머신러닝을 활용한 리얼월드 데이터 분석

Machine Learning with Real World Data

홍참길

charmgil@handong.edu

홍참길 (한동대학교)

- Interests: *Machine learning*, *Outlier detection*, *Artificial intelligence in medicine*

- *PhD. in Computer Science, University of Pittsburgh*
  - *Dissertation title:* ***Multivariate Data Modeling and Its Application to Conditional Outlier Detection*** (Advisor: Dr. Milos Hauskrecht)

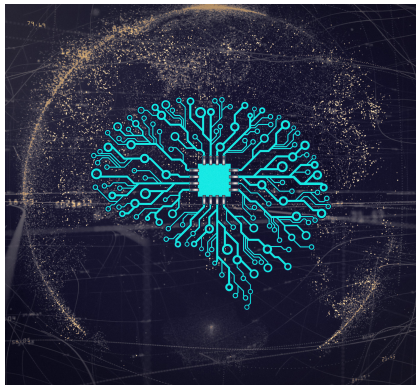
- *ML Projects*
  - NOMA (2015-current): Real-time clinical outlier (mistakes) monitoring and alerting
  - HiDenIC (2011-2013): Outlier detection on High-density Intensive Care data
  - PCP (2011-2012): Post-cardiac surgical patients
  - MCODE (2014-2018): Multivariate conditional outlier detection
  - CCF (2013-2015): Multi-label classification with the *Classifier Chains* Family
  - CARE (2015): Multi-target regression exploiting canonical correlation structures
  - RBDT (2014): Rényi entropy-based decision theory

Agenda

- Introduction
- Getting started with Python (on Google Drive)
- Classification with *k*-Nearest Neighbors

Introduction

- What comes to your mind when you hear the following buzzwords?
  - Artificial intelligence
  - Machine learning
  - Deep learning



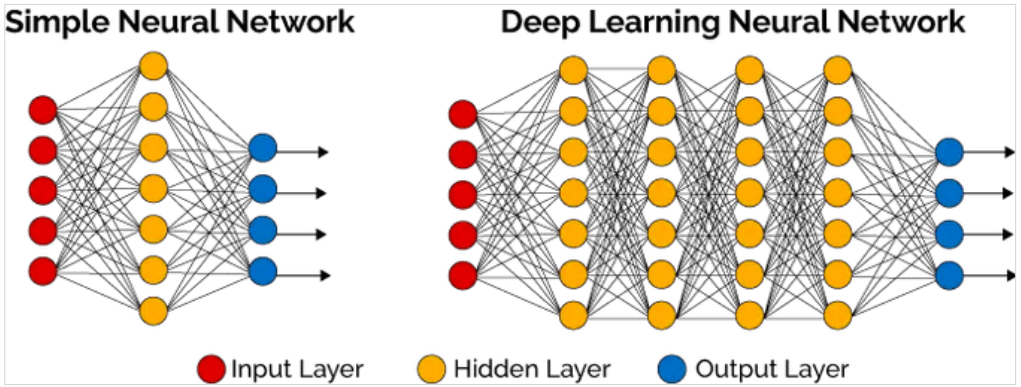
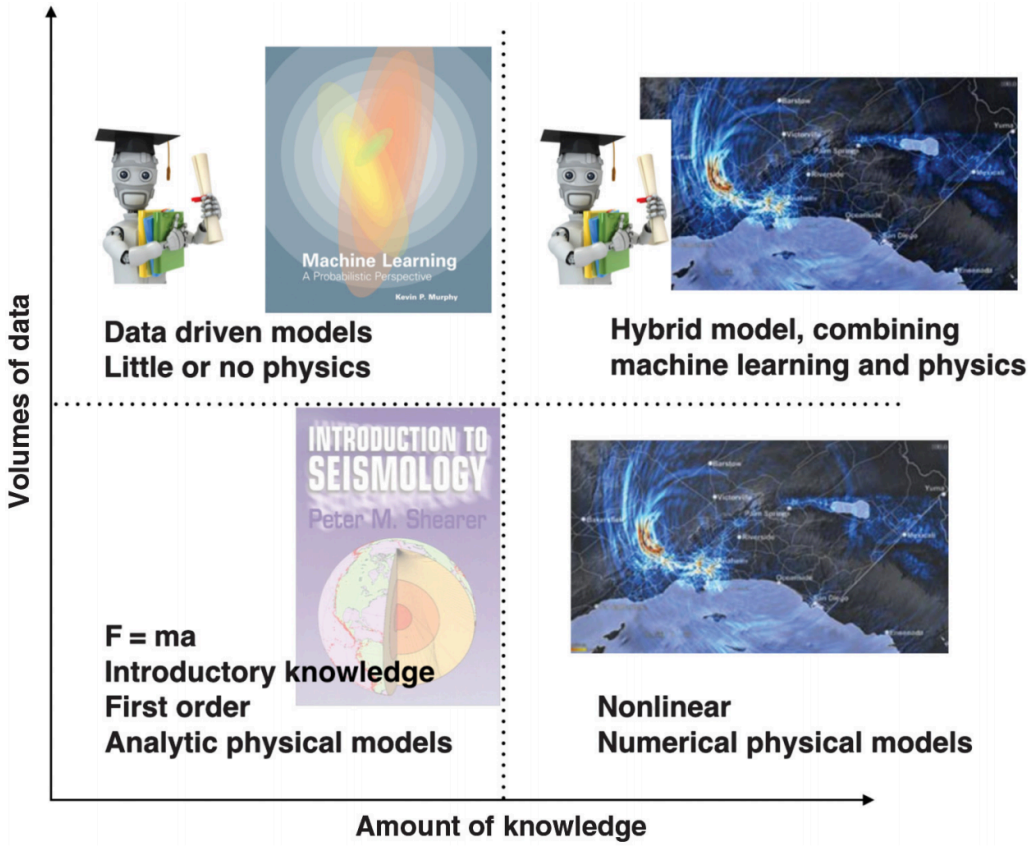
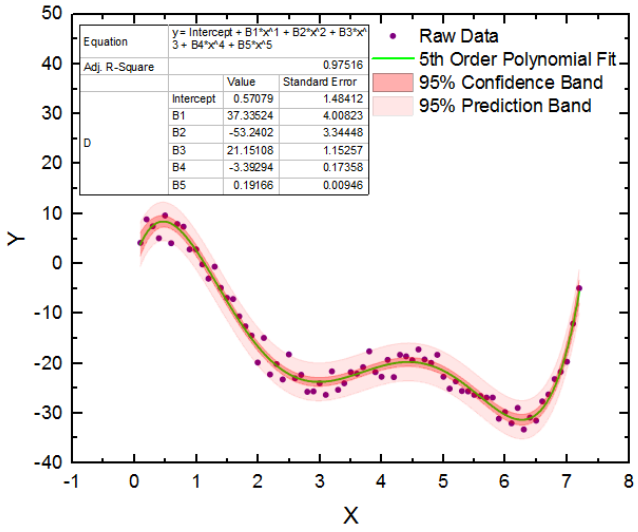


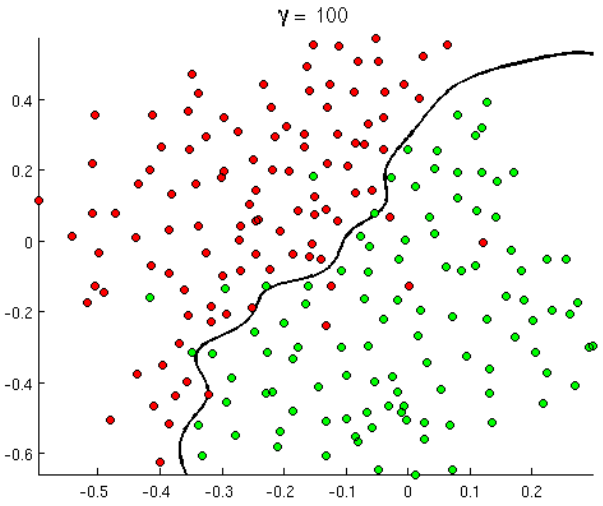
Image sources: https://www.flickr.com/photos/mikemacmarketing/42271822770
https://www.einfochips.com/blog/how-to-develop-machine-learning-applications-for-business/
https://medium.com/@rgrgrajat1/mnist-vs-mnist-how-i-was-able-to-speed-up-my-deep-learning-11c0787e6935

Introduction

- Traditional vs ML approaches







Introduction

- Artificial intelligence (AI)
  - Technologies that enable computers to mimic human behavior
  - Engineering of making intelligent machines and programs
- Machine learning (ML)
  - A subset of AI techniques that use statistical methods to enable machines to improve with experiences (data)
  - A scientific or engineering discipline that aims to learn without being explicitly programmed
- Deep learning
  - A subset of ML
  - Learning based on deep neural networks

