac/treccorpus07/

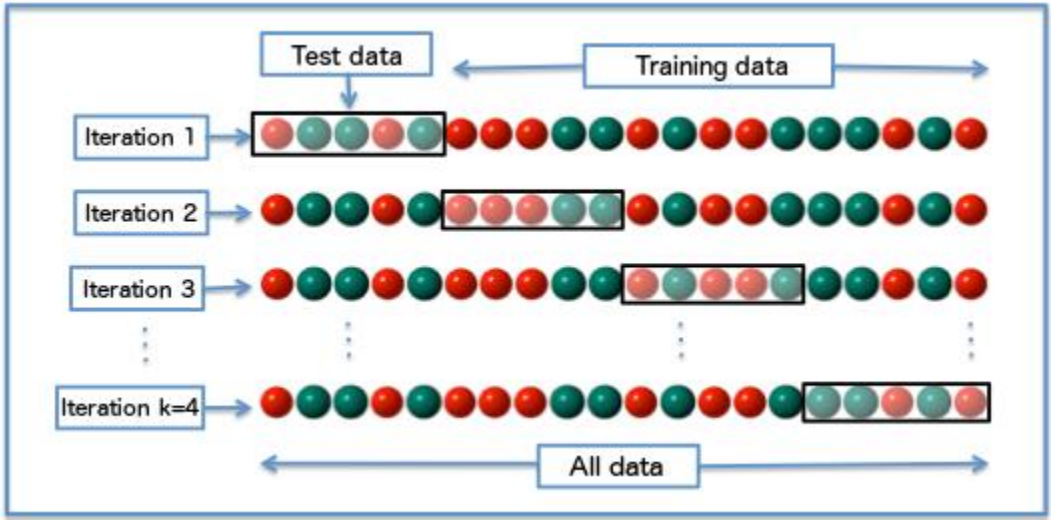
add info about dataset

Important machine learning concepts

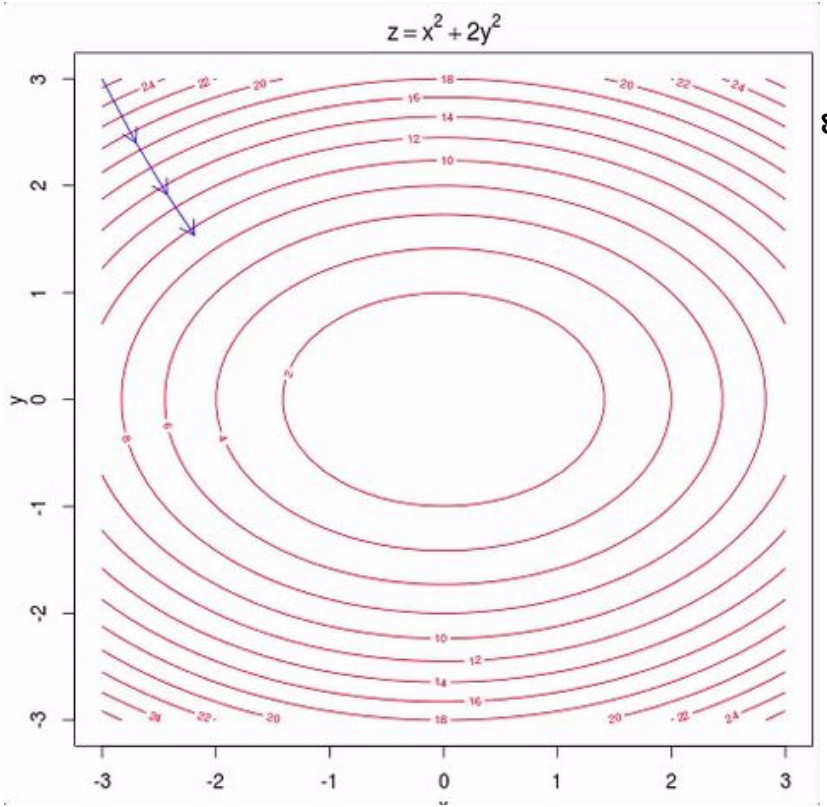
- Cross validation
- Model optimization (gradient descent)
- Ensembles
- Gradient Boosting
- Reinforcement learning
- Deep learning

