Data Science

Yogesh Kulkarni

About Me

Yogesh Haribhau Kulkarni

Bio:

- 20+ years in CAD/Engineering software development, in various capacities, including R & D group/site manager.
- Got Bachelors, Masters and Doctoral degrees in Mechanical Engineering (specialization: Geometric Modeling Algorithms).
- Using Python-Machine/Deep Learning for Natural Language Processing.

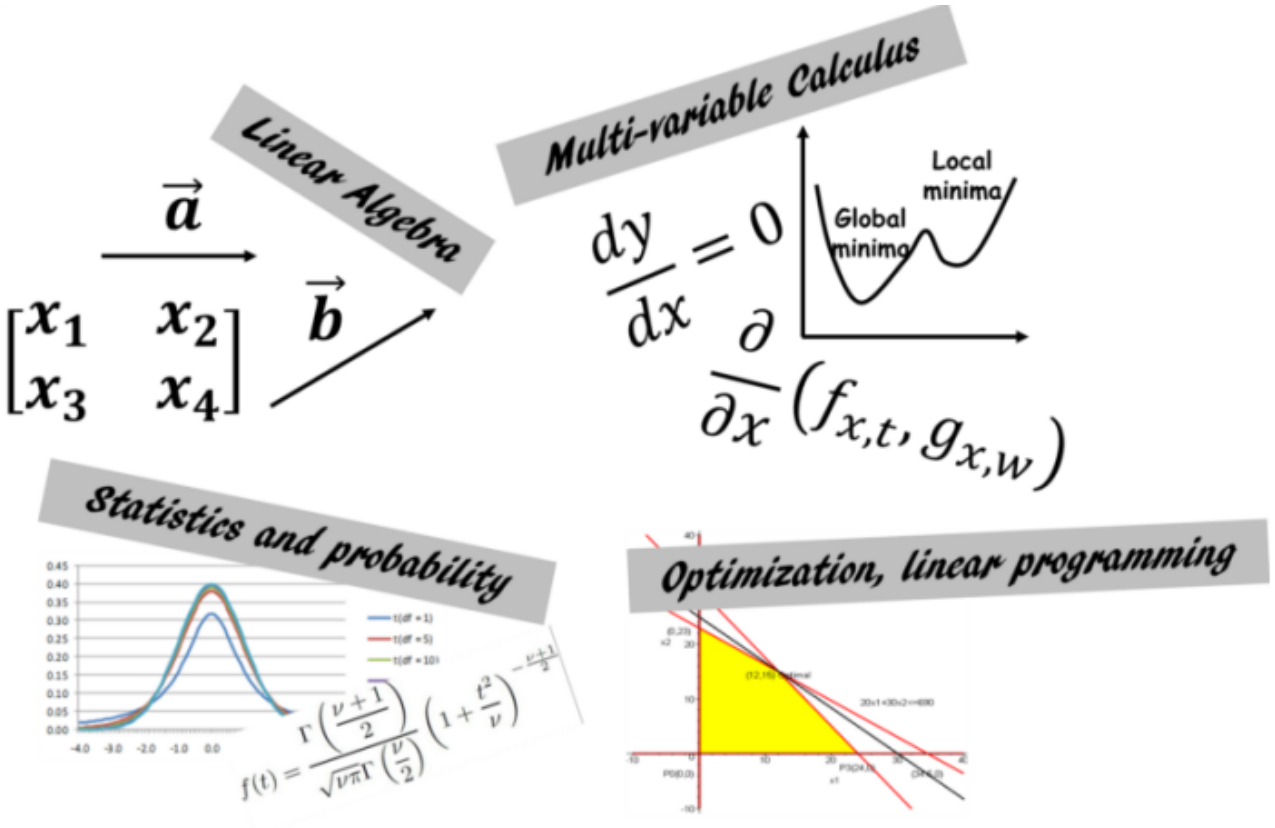
Contact:
https://www.linkedin.com/in/yogeshkulkarni/
https://github.com/yogeshhk
Office Hours: Saturdays, 2 to 5pm (IST); Free-Open to all; email for appointment.

Course logistics

- TextBook: None
- Per week course-material on github: yogeshhk/TeachingDataScience.
- Have personal python environment.
- Backup code: Google drive or somewhere.
- Communication: Google Spaces (send your gmail to add)
- Attendance: Strongly advised.

Pre-requisites

- Some Programming
- College level Mathematics



(Ref: "How Much Mathematics Does an IT Engineer Need to Learn to Get Into Data Science?" -KDnuggets)

Evaluation (subject to change)

- Assignments: 4-5 coding assignments.
- T1, T2 coding projects
- Mid-sem and final exam: Written + code questions

Warm-up

Give out papers . . .

- Write compilable code, in any language, for Fibonacci Series (10 lines)
- What is Machine Learning? Your thoughts (5 lines)