Introduction

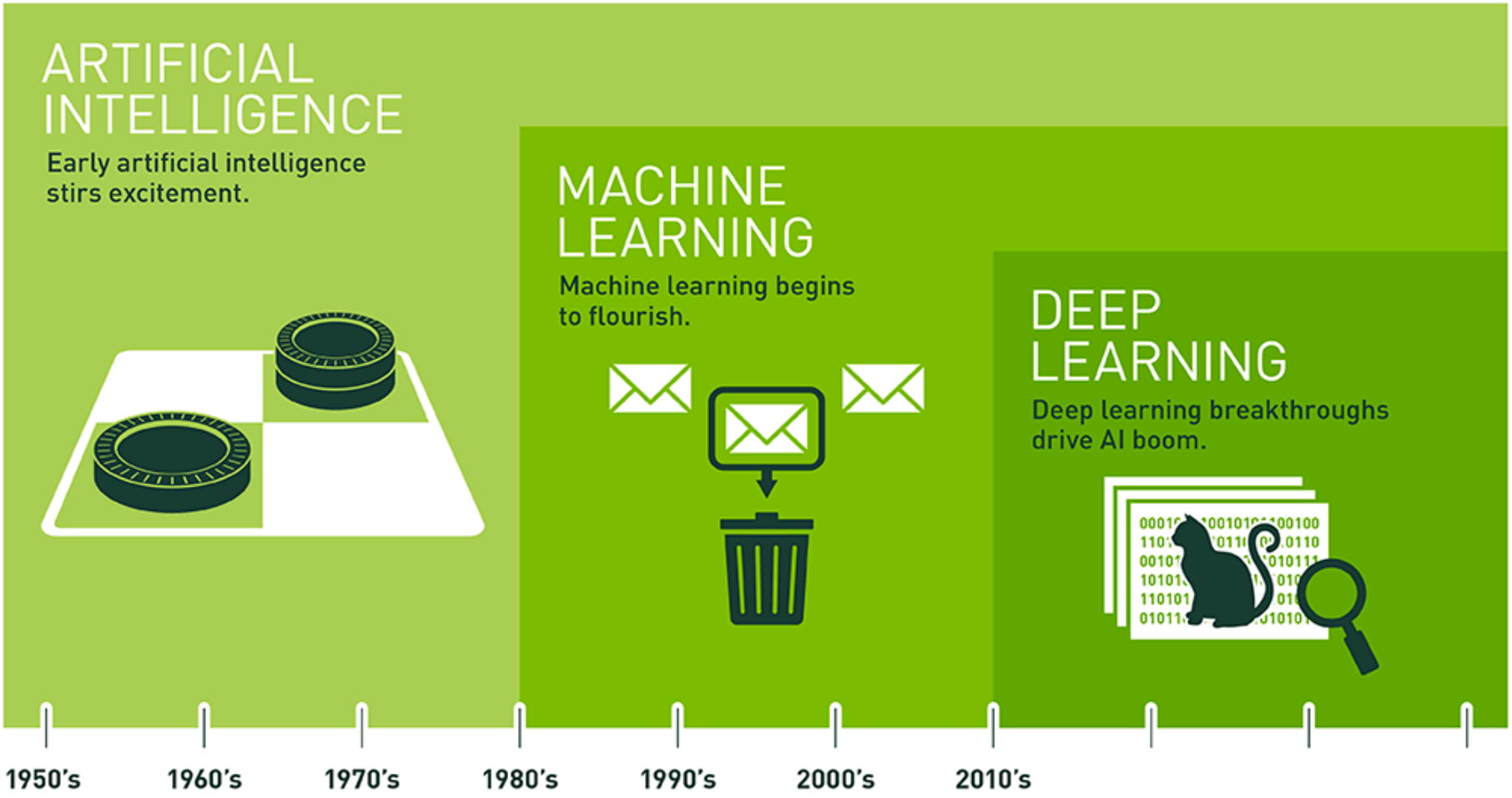


Image source: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

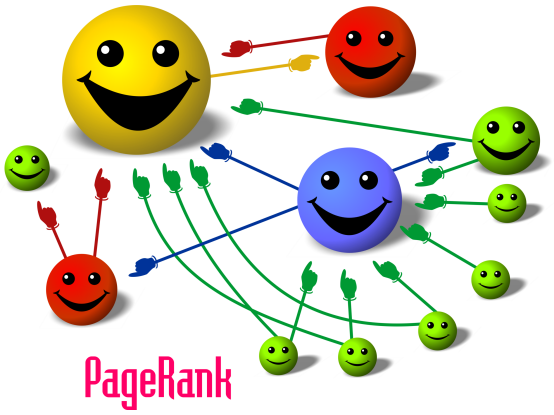
Introduction

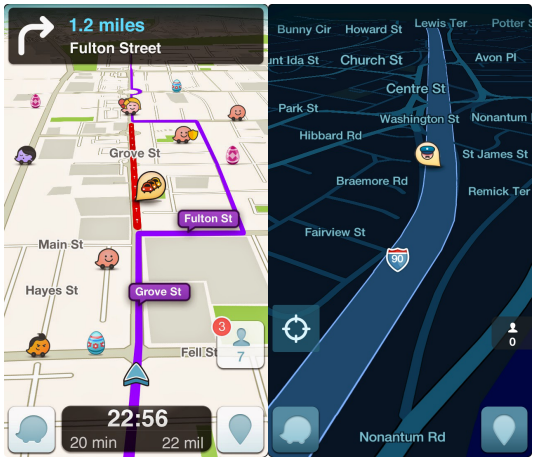
- ML in our life

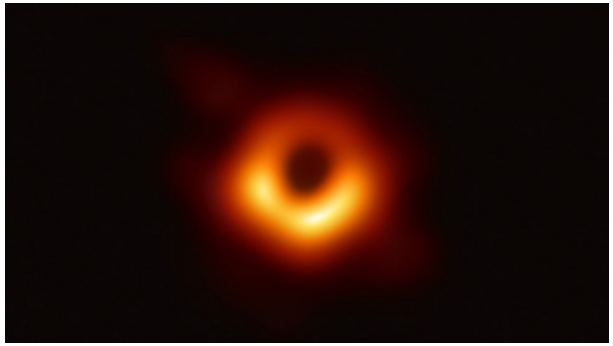
Image sources: https://www.ansys.com/blog/challenges-level-5-autonomous-vehicles
https://www.uber.com/blog/pennsylvania/new-wheels/
https://www.pocket-lint.com/phones/buyers-guides/120309-best-smartphones-and-flagship-mobile-phones-to-buy
https://www.smartly.ai/blog/tag/smart-speakers/
https://9to5google.com/guides/nest/