slides for each of these

Cross validation

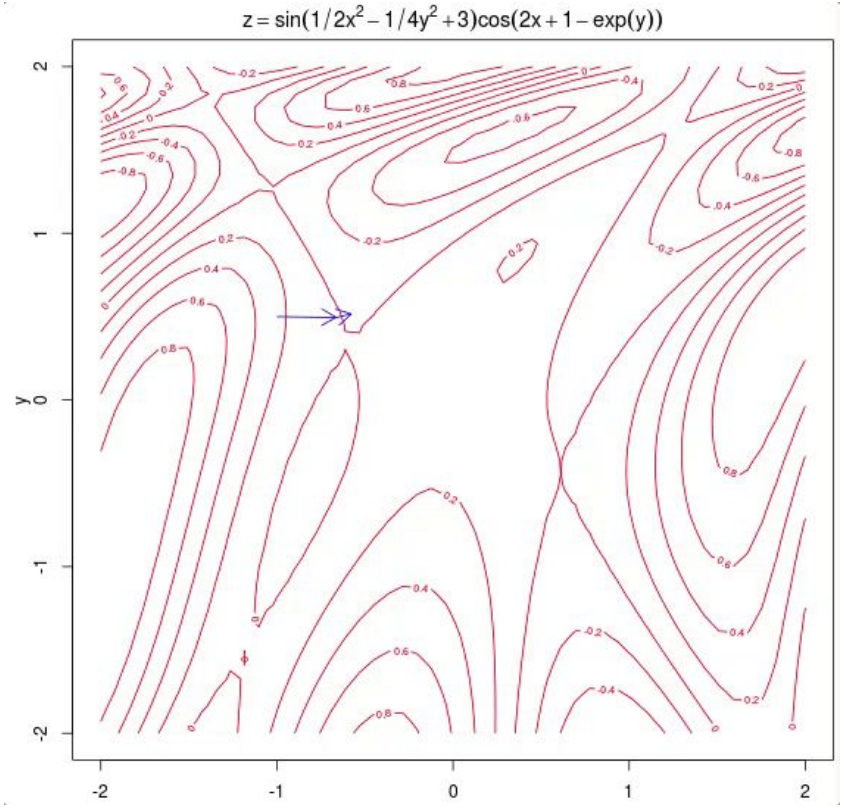


Model optimization - Gradient descent



success

Model optimization - Gradient descent



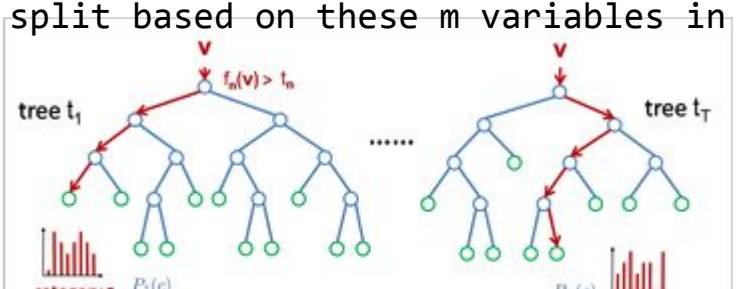
failure

Ensemble classifiers - Bucket of models

```
For each model m in the bucket:
  Do c times: (where 'c' is some constant)
    Randomly divide the training dataset into two datasets: A, and B.
    Train m with A
    Test m with B
Select the model that obtains the highest average score
```