Introduction to Artificial Intelligence

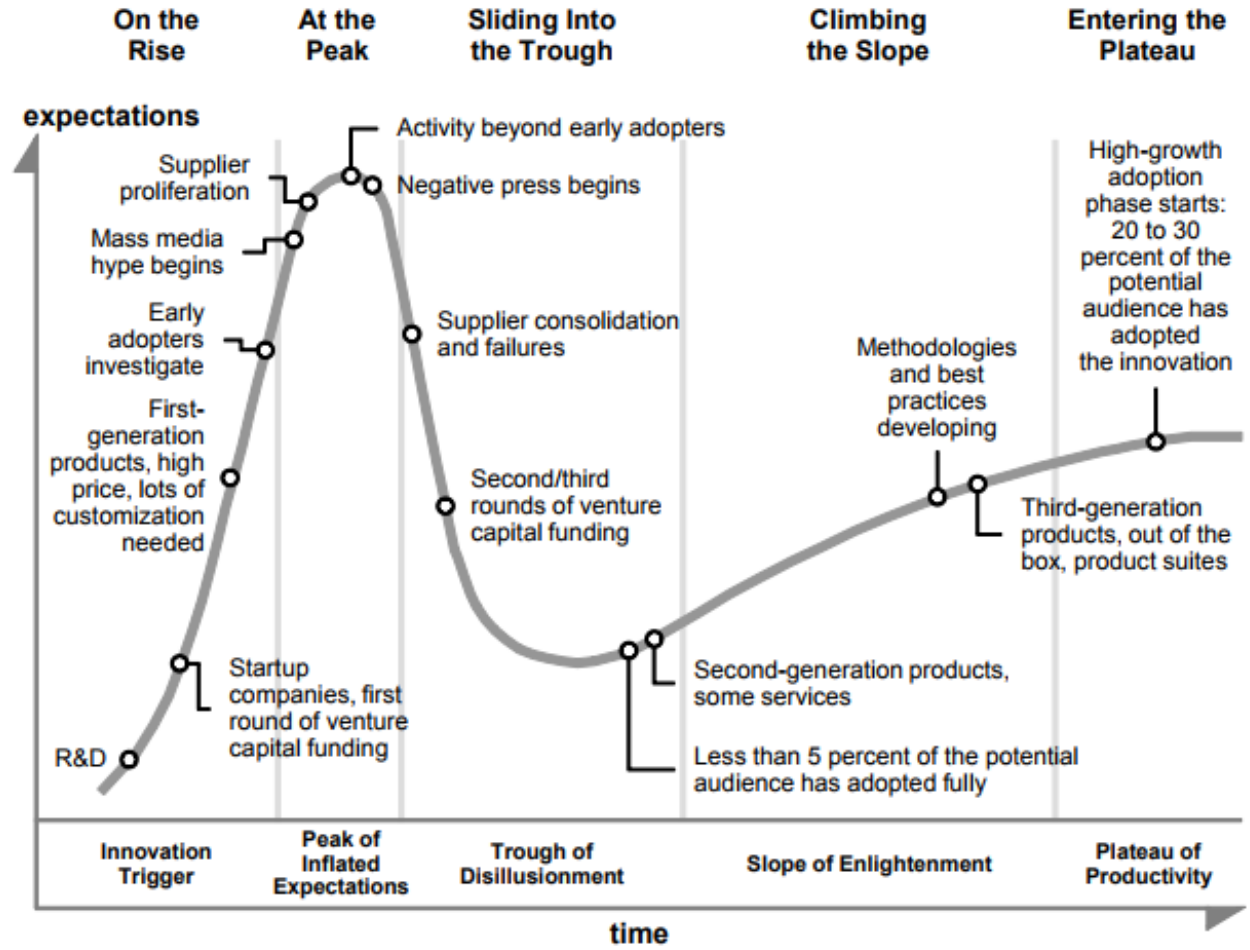
The Problem

Every company is claiming to be working in AI-ML

- Is it really so?
- What exactly is AI (ML)?
- What is not AI?

Or is it just a plain BIG hype?

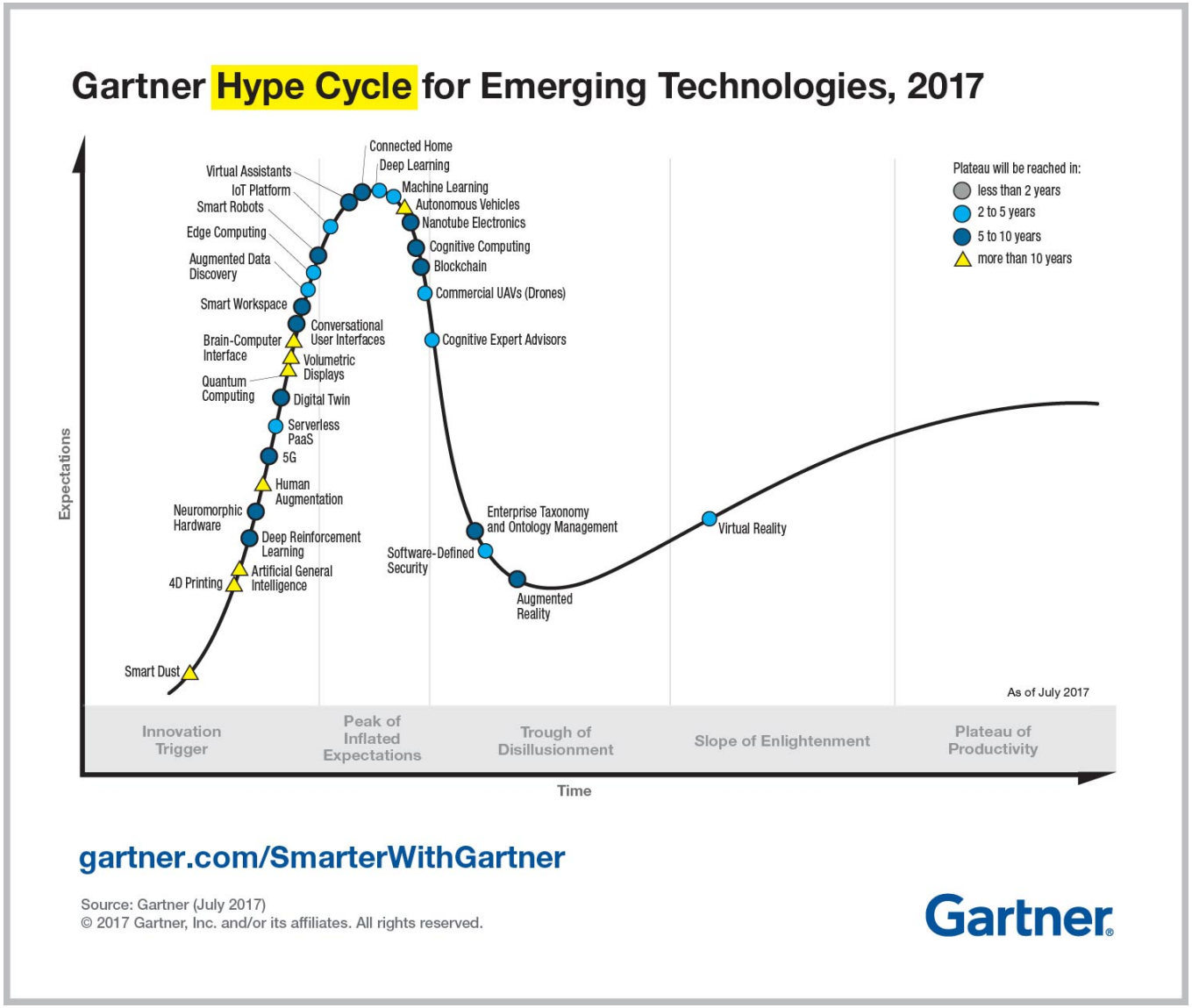
Technology Phases



Source: Gartner

(Ref: Understanding Gartner's Hype Cycles - Jackie Fenn, Mark Raskino, Betsy Burton)

2017 Hype Cycle



The Peak

- "Machine Learning", "Deep Learning" at the Peak
- May take 2 to 5 years to mature well
- If they survive disillusionment, then can be long term players

What is the Core Idea?

What's the core idea?'

- behind problem solving?
- behind writing software algorithms?
- solving research problems?

Desire

- To find a "function"
- To find a relation
- To find a transformation
- To build a model
- From given inputs to desired outputs.

That's it.

Functions

- Some functions are straight forward
- *"In summer, ice-cream sale goes up"*
- Cause and effect
- Relation (function, Mathematical model) is found out
- Here, simple rule based programming suffices

Functions

- But some functions are complex
- *"More you put efforts, your business flourishes."*
- Cause and effect again, but the relation is far to complex
- Too many variables
- Here, simple rule based programming not humanly possible.
- Lots of research needed to come up with equations.

Functions

- $E = mc^2$
- What's this? a function?
- Input variable(s)?
- Output variable(s)?
- Parameters?
- How's the relation? linear?

Controversial Example

- Even astrology is a model, based on the past cases.
- Could claim imperical evidence.
- Given this planetary position, it predicts.
- Represented by “Horoscope”
- Got weights for each planets (real or ficticous)
- Reliable??

Functions

- But most real-life functions are not deterministic
- Some are probabilistic, some non-linear.
- *"Detecting if the tumor is benign or malignant"*
- *"At any state in the game of chess, whats the next move?"*

Chess: next move?

- Needs extreme expertise
- Needs "intelligence"
- How do you get that?
  - Built by lots of training.
  - By studying lots of past games.
- This is how Humans build intelligence

Intelligence

- Can machine (software/program) also do the same?
- Can it play chess?
- Can it build intelligence?
- By looking at past experiences (data),
- Training Data: games played, moves used, etc.

Yes, it can!! Thats Artificial Intelligence.

What is AI?

What is Artificial Intelligence (AI)?

My definition:
"If machines (or computer programs) start doing some/all of these "intelligent" tasks, then that's Artificial Intelligence"

Intelligence: the differentiation

- Ability to think various domains
- Ability produce something new
- Ability to detect the unseen
- Ability to enhance knowledge (rules, patterns)

All these, AI has started doing. The AI era has arrived!!