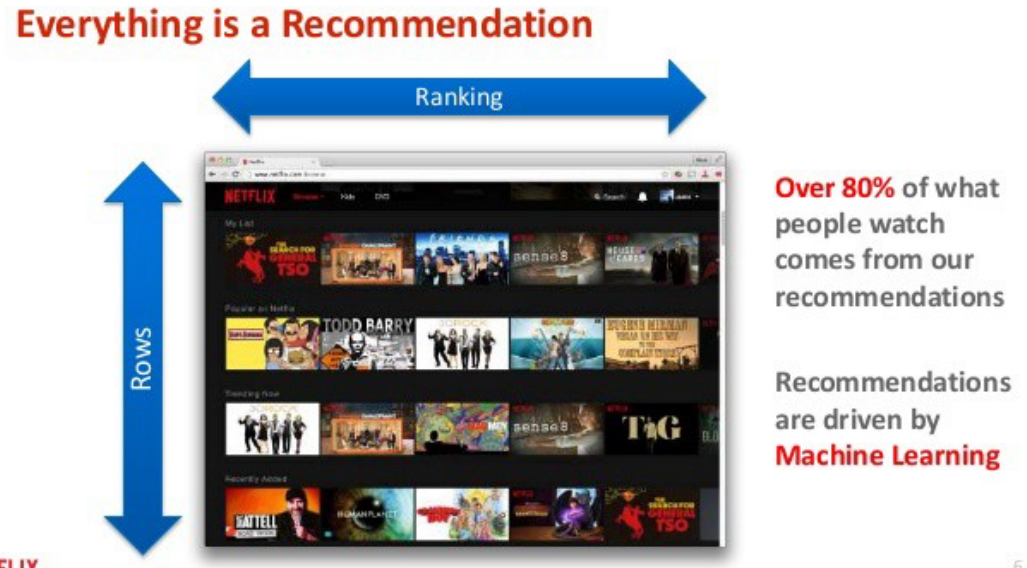
Introduction

- ML in our life



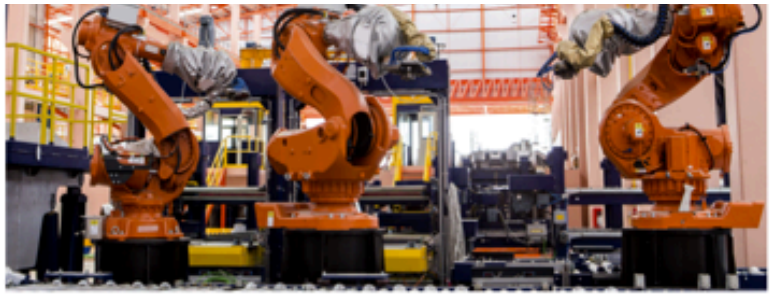






Introduction

- ML in our life



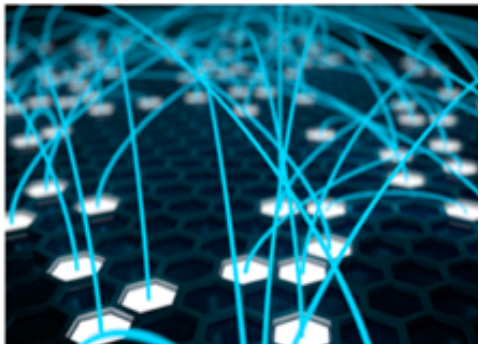
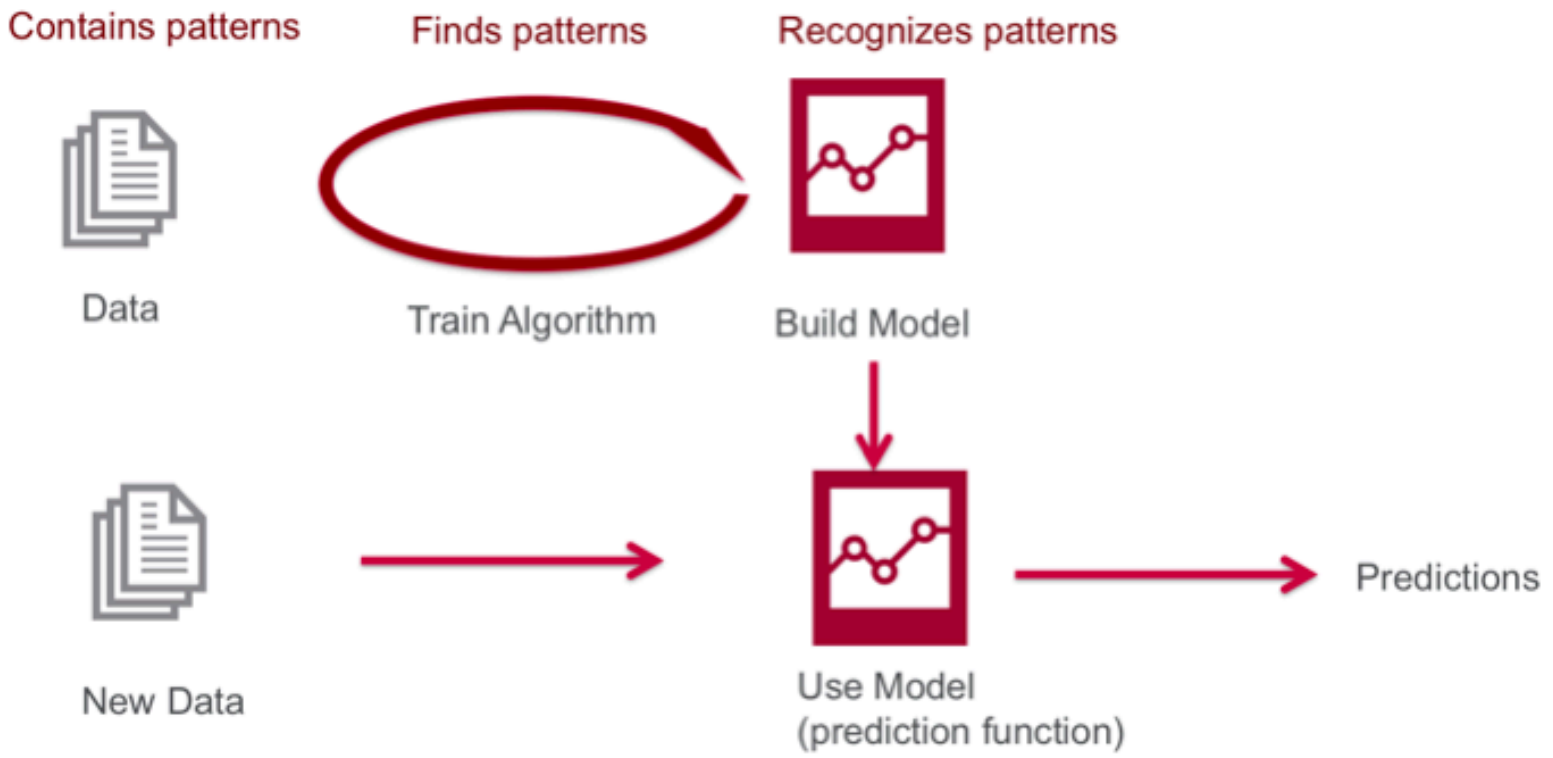



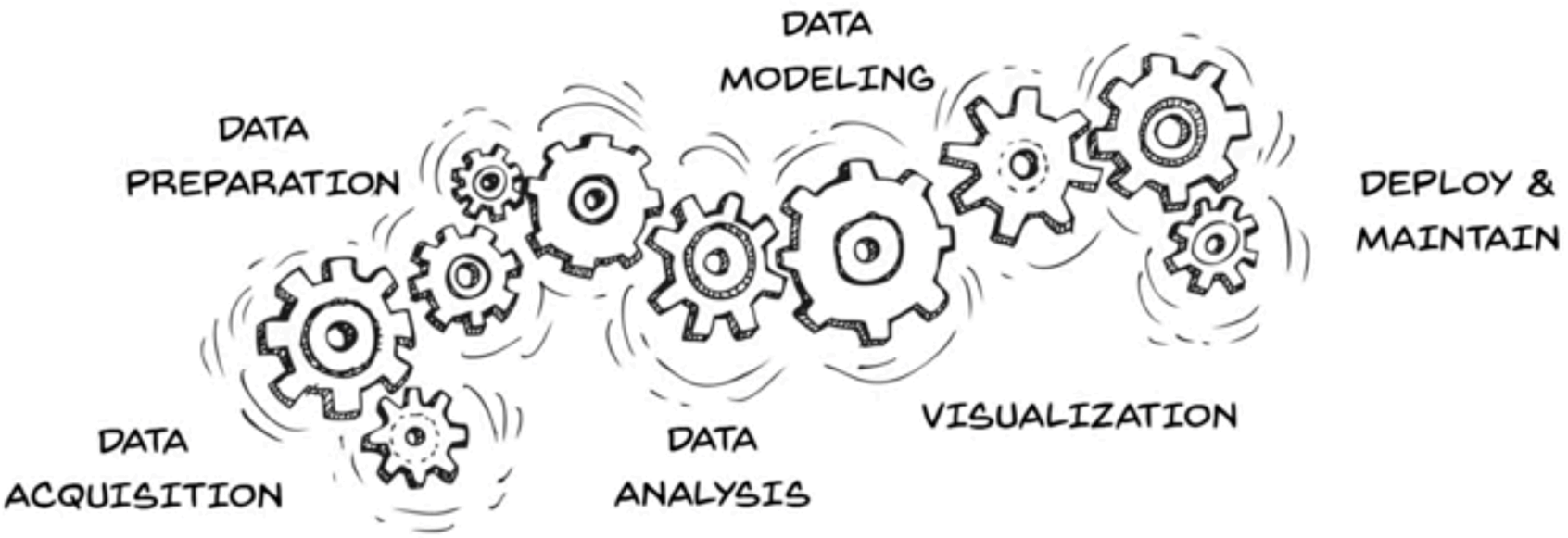
Image sources: https://mapr.com/blog/top-technology-trends-machine-learning-event-driven-microservices-dataops-and-cloud-to-edge/