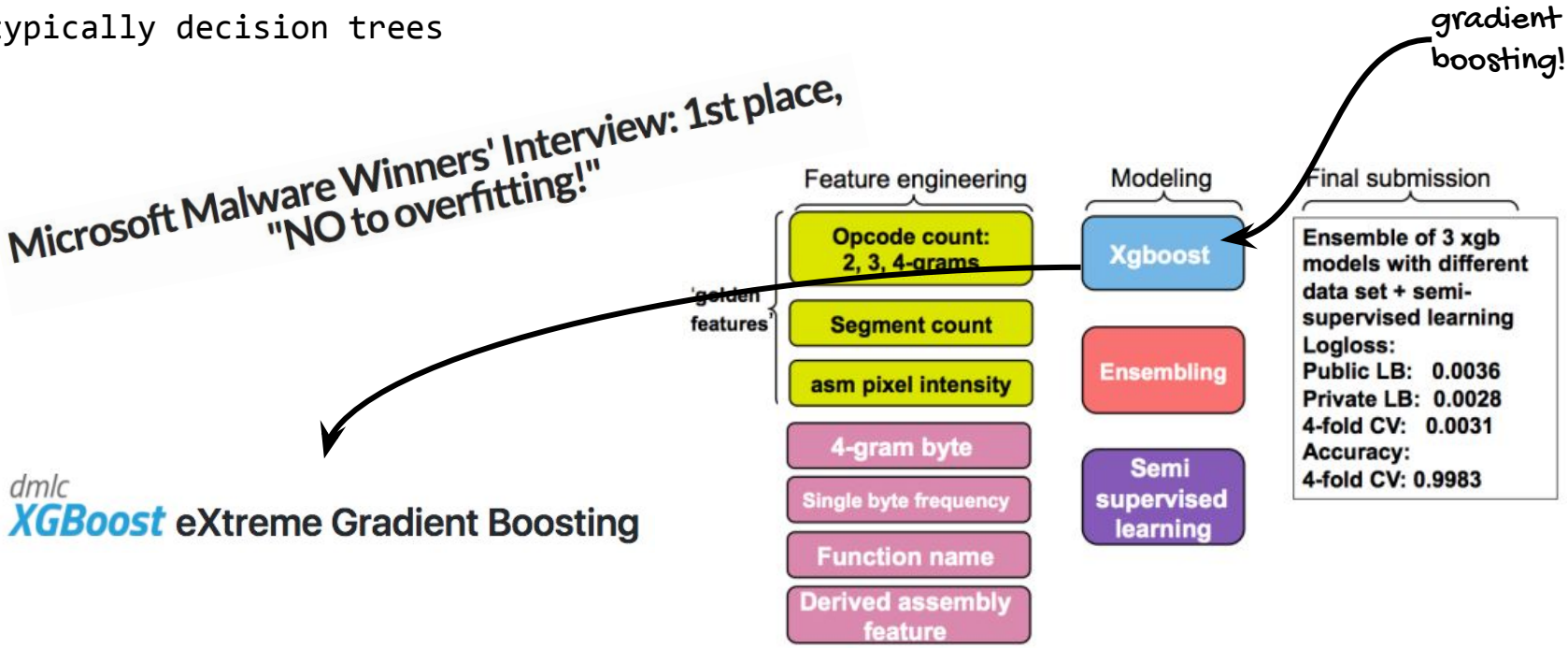
Ensemble classifiers - Random Forest

- Let the number of training cases be N, and the number of variables in the classifier be M.
- The number m of input variables are used to determine the decision at a node of the tree; m should be much less than M.
- Choose a training set for this tree by choosing N times with replacement from all N available training cases. Use the rest of the cases to estimate the error of the tree, by predicting their classes.
- For each node of the tree, randomly choose m variables on which to base the decision at that node. Calculate the best split based on these m variables in the training set.
- Each tree is fully grown and not pruned.



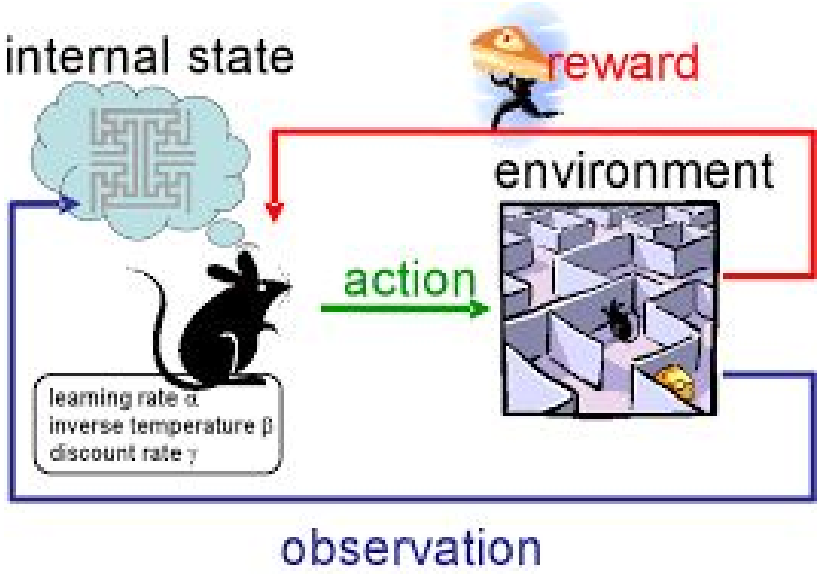
Gradient Boosting (strictly also an ensemble technique)

- A prediction model in the form of an ensemble of weak prediction models, typically decision trees