What is Artificial Intelligence (AI)?

As Bernard Marr comments in Forbes, there is a need to distinguish between "the ability to replicate or imitate human thought" that has driven much AI to more recent models which "use human reasoning as a model but not an end goal".

AI era

- Coming of the fourth industrial revolution
- More important than Electricity - Google

"AI happening ten times faster and at 300 times the scale or at roughly 3,000 times the impact of the Industrial Revolution" - McKinsey

(Ref: https://www.mckinsey.com/ /media/McKinsey/Business Functions/Strategy and Corporate Finance/Our Insights/Strategy and corporate finance special collection/Final PDFs/McKinsey-Special-Collections_Trends-and-global-forces.ashx)

Everyday usage

Artificial intelligence seems to have become ubiquitous.

- Replying to our emails on Gmail
- Learning how to drive our cars,
- Sorting our holiday photos.
- etc.

Too good to be true, isn't it, sort of Magical !!

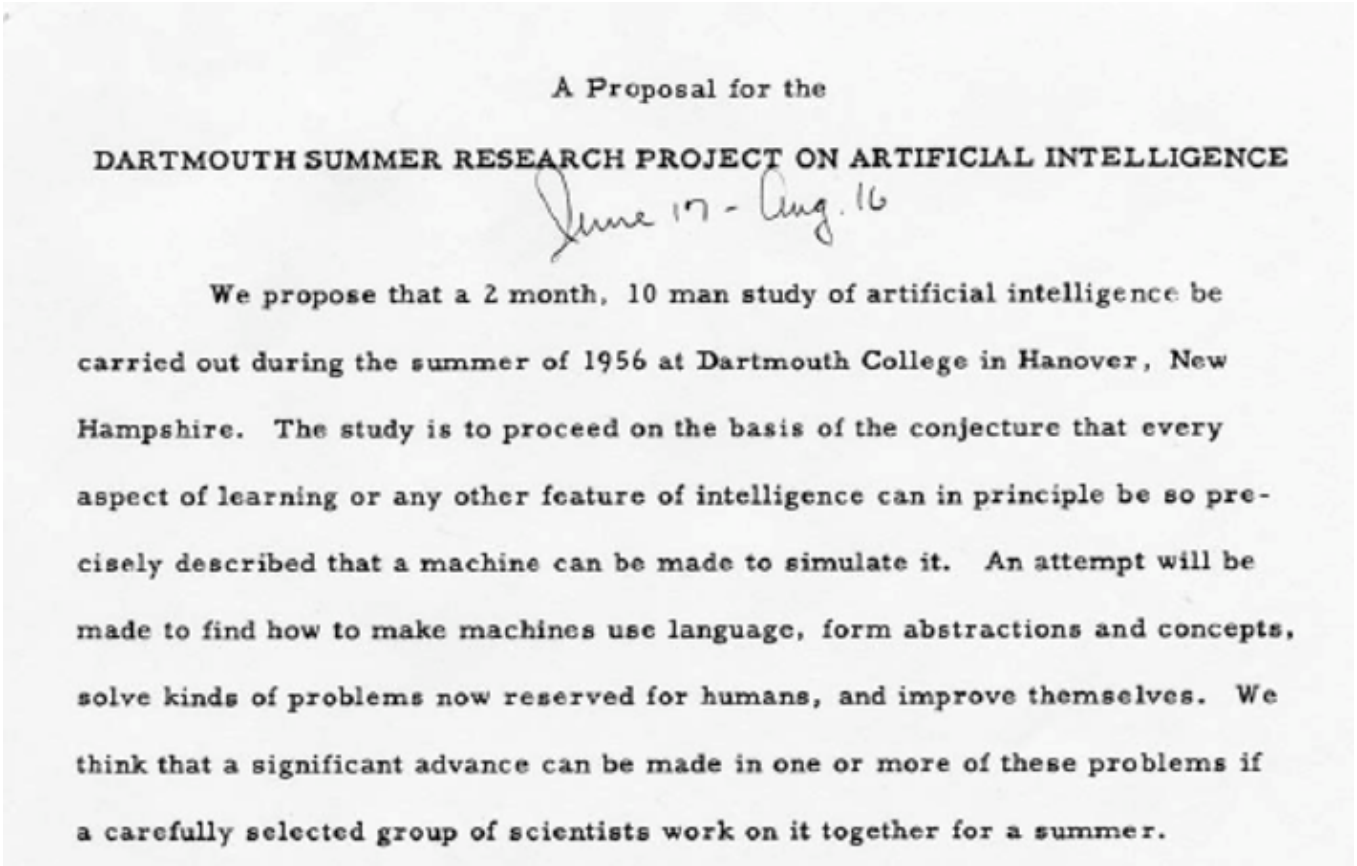
But then ...

- When its too good, you start suspecting
- Is it for real!!
- How can such thing happen?
- How far will it go?

The next thing you know, people are worrying about exactly how and when AI is going to doom humanity.

Is AI new?

Is AI new? A little history



A Proposal for the

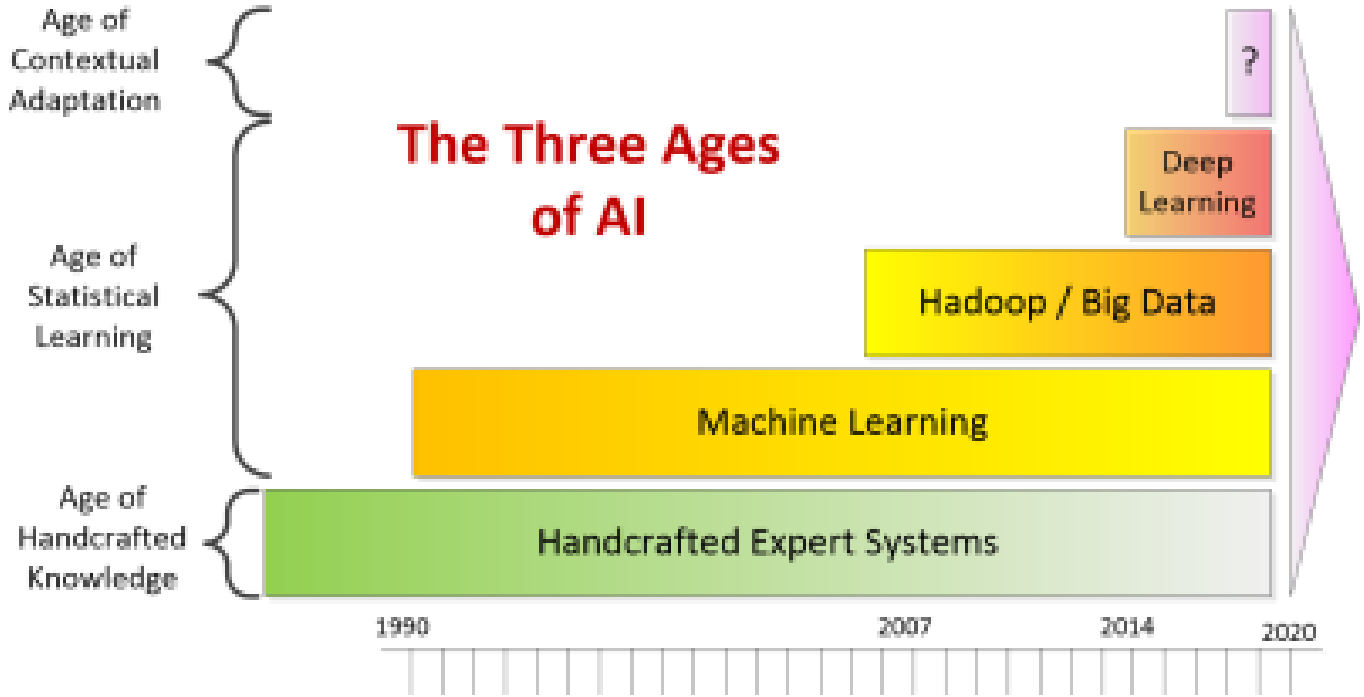
DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