Common ML Workflow

- In a narrow scope, ML refers to…



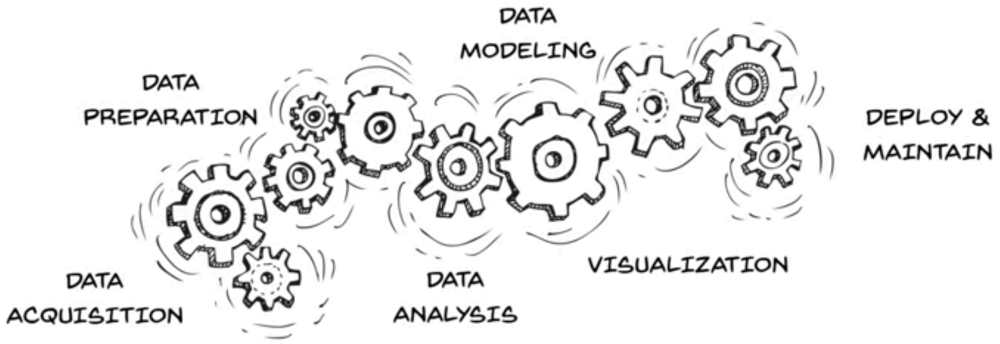
Common ML Workflow



- Phase 1: Acquisition and Preparation
- Phase 2: Analysis, Modeling, and Visualization
- Phase 3: Deployment and Maintenance

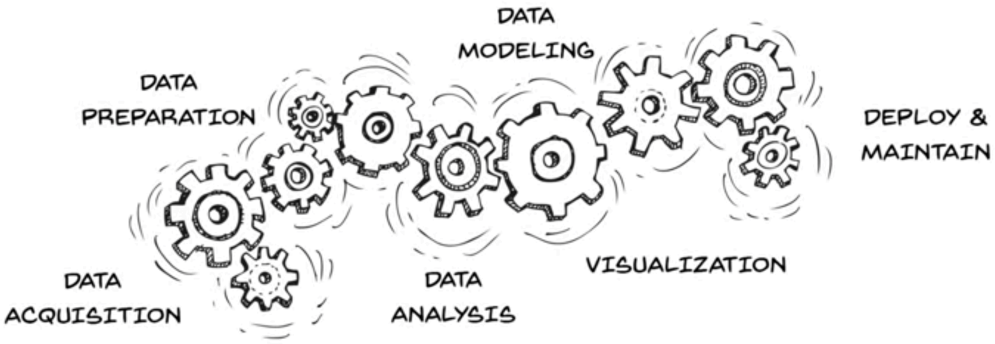
Common ML Workflow

ETL
Extract-Transform-Load



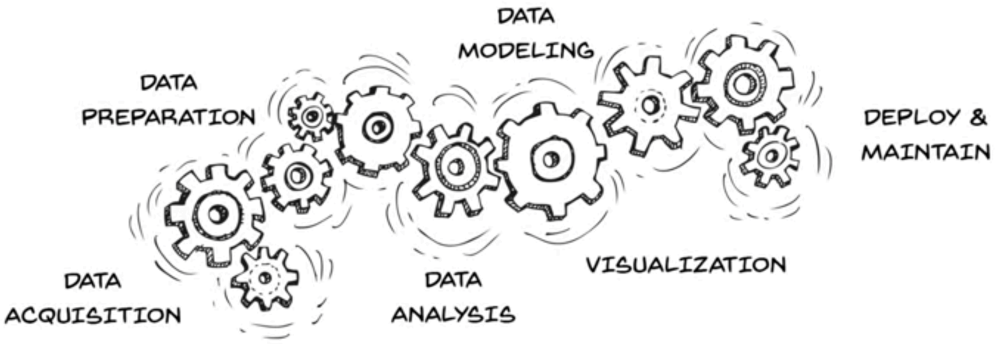
- *Acquisition* - where data is gathered/collected from various sources
  - Data records, sensors, activity trackers (apps), social media platforms
  - Experiments, surveys, meta-data analysis
  - Manual collection from non-digital sources
- *Preparation* - where data is cleaned, preprocessed, structured and eventually becomes a dataset
  - Removing errors, mistakes, duplicates, and inconsistencies in data
  - Data curation or annotation
  - Data integration - combining data from different sources

Common ML Workflow

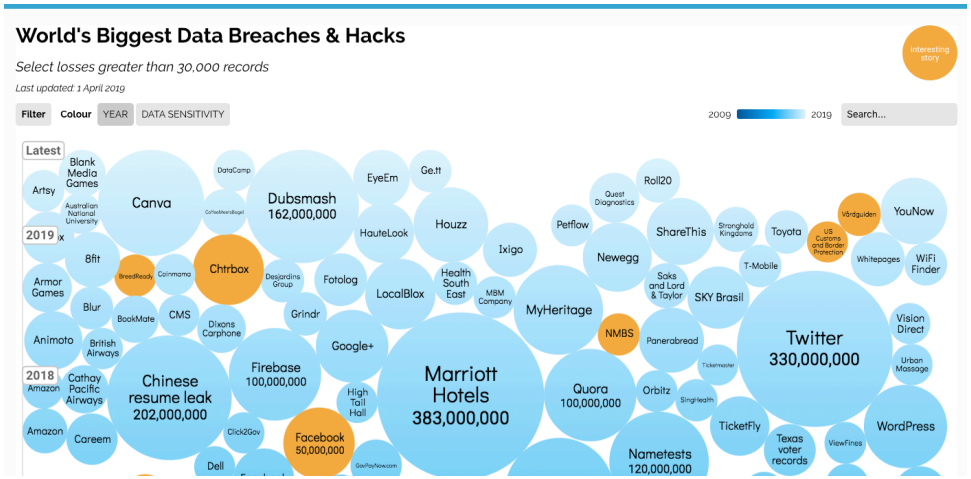


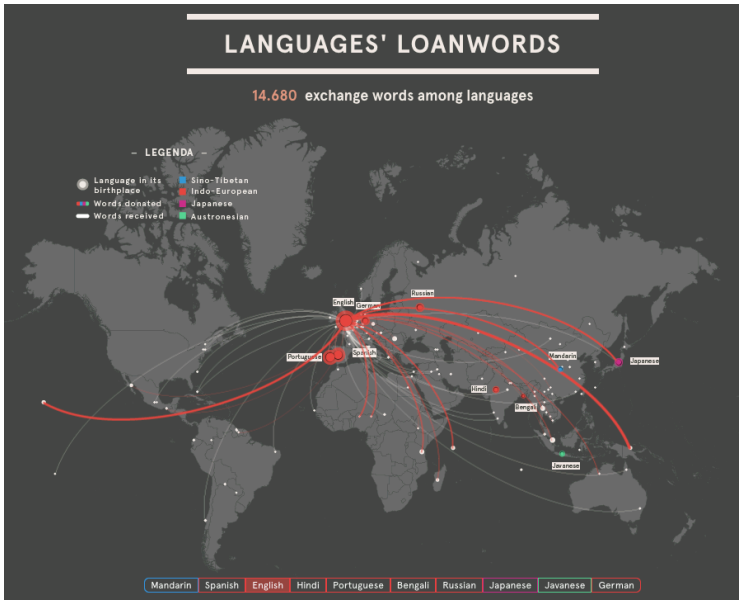
- ***Analysis*** - where data is evaluated to run and customize reports (to better understand data)
  - Various queries and data mergers are applied to tell a better and more informed story than when you look at each source independently
- ***Modeling*** - where data is patternized and generalized as models
  - Models explain the general patterns that frequently observed in data
  - Models are often used to make predictions or inferences

Common ML Workflow



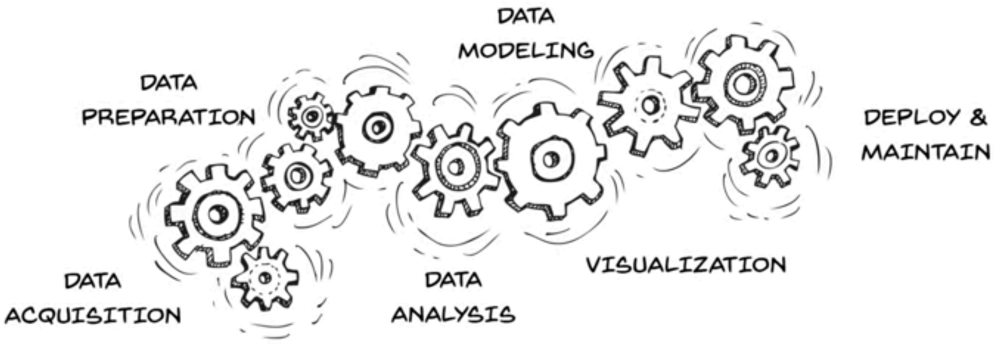
- *Visualization* - where data is visualize to provide intuitive overview; *E.g.*,