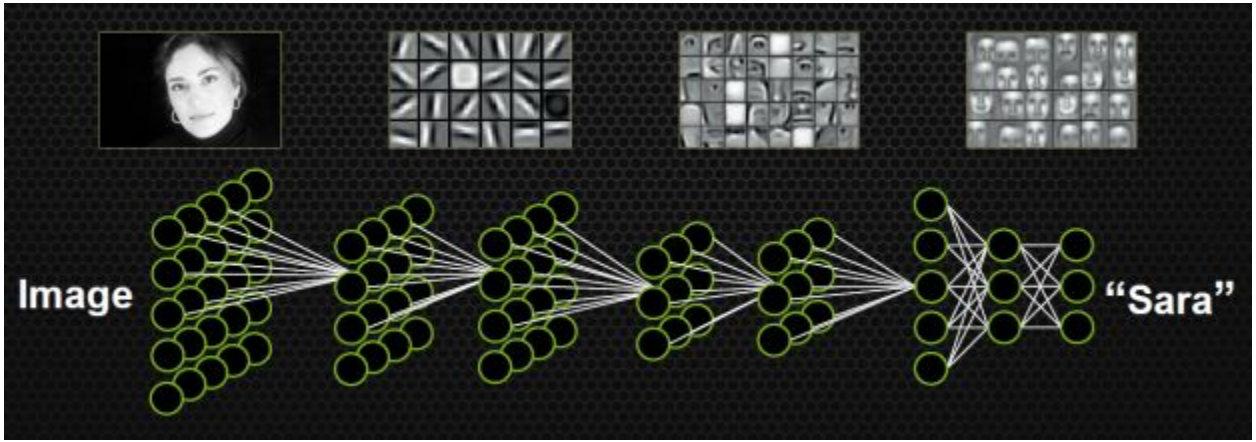
Reinforcement learning

- Inspired by behavioral psychology
- Take actions in an environment so as to maximize some notion of cumulative reward



Deep learning

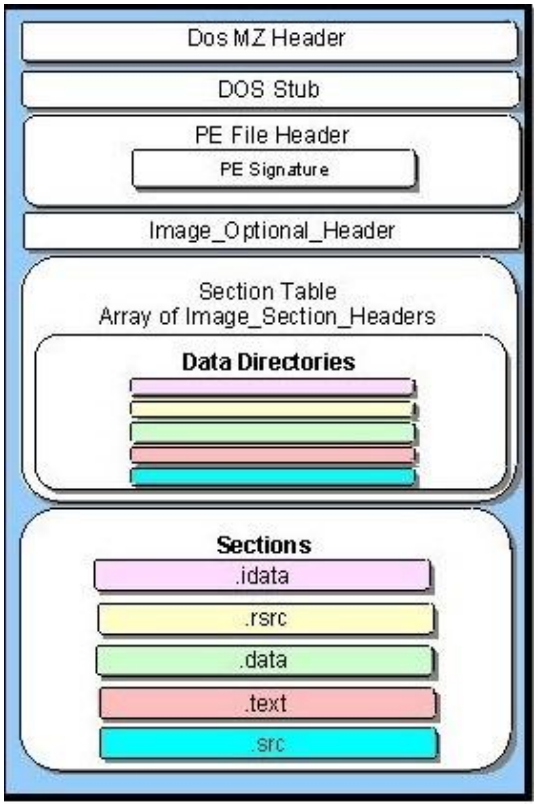
- More tomorrow!



HANDS ON

classifying malware

Portable executable (PE)



pefile dump

----------Parsing Warnings----------

Suspicious NumberOfRvaAndSizes in the Optional Header. Normal values are never larger than 0x10, the value is: 0xdfffddde

Error parsing section 2. SizeOfRawData is larger than file.

----------DOS_HEADER----------

[IMAGE_DOS_HEADER]
e_magic: 0x5A4D
e_cblp: 0x50
e_cp: 0x2

----------NT_HEADERS----------

[IMAGE_NT_HEADERS]
Signature: 0x4550

----------FILE_HEADER----------

[IMAGE_FILE_HEADER]
Machine: 0x14C
NumberOfSections: 0x4
TimeDateStamp: 0x851C3163 [INVALID TIME]
PointerToSymbolTable: 0x74726144
NumberOfSymbols: 0x455068
SizeOfOptionalHeader: 0xE0
Characteristics: 0x818F

----------OPTIONAL_HEADER----------

[IMAGE_OPTIONAL_HEADER]
Magic: 0x10B
MajorLinkerVersion: 0x2
MinorLinkerVersion: 0x19
SizeOfCode: 0x200
SizeOfInitializedData: 0x45400
SizeOfUninitializedData: 0x0
AddressOfEntryPoint: 0x2000
BaseOfCode: 0x1000
BaseOfData: 0x2000
ImageBase: 0xDE0000
SectionAlignment: 0x1000
FileAlignment: 0x1000
MajorOperatingSystemVersion: 0x1
MinorOperatingSystemVersion: 0x0