June 17 - Aug. 16

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

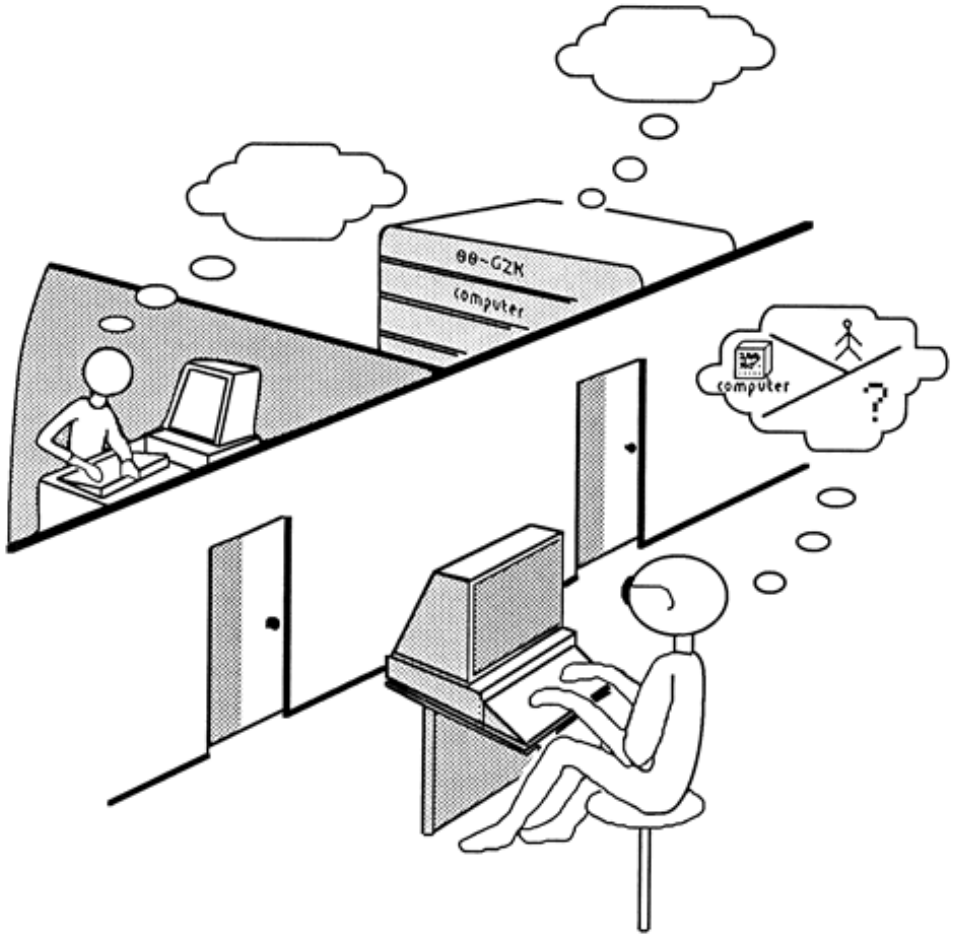
(Ref: John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon (1955))

Is AI new? A little history



(Ref: What Exactly is Artificial Intelligence and Why is it Driving me Crazy - William Vorhies)

Turing Test



Simplistically: If you cannot decide if you are talking to a human or a machine then AI has arrived. (Ref: What is Artificial Intelligence — Artificial Intelligence Tutorial For Beginners — Edureka)

Major AI Approaches

- Logic and Rules-Based Approach
- Machine Learning (Pattern-Based Approach)

Logic and Rules-Based Approach

- Representing processes or systems using logical rules
- Top-down rules are created for computer
- Computers reason about those rules
- Can be used to automate processes

Logic and Rules-Based Approach

Example : Expert Systems, Turbotax/Tally

- Personal income tax laws
- Represented as logical computer rules
- Software computes tax liability

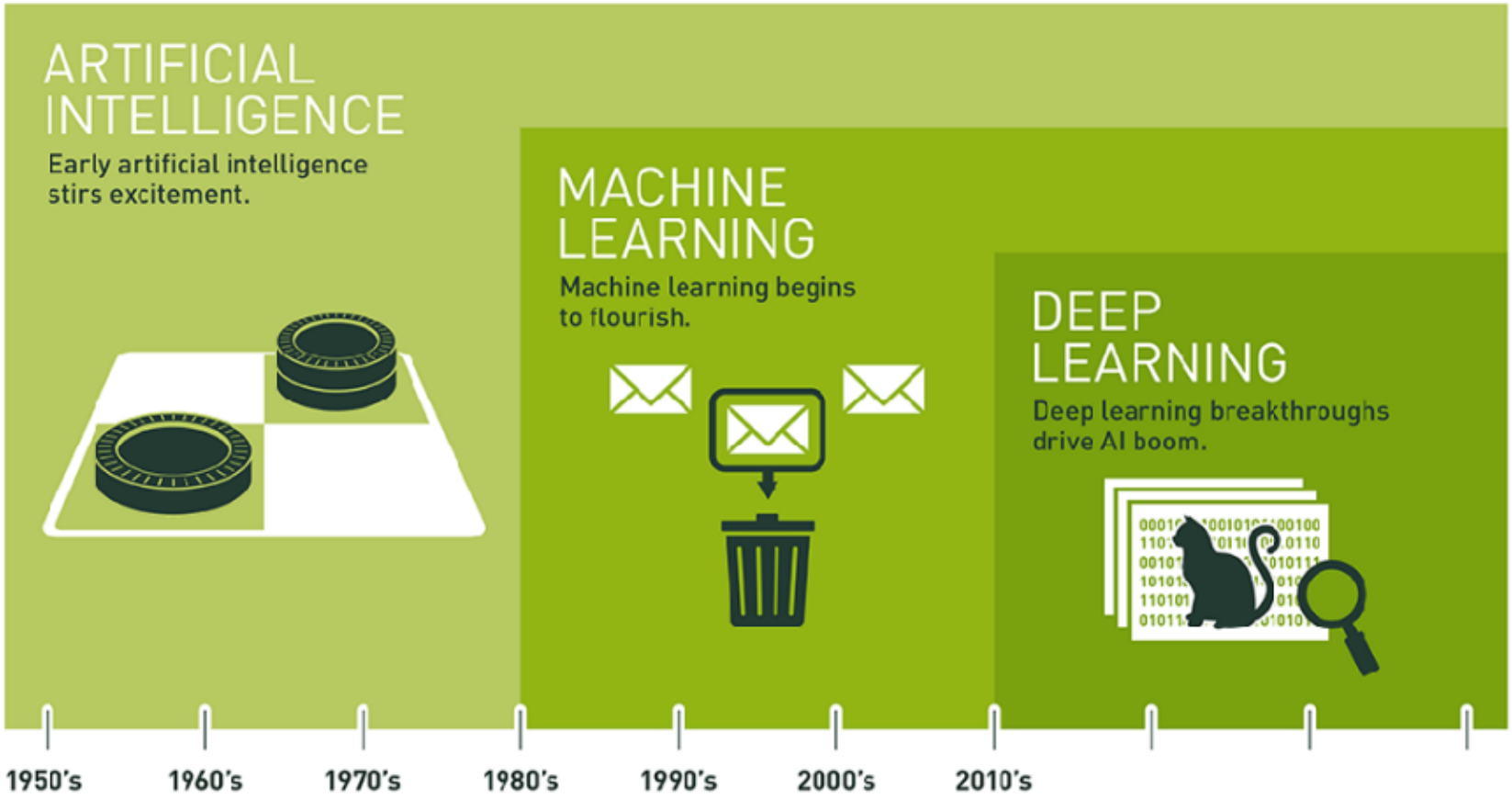
Machine Learning (Pattern based)