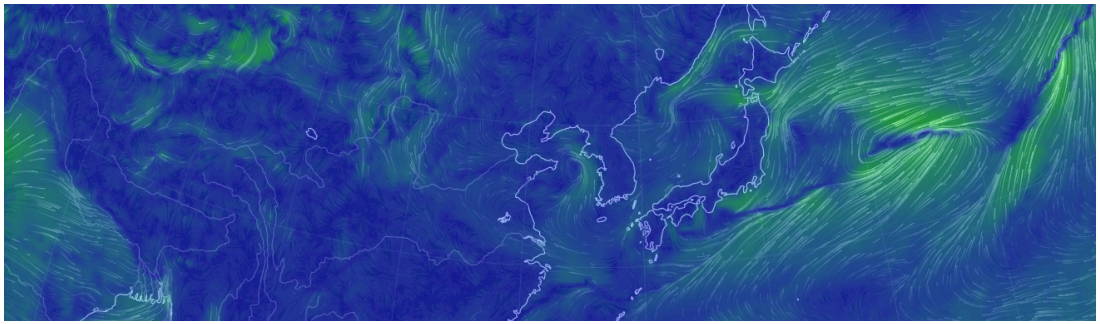


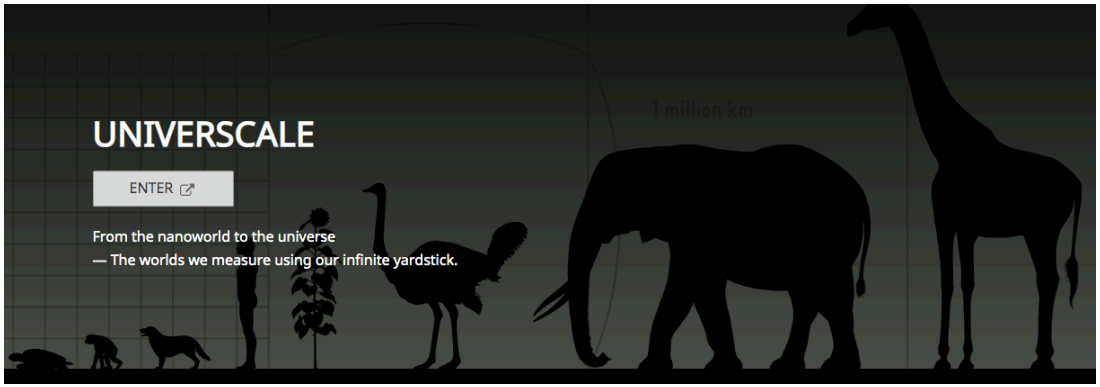
Sources: https://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/
http://www.puffpuffproject.com/languages.html

Common ML Workflow



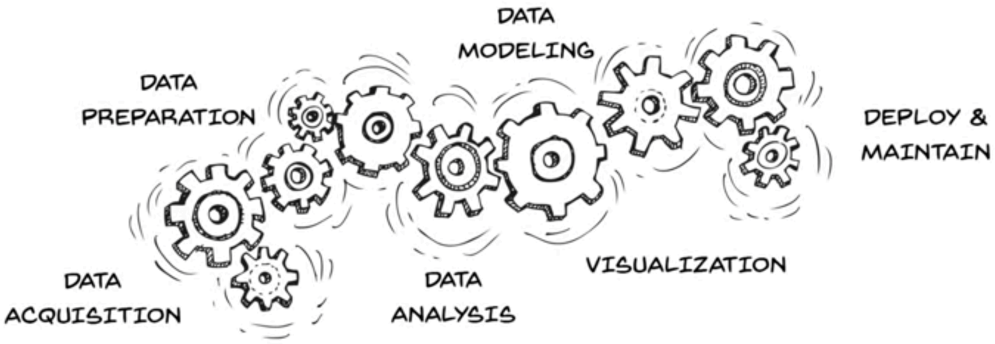
- *Visualization* - where data is visualize to provide intuitive overview; *E.g.*,





Sources: https://earth.nullschool.net/#current/wind/surface/level/orthographic=126.19,33.19,979
https://www.nikon.com/about/sp/universcale/

Common ML Workflow



- *Deployment and Maintenance* - where the outcomes of the work are applied to the field/domain to make productive effects

Camp Outline

- Camp objectives
  - To learn how to handle real world data and apply ML techniques on them
  - To understand some machine learning ideas and methodologies
    - *k*-Nearest Neighbors
    - Decision Trees
    - Logistic Regression
  - To obtain hands-on experiences with ML models and algorithms

- Target audiences
  - 2nd year CS/CE students who have just taken Data Structures
    - Not assuming a thick concept of math/stat knowledge
  - Requirements
    - General knowledge on data structures (with C or C++)
    - Strong motivation

Camp Outline

	Day 1	Day 2	Day 3
13:00	Introduction Getting Started with Python	Decision Trees	Logistic Regression
14:00	Hands-on Session	Hands-on Session	Hands-on Session
15:00	*k*-Nearest Neighbors	Visualization using Matplotlib	Data Analytics Practicum
16:00	Hands-on Session	Hands-on Session	

Agenda

- Introduction
- **Getting started with Python (on Google Drive)**
- Classification with *k*-Nearest Neighbors

Python

- An interpreted, high-level, general-purpose programming language
  - Supports multiple programming paradigms, including procedural, object-oriented, and functional programming
  - Dynamically typed and garbage-collected
  - Emphasizes code readability (with its notable use of significant whitespace)
  - Written in C

Why Python?

- Clear syntax
  - "*Executable pseudo-code*"
- Easiness in data manipulation
  - Packages like NumPy, SciPy, Matplotlib, Pandas, …
- Popularity — a large number of users and user groups
  - There is ample development and documentation

- Drawbacks
  - Not as fast as C or Java

Python History

- Late 80s - Conceptualization and initial implementation
  - Led by Guido van Rossum (National Research Institute of Mathematics and Computer Science)

- 1991 - Python (version 0.9.0) was first released
  - Included classes, lists, strings, and exception handling
  - Included supports for functional programming (lambda, map, filter, and reduce)

- 2000 - Python 2.0 was released
  - Included list comprehensions and a fully-functional garbage collector
  - Started to support Unicode

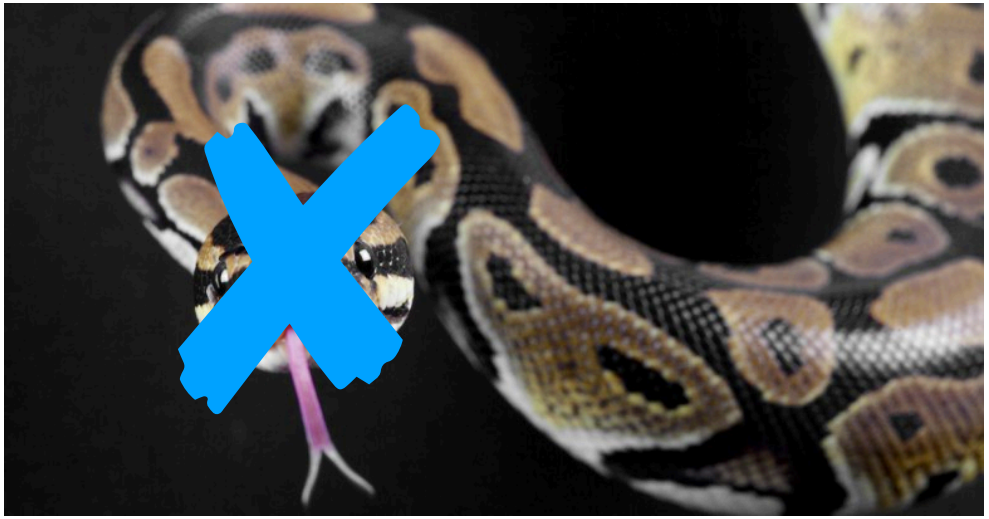
Python History

- 2008 - Python 3.0 was released
  - Improved the internal mechanisms, grammar and expressions
    - Useful link: *Key differences between Python 2 and 3: How to navigate change* https://jaxenter.com/differences-python-2-3-148432.html
  - Broke backward compatibility; *c.f.*, a tool called "`2to3`"

- Current versions
  - Python 3.7 (June 2018)
  - Python 2.7 (July 2010)

Python History

- Fun fact
  - Python is not named after the snake; It is named after the British TV show *Monty Python's Flying Circus*



  - Old and current logos (current logo has two snakes on it)