----------PE Sections----------

[IMAGE_SECTION_HEADER]
Name: CODE
Misc: 0x1000
Misc_PhysicalAddress: 0x1000
Misc_VirtualSize: 0x1000
VirtualAddress: 0x1000
SizeOfRawData: 0x1000
PointerToRawData: 0x1000
PointerToRelocations: 0x0
PointerToLinenumbers: 0x0
NumberOfRelocations: 0x0
NumberOfLinenumbers: 0x0
Characteristics: 0xE0000020
Flags: MEM_WRITE, CNT_CODE, MEM_EXECUTE, MEM_READ
Entropy: 0.061089 (Min=0.0, Max=8.0)

[IMAGE_SECTION_HEADER]
Name: DATA
Misc: 0x45000
Misc_PhysicalAddress: 0x45000
Misc_VirtualSize: 0x45000
VirtualAddress: 0x2000
SizeOfRawData: 0x45000
PointerToRawData: 0x2000
PointerToRelocations: 0x0
PointerToLinenumbers: 0x0
NumberOfRelocations: 0x0
NumberOfLinenumbers: 0x0
Characteristics: 0xC0000040
Flags: MEM_WRITE, CNT_INITIALIZED_DATA, MEM_READ
Entropy: 7.980693 (Min=0.0, Max=8.0)

[IMAGE_SECTION_HEADER]
Name: NicolasB
Misc: 0x1000
Misc_PhysicalAddress: 0x1000
Misc_VirtualSize: 0x1000
VirtualAddress: 0x47000
SizeOfRawData: 0xEFEFADFF
PointerToRawData: 0x47000
PointerToRelocations: 0x0
PointerToLinenumbers: 0x0
...

PE feature vector

Name|md5|Machine|SizeOfOptionalHeader|Characteristics|MajorLinkerVersion|MinorLinkerVersion|SizeOfCode|SizeOfInitializedData|SizeOfUninitializedData|AddressOfEntryPoint|BaseOfCode|BaseOfData|ImageBase|SectionAlignment|FileAlignment|MajorOperatingSystemVersion|MinorOperatingSystemVersion|MajorImageVersion|MinorImageVersion|MajorSubsystemVersion|MinorSubsystemVersion|SizeOfImage|SizeOfHeaders|CheckSum|Subsystem|DllCharacteristics|SizeOfStackReserve|SizeOfStackCommit|SizeOfHeapReserve|SizeOfHeapCommit|LoaderFlags|NumberOfRvaAndSizes|SectionsNb|SectionsMeanEntropy|SectionsMinEntropy|SectionsMaxEntropy|SectionsMeanRawsize|SectionsMinRawsize|SectionMaxRawsize|SectionsMeanVirtualsize|SectionsMinVirtualsize|SectionMaxVirtualsize|ImportsNbDLL|ImportsNb|ImportsNbOrdinal|ExportNb|ResourcesNb|ResourcesMeanEntropy|ResourcesMinEntropy|ResourcesMaxEntropy|ResourcesMeanSize|ResourcesMinSize|ResourcesMaxSize|LoadConfigurationSize|VersionInformationSize|**legitimate**

legitimate:

memtest.exe|631ea355665f28d4707448e442fbf5b8|332|224|258|9|0|361984|115712|0|6135|4096|372736|4194304|4096|512|0|0|0|0|1|0|1036288|1024|485887|16|1024|1048576|4096|1048576|4096|0|16|8|5.7668065537|3.60742957555|7.22105072892|59712.0|1024|325120|126875.875|896|551848|0|0|0|0|4|3.26282271103|2.56884382364|3.53793936419|8797.0|216|18032|0|16|1

malware:

VirusShare_76c2574c22b44f69e3ed519d36bd8dff|76c2574c22b44f69e3ed519d36bd8dff|332|224|258|10|0|28672|445952|16896|14819|4096|32768|4194304|4096|512|5|0|6|0|5|0|3977216|1024|680384|2|34112|1048576|4096|1048576|4096|0|16|6|2.6506418400 9|0.0|6.49788465186|30634.6666667|0|139264|661773.333333|3978|3362816|8|172|1|0|21|3.42072662405|1.86523352037|7.9688495098|6558.42857143|180|67624|0|0|0