- Algorithms find patterns in data and infer rules on their own
- "Learn" from data and improve over time
- These patterns can be used for automation or prediction
- ML is the dominant mode of AI today
- Deep Learning is one set of methods within ML

Machine Learning (Pattern based)

- Learning from Data
- Pattern Detection
- Self-Programming/Automation

Relationship between AI, ML, DL



(Ref: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/)

Is AI a threat?

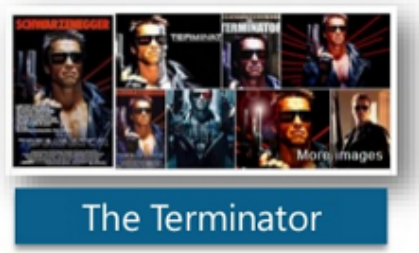
Is AI a threat?

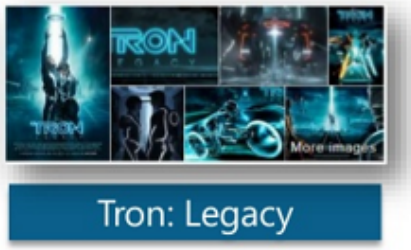
If you believe in what Elon Musk says, then YES.

(Ref: What is Artificial Intelligence — Artificial Intelligence Tutorial For Beginners — Edureka)

Is AI a threat?

If you believe in these movies, then YES.





Well, AI based War robots are not impossible anymore.

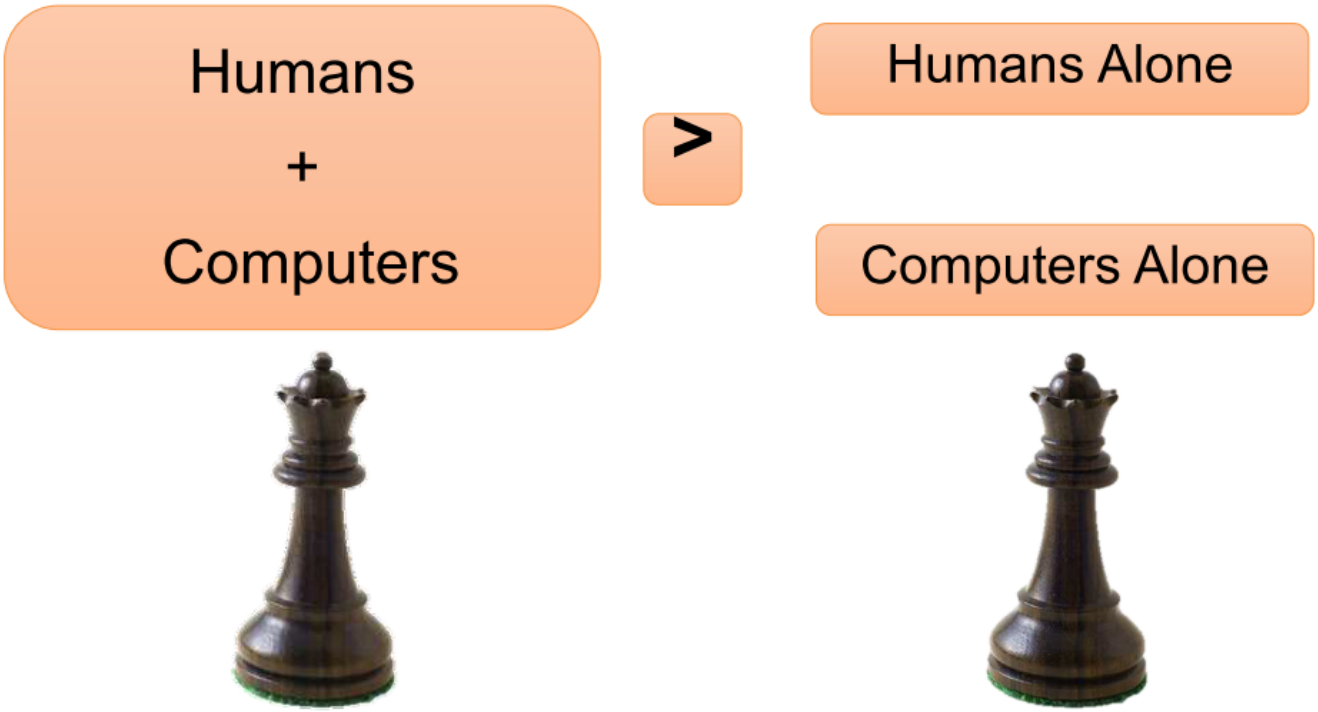
(Ref: What is Artificial Intelligence — Artificial Intelligence Tutorial For Beginners — Edureka)

Fear: Are we being replaced?

- Yes. in tasks that are repetitive
- But not which require complex thinking and creativity

Mostly

Technology Enhancing (Not Replacing) Humans



(Ref: "Artificial Intelligence Overview" - Harry Surden)

Limits on Artificial Intelligence

- Many things still beyond the realm of AI
- No thinking computers
- No Abstract Reasoning
- Often AI systems Have Accuracy Limits
- Many things difficult to capture in data
- Sometimes Hard to interpret Systems

Introduction to Machine Learning

How do we learn?

- What do we do when we have to prepare for an examination?
- Study. Learn. Imbibe. Take notes. Practice mock papers.
- Thus, prepare for the unseen test.

What is Learning?

"Learning is any process by which a system improves performance from experience."

- Herbert Simon, Turing Award 1975, Nobel in Economics 1978.

What is Machine Learning?