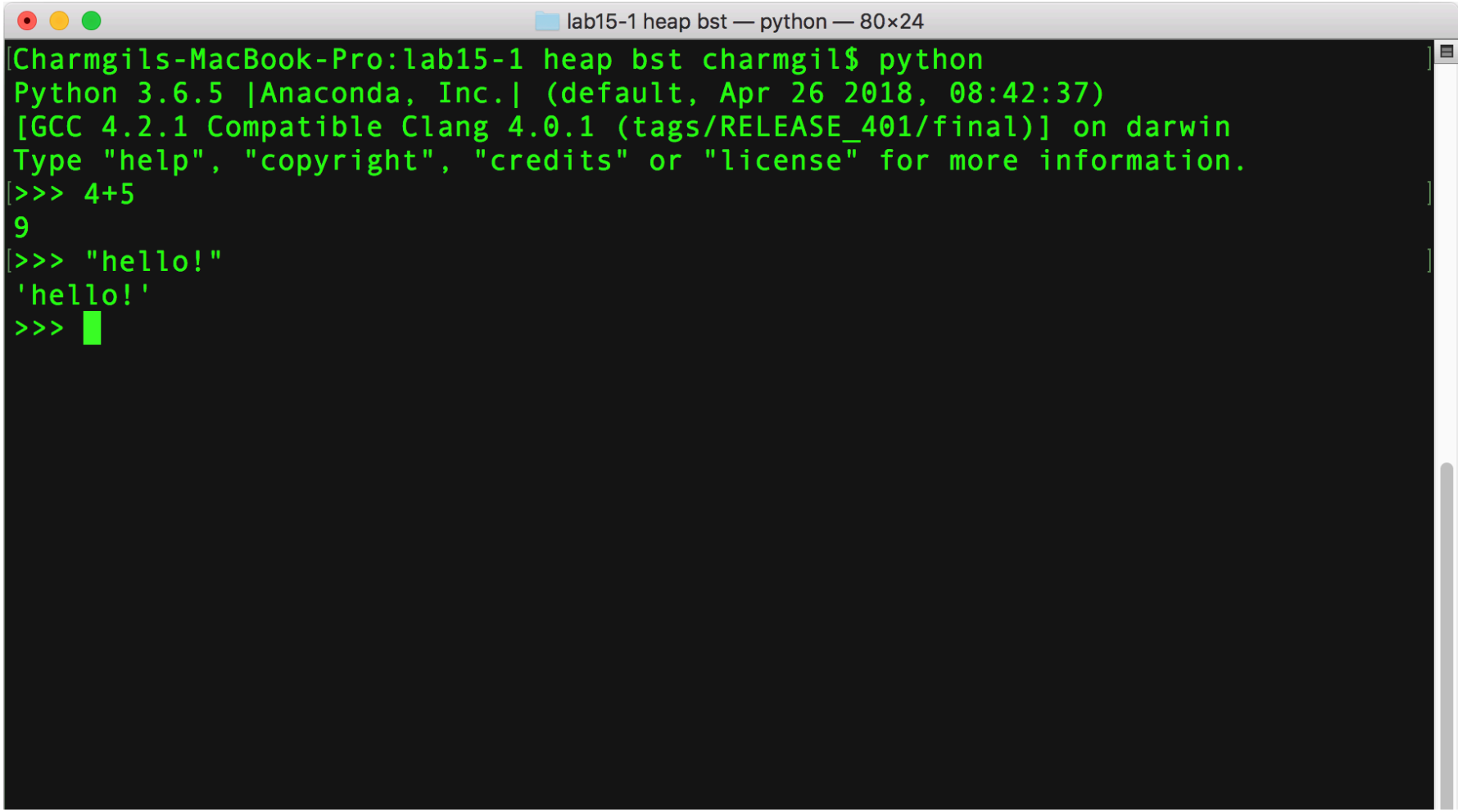
Image source: https://hackernoon.com/10-common-security-gotchas-in-python-and-how-to-avoid-them-e19fbe265e03
https://www.irishtimes.com/culture/tv-radio-web/monty-python-reveal-events-to-mark-50th-anniversary-1.3938624

Python Philosophy

- Python strives for a simpler, less-cluttered syntax and grammar
- Python embraces a "*there should be one — and preferably only one — obvious way to do it*" design philosophy
  - Beautiful is better than ugly
  - Explicit is better than implicit
  - Simple is better than complex
  - Complex is better than complicated
  - Readability counts
- Key features
  - Open source language
  - Extensive library support
  - Cross-platform