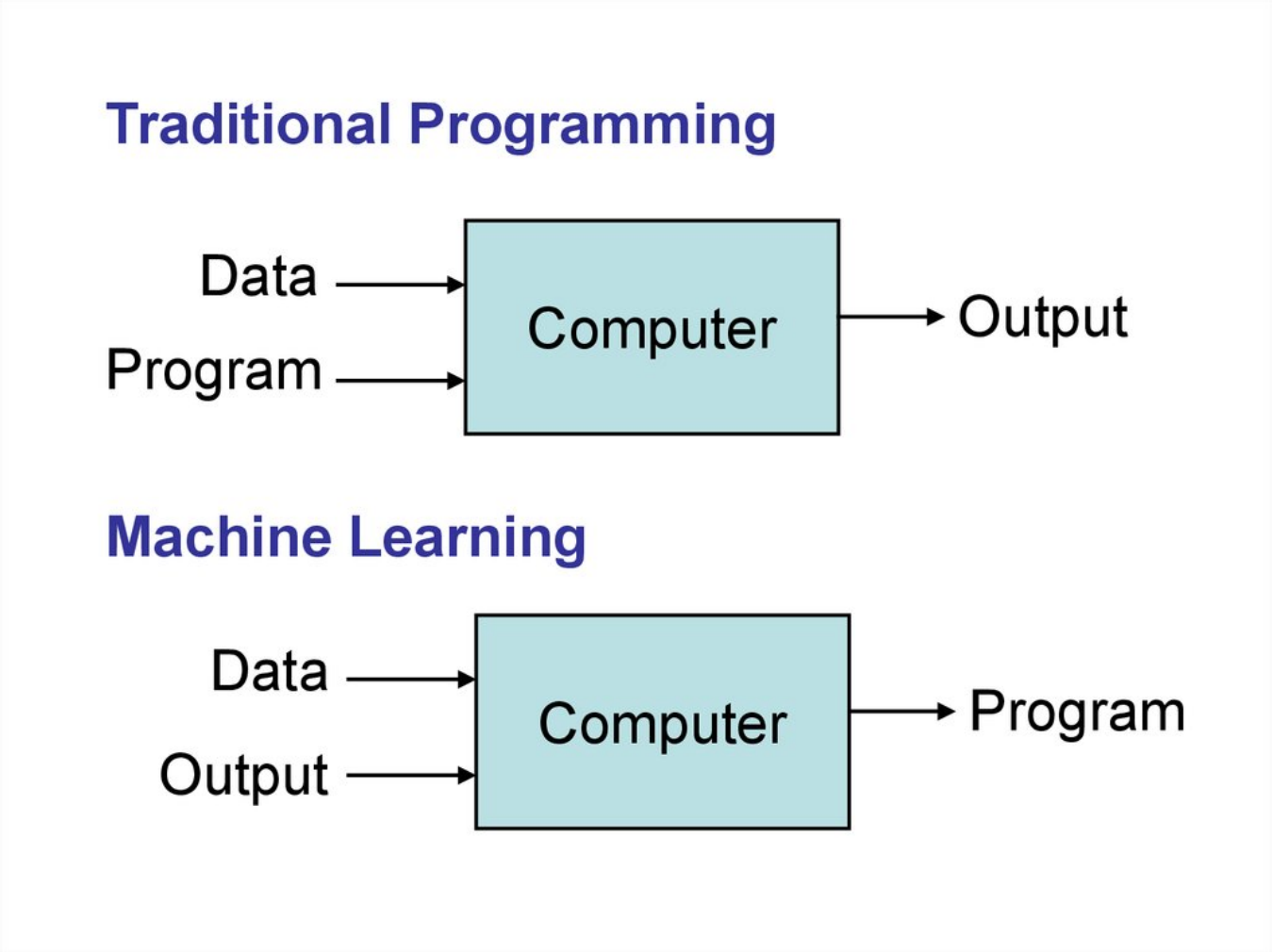
Machine learning is a type of artificial intelligence (AI) which:

- Learns function without being explicitly programmed.
- Can grow and change when exposed to new data.

So, What is Machine Learning?

- Ability of computers to “learn” from “data”
- Learn: Discover patterns, underlying structure
- Data: Comes from sensors, transactions, etc.

Traditional vs. Machine Learning?



Why Machine Learning?

- Problems with High Dimensionality
- Hard/Expensive to program manually
- Techniques to model 'ANY' function given 'ENOUGH' data.
- Job $$$

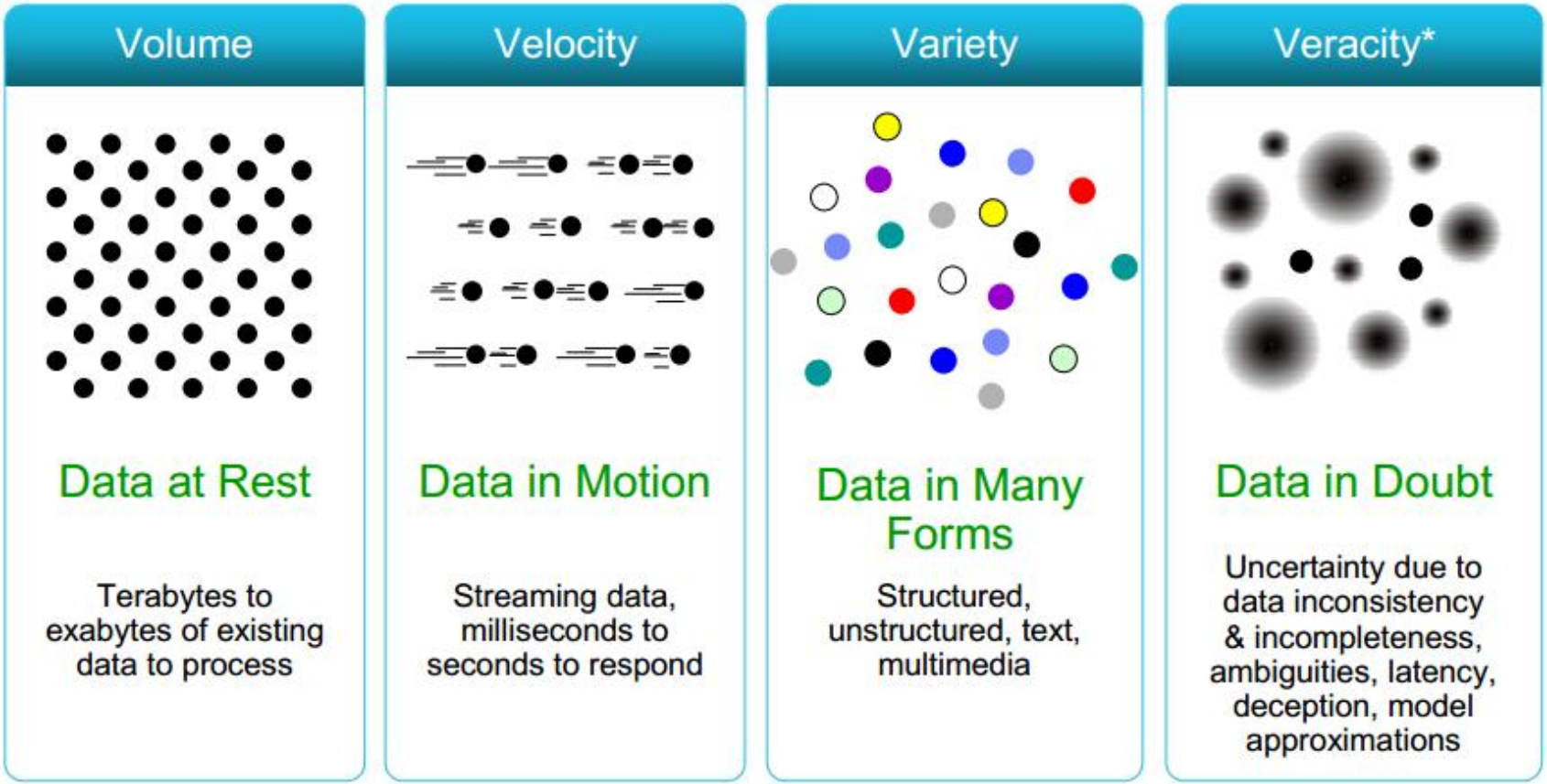
Why now?