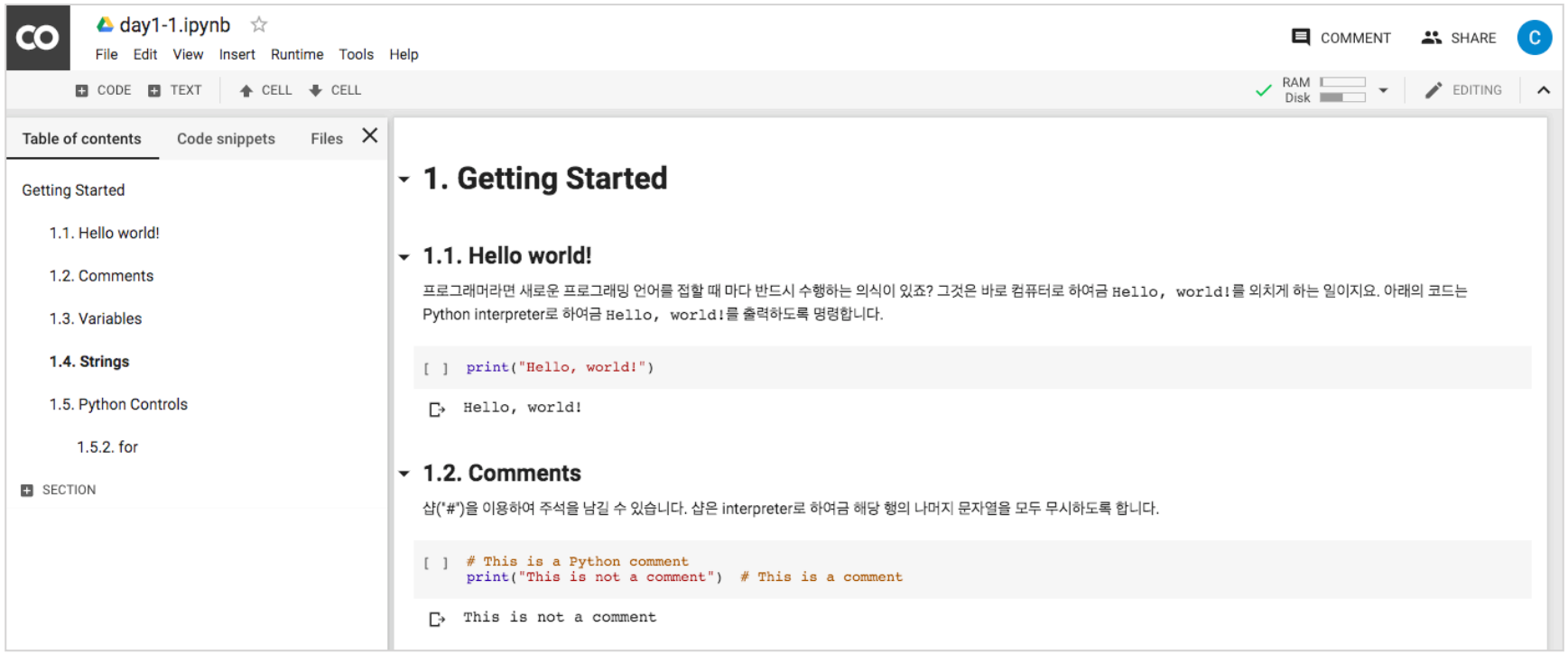
Python ABC

- How to use Python?
  - Local terminal



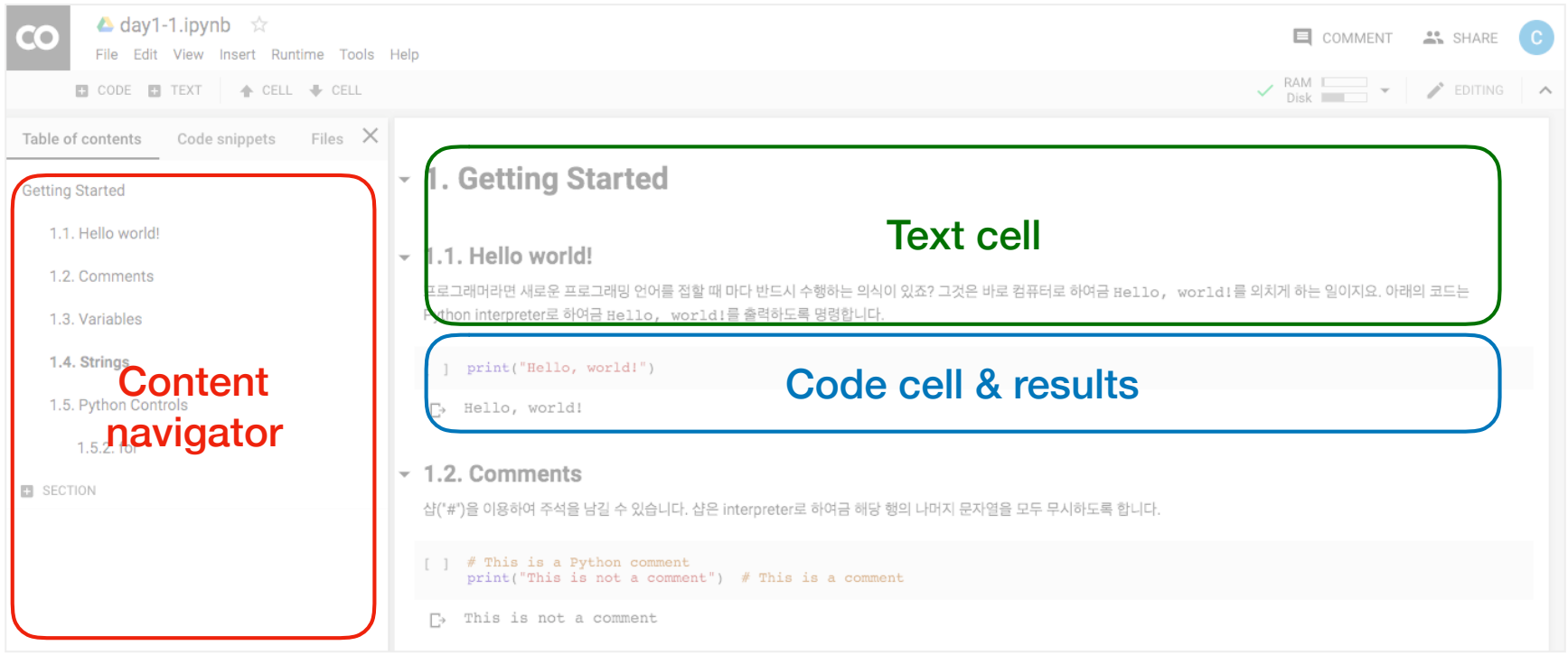
Python ABC

- How to use Python?
  - Google Colaboratory (Jupyter Notebook)



Python ABC

- How to use Python?
  - Google Colaboratory (Jupyter Notebook)



Python ABC

- Hello, world

```
1 print("Hello, world!")
```

```
Hello, world!
```

Python ABC

- Variables
  - In Python everything is an object
  - Every object has a type: scalar type or non-scalar type
    - Scalar: atomic (non-separable) data types; `int`, `float`, `bool`, `None`
    - Non-scalar: data types with internal structures; `string`, `tuple`, `list`, `dictionary`

Python ABC

- Controls
  - `if-elif-else`
  - `for`
  - `while`

Agenda

- Introduction
- Getting started with Python (on Google Drive)
- **Classification with k-Nearest Neighbors**

Machine Learning Taxonomy

- Unsupervised learning
- Supervised learning
- Semi-supervised learning (reinforcement learning)

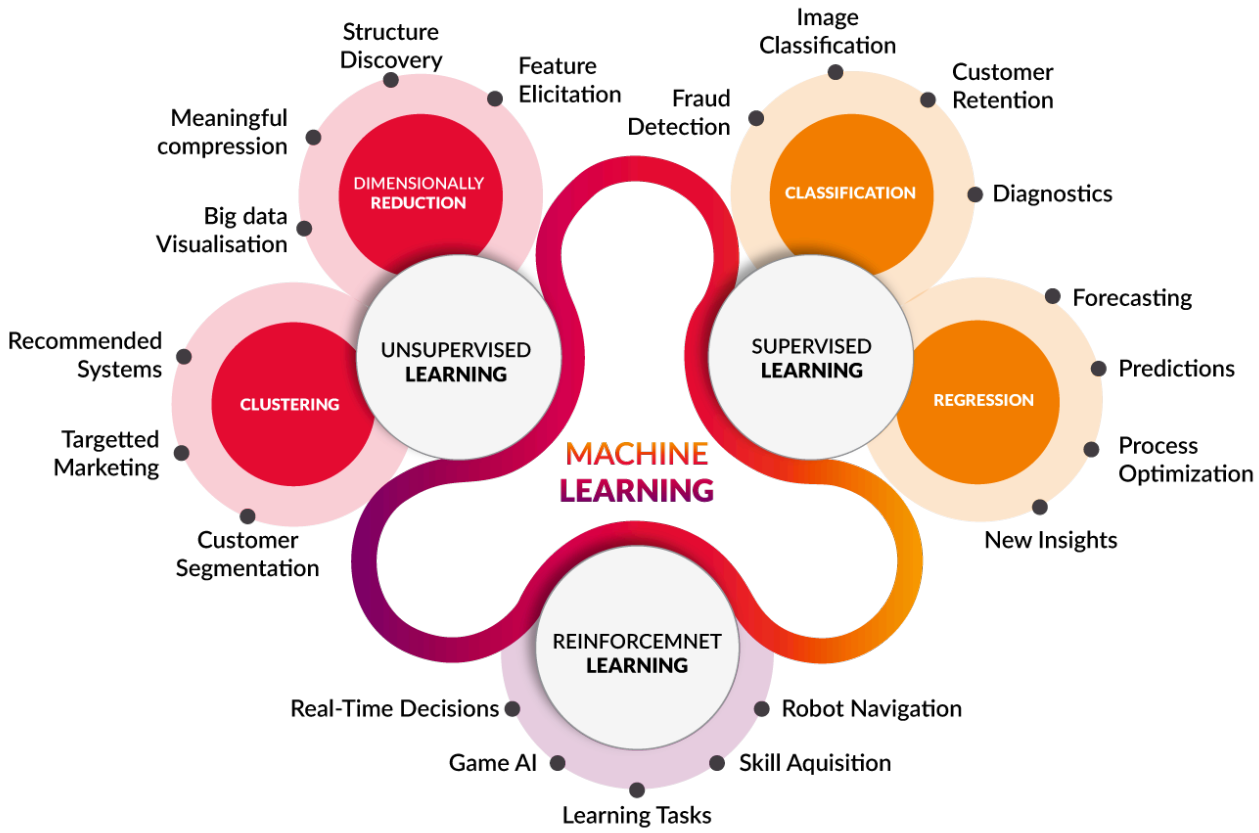


Image source: http://www.cognub.com/index.php/cognitive-platform/

Machine Learning Taxonomy

- Unsupervised learning
  - Draw inferences from data without labels (*there are no correct answers*)
  - To discover unknown data patterns and internal structure
  - *E.g.*, Clustering, outlier detection, autoencoders, self-organizing map, …

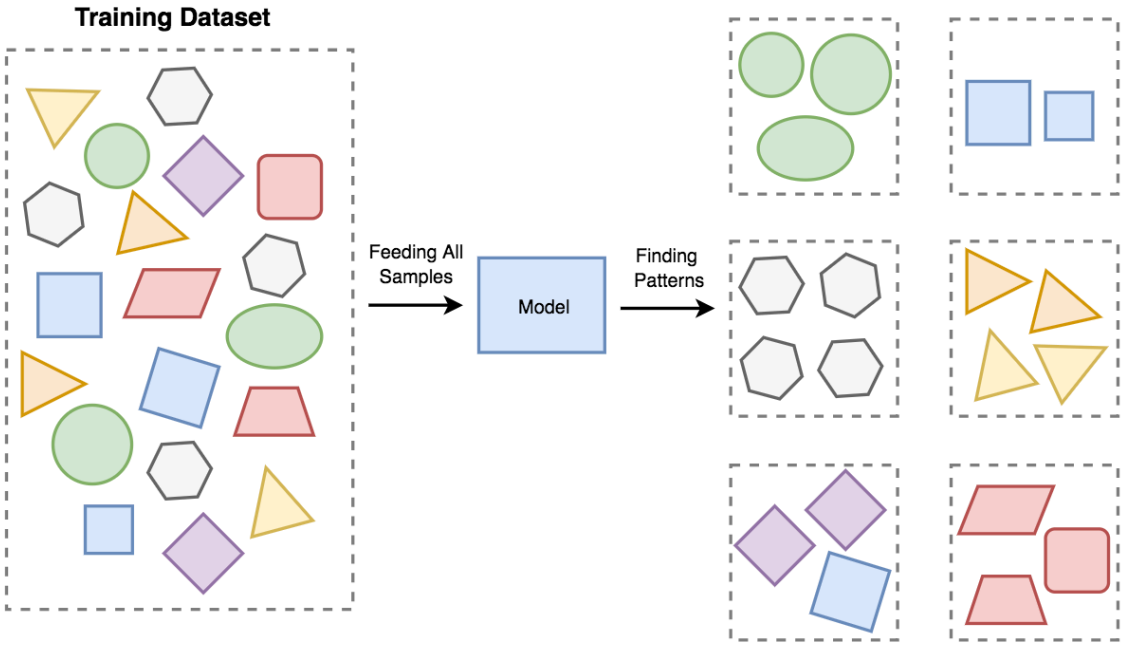


Image source: https://towardsdatascience.com/coding-deep-learning-for-beginners-types-of-machine-learning-b9e651e1ed9d