- Flood of data (Internet, IoT)
- Increasing computational power
- Easy/free availability of algorithms
- Increasing support from industries

The storm: The Big Data is coming

- In 2012, HBR put Data Scientists on the radar
- "The Sexiest Job of the 21st Century".
- Industry, trying to be data-driven, than manual.

(Big) Data Characteristics



(Image Credit: http://www.rosebt.com/blog/data-veracity)

What's the answer?

AI-ML-DL

- Machines showing intelligence of Humans
- Machine Learning: part of AI
- Logic is not programmed by hand,
- Gets emerged in training with data.

Types of Machine Learning

Two kinds of learning

- Supervised
- Unsupervised

Supervised

- Training data with correct answers
- Both used to train the model
- Then apply unseen data on model

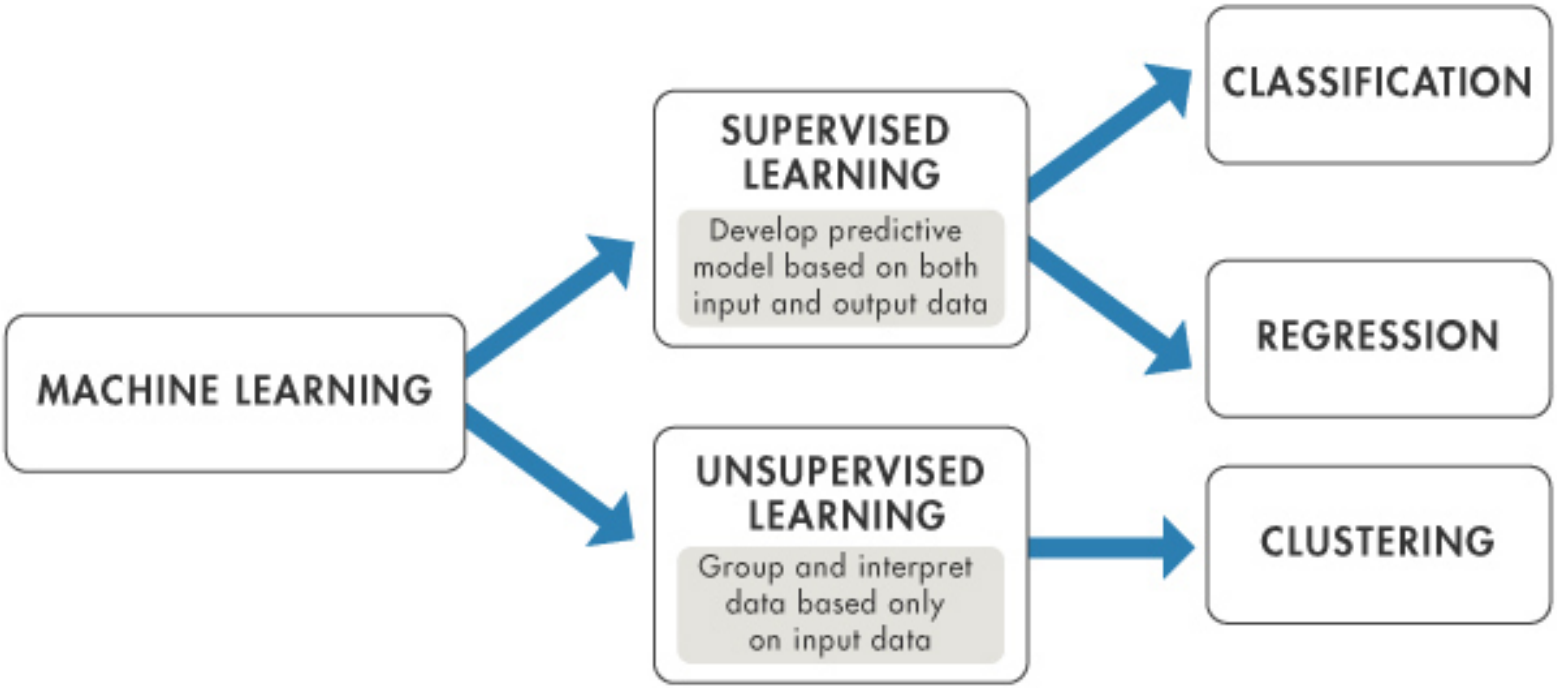
Unsupervised

- Training data with no answers
- Extract patterns, groups

Some types of algorithms

- Prediction: predicting a continuous variable from data
- Classification: assigning records to predefined groups
- Clustering: splitting records into groups based on similarity
- Association learning: seeing what often appears together

Machine Learning Learning Algorithms



(Reference: Machine Learning in MATLAB - MATLAB & Simulink - MathWorks)

Machine Learning Learning Algorithms

- Is this A or B? : Classification algorithms
- Is this weird? : Anomaly detection algorithms
- How much—or—How many? : Regression algorithms
- How is this organized? : Clustering algorithms, Dimensionality reduction
- What should I do next? : Reinforcement learning algorithms

(Ref: Brandon Rohrer's breakdown of the "5 questions data science answers")

Classification

- **Description:** Identifying the category an object belongs to.
- **Applications:** Spam detection, Image recognition.
- **Algorithms:** SVM, nearest neighbors, random forest, Logistic Regression

Regression

- **Description:** Predicting a continuous-valued attribute associated with an object.
- **Applications:** Drug response, Stock prices.
- **Algorithms:** Linear Regression

Clustering

- **Description:**Automatic grouping of similar objects into sets.
- **Applications:** Customer segmentation, Grouping experiment outcomes
- **Algorithms:** k-Means

Dimensionality Reduction

- **Description:** Reducing the number of random variables to consider.
- **Applications:** Visualization, Increased efficiency
- **Algorithms:** PCA, Singular Value Decomposition

Popular Algorithms in Machine Learning

- Linear, Logistic Regression
- Decision Trees
- SVM - Support Vector Machines, Naive Bayes
- K-Means

Applications of Machine Learning

Everyday Applications of Machine Learning

- Face Recognition (Facebook)
- Spam recognition in Emails
- Recommender Systems
- Feelings Analysis, Sentiments
- Natural language: Translate a sentence from Hindi to English, question answering, etc.
- Speech: Recognize spoken words, speaking sentences naturally
- Game playing: Play games like chess
- Robotics: Walking, jumping, displaying emotions, etc.
- Driving a car, flying a plane, navigating a maze, etc.

Cool-down: Summary

SO . . .

- What is Machine learning, after-all?
- Its usage in your domain?

Python: Quick Introduction

Guess

What are the differences with the programming languages you know?

```
x = 34 - 23
y = ‘‘Hello’’
z = 3.45
if z == 3.45 or y == ‘‘Hello’’:
  x = x + 1
  y = y + ‘‘ World’’
print(x)
print(y)
```

Why Python?

- Readability
- Ease of use
- “Fits in your head”
- Incremental sense of accomplishment, aka“gets things done”
- Good libraries
- Deployment, aka “Lookie what I did!”

Truths about Good Programmers

- Lazy (in a good way)
- Just want things to work
- Spoiled kids who just want to have fun
- And sometimes create Fortune 100 companies

One Truth About Python

- Power scales with the ability of the programmer
- Novices can do simple things
- Really bright people build tools
- Novices leverage these tools
- Lone sys-admins <3 perl
- Mavericks in small work-groups <3 Python

Brief History of Python

- Invented in the Netherlands, early 90s by Guido van Rossum
- Named after Monty Python (a British comedy group, the language has a playful approach)
- Open sourced from the beginning
- Considered a scripting language, but is much more
- Used by Google from the beginning
- Increasingly popular

Syntax

What is Python?

- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.
- Python is simple and easy to learn.
- Python is open source, free and cross-platform.
- Python provides high-level built in data structures.
- Python is useful for rapid application development.
- Python can be used as a scripting or glue language.
- Python emphasizes readability.
- Python supports modules and packages.
- Python bugs or bad inputs will never cause a segmentation fault.

The Python shell, I

- Python can be run from “shell”, IDE, Notebook
- Start writing commands/expressions at the >>> prompt.
- Shell/Command Line:

```
> python

Python 3.5.3 | packaged by conda-forge | (default, May 12 2017, 16:16:49)
     [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

The Python shell, II

- Expressions are evaluated and the result is printed:
- Line continuation with
- The prompt changes to '...' on continuation lines and for loops, function definitions, etc.

```
>>> 2+2
4

>>> "hello" + \
... " world!"
'hello world!'
```

Overall Syntax

- Comments are indicated with "#"
- Multiple statements on the same line are separated with ";"
- No semicolon at the end of lines.
- Scope is obtained through indentation.
- Always indent next line if ":" is at the end of current line.
- One script is can be run or imported by other modules.

Assignment

- Assignment creates references, not values: `tmp = "hello"; tmp = 10` the first string will be deallocated
- As in C programming: `x += 1` is valid
- Pre/post increment/decrements: `x++; ++x; x--;--x` are invalid
- Multiple assignment (references to a unique object):`x=y=z=1`
- Multiple assignments: `(x,y,z)=(3.5,5.5,'string')`
- Example of swapping variables value: `(x,y)=(y,x)`

Built-in object types

- Numbers : `3.1415, 1234, 999L, 3+4j`
- Strings : `'spam', ''guido's''`
- Lists : `[1, [2, 'three'], 4]`
- Dictionaries :`{'food':'spam', 'taste':'yum'}`
- Tuples : `(1,'spam', 4, 'U')`
- Sets: `{1,2,3,'foo','bar'}`

Numbers

- Integers : `1234, -24, 0`
- Unlimited precision integers : `999999999999L`
- Float : `3.1415, 2.7122`
- Oct and hex :`0177, 0x9ff`
- Complex : `3+4j, 3.0+4.0j, 3J`

Strings (immutable sequences)

- single quote `s1 = 'egg'`
- double quotes `s2 = ''spam's''`
- triple quotes `block = '''...'''`
- concatenate `s1 + s2`
- repeat `s2 * 3`
- index,slice `s2[i], s2[i:j]`
- length `len(s2)`
- formatting `''a {} parrot''.format('dead')`
- iteration `for x in s2 # x loop through each character of s2`
- membership `'m' in s2`

Lists

- Ordered collections of arbitrary objects
- Accessed by offset
- Variable length, heterogeneous, arbitrarily nest-able
- Mutable sequence
- Arrays of object references

Lists operations

- empty list `L = []`
- four items `L2 = [0, 1, 2, 3]`
- nested `L3 = ['abc', ['def', 'ghi']]`
- index `L2[i]`, `L3[i][j]`
- slice `L2[i:j]`, length `len(L2)`
- concatenate `L1 + L2`, repeat `L2 * 3`
- iteration `for x in L2`, membership `3 in L2`
- methods `L2.append(4)`, `L2.sort()`, `L2.index(1)`, `L2.reverse()`
- shrinking `del L2[k]`, `L2[i:j] = []`
- assignment `L2[i] = 1`, `L2[i:j] = [4,5,6]`

Dictionaries

- Accessed by key, not offset
- Unordered collections of arbitrary objects
- Variable length, heterogeneous, arbitrarily nest-able
- Of the category mutable mapping
- Tables of object references (hash tables)

Dictionaries operations

- empty `d1 = {}`
- two-item `d2 = {'spam': 2, 'eggs': 3}`
- nesting `d3 = {'food': {'ham': 1, 'egg': 2}}`
- indexing `d2['eggs'], d3['food']['ham']`
- methods `d2.keys(), d2.values()`
- length `len(d1)`
- add/change `d2[key] = new`
- deleting `del d2[key]`

tuples

- They are like lists but immutable. Why Lists and Tuples?
- When you want to make sure the content won't change.

Files

- input `input = open('data', 'r')`
- read all `S = input.read()`
- read N bytes `S = input.read(N)`
- read next `S = input.readline()`
- read in lists `L = input.readlines()`
- output `output = open('/tmp/spam', 'w')`
- write `output.write(S)`
- write `strings output.writelines(L)`
- close `output.close()`

Comparisons vs. Equality

- `L1 = [1, ('a', 3)]`
- `L2 = [1, ('a', 3)]`
- `L1 == L2` is 1
- The `==` operator tests value equivalence
- `L1 is L2` is 0
- The `is` operator tests object identity

if, elif, else

```
if not done and (x > 1):
  doit()
elif done and (x <= 1):
  dothis()
else:
  dothat()
```

while, break

```
while 1:
  line = ReadLine()
  if len(line) == 0:
    break
```

for

```
# String:
for letter in 'hello world':
  print(letter)
# List:
for item in [12, 'test', 0.1+1.2J]:
  print(item)

# Range with bounds and step:

for i in range(2,10,2):
  print(i)

# Equivalent to the C loop:

for (i = 2; i < 10; i+=2){
  printf("%d\n",i);
}

```

pass

Temporary filler, the stub. Functions, for loop, wherever there is ":", then on the indented next line *pass* can be put.

```
1 pass
```

errors and exceptions

- NameError attempt to access an undeclared variable
- ZeroDivisionError division by any numeric zero
- SyntaxError Python interpreter syntax error
- IndexError request for an out-of-range index for sequence
- KeyError request for a non-existent dictionary key
- IOError input/output error
- AttributeError attempt to access an unknown object attribute

```
try:
  f = open('blah')
except IOError:
  print(‘could not open file’)

```

Functions

- Functions can return any type of object.
- When nothing is return the None object is returned by default.
- Multiple values can be returned.
- Anonymous functions “lambda”.
- Parameters can have default arguments.
- Variable-length arguments are supported.

```
def test(a,b=2,d=func):
    return d(a,b)

test(3)
test(b=4,a=3)
test(1,2,lambda x,y: x*y)
test(1,2,g)

```

Modules, namespaces and packages

- A file is a module, e.g. ‘myio.py’, with a function ’load’
- To use that function from another file:
- Code in ’myio.py’ will be in the ’myio’ namespace.
- Selective import:
- Packages are bundle of modules.

```
import myio
myio.load()

from myio import load
load()
```

Class

```
class Cone(SomeParantClass):
  def __init__(self,d0,de,L):
    self.a0 = d0/2
    self.ae = de/2
    self.L = L
  def __del__(self):
    pass
  def radius(self,z):
    return self.ae + (self.a0-self.ae)*z/self.L
  def radiusp(self,z):
    return (self.a0-self.ae)/self.L

c = Cone(0.1,0.2,1.5)
c.radius(0.5)

```

Standard library core modules

- **os** file and process operations.
- **time** dates and times related functions.
- **string** commonly used string operations.
- **re** regular expressions.
- **copy** allow to copy object.

other library modules

- **Tkinter**: Tk GUI toolkit (cross-platform).
- **NumPy**: Numerical array processing.
- and many many more . . .
- Visit https://pypi.python.org/pypi for a comprehensive listing.

Thanks ... yogeshkulkarni@yahoo.com