Machine Learning Taxonomy

- Unsupervised learning Example
  - Dividing data from various countries around the world into three clusters representing Developed, Developing and Underdeveloped nations

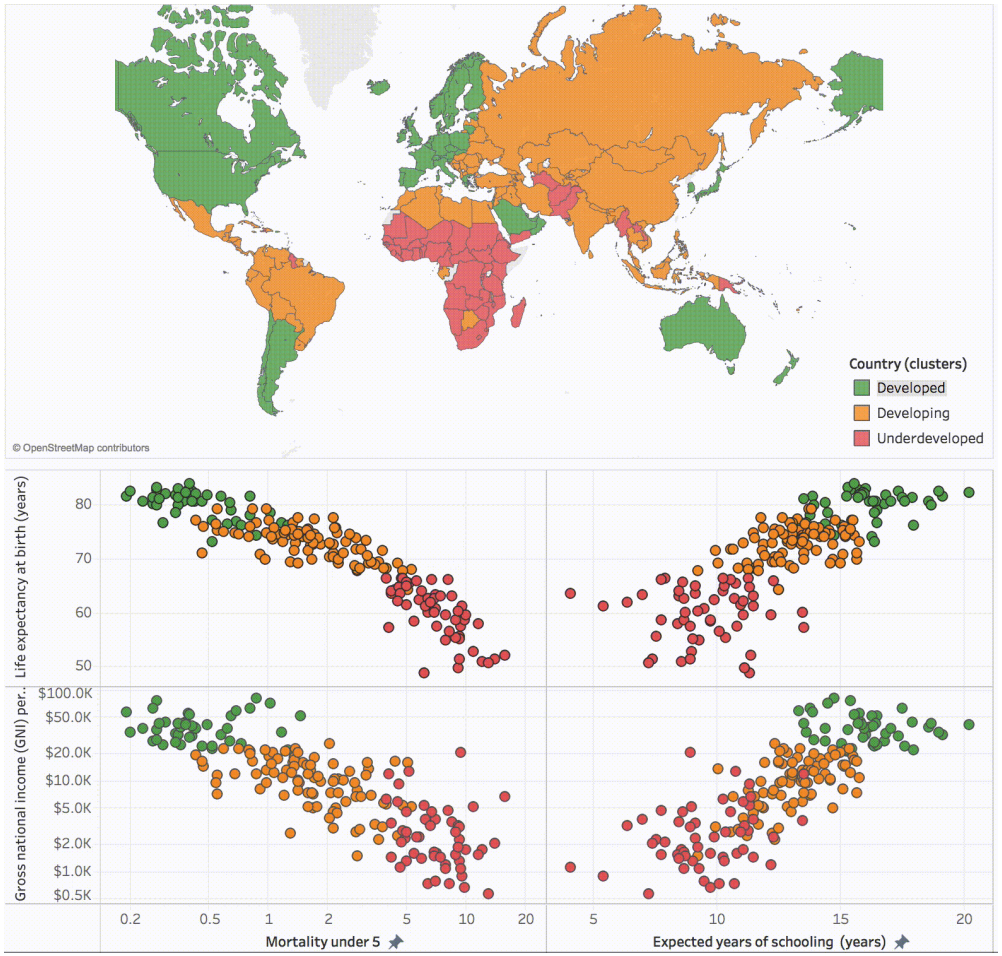


Image source: https://www.tableau.com/about/blog/2016/7/uncover-patterns-your-data-tableau-10s-clustering-feature-56373

Machine Learning Taxonomy

- Supervised learning
  - Build predictive models based on data with labels (input-output pairs)
  - *E.g.*, classification, regression
    - Classification - the outcome is discrete (binary or categorical)
      - *E.g.*, Is this person healthy or with a disease?
    - Regression - the outcome is real or continuous number
      - *E.g.*, Predict one's life expectancy, based on his/her genetic info

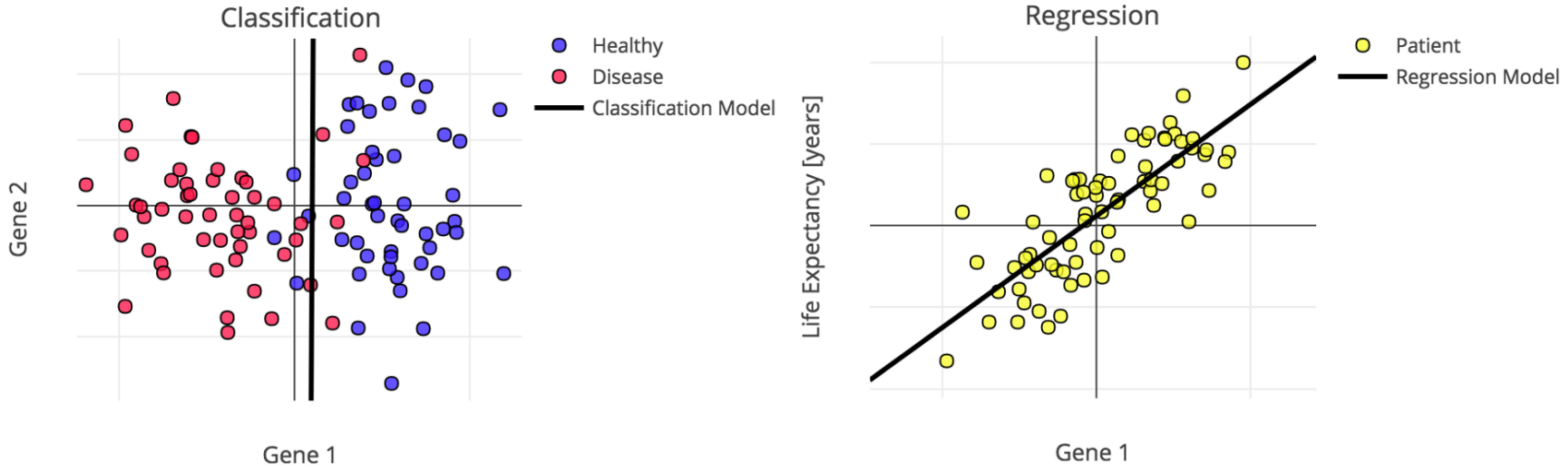
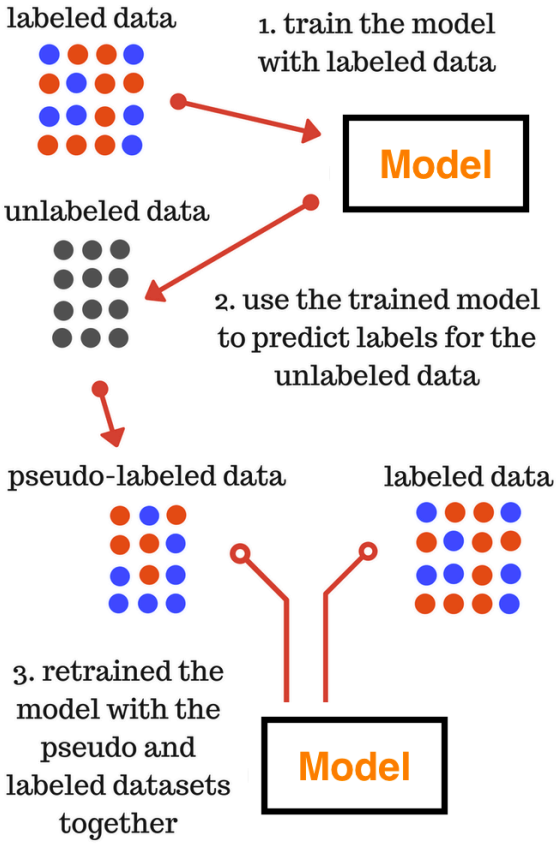
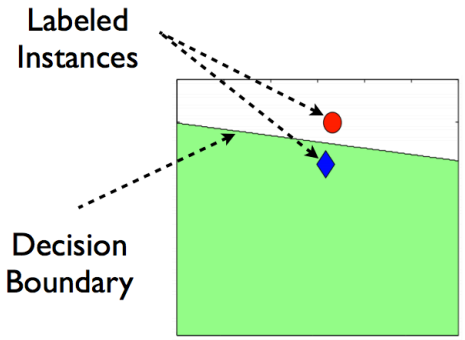


Image source: https://towardsdatascience.com/coding-deep-learning-for-beginners-types-of-machine-learning-b9e651e1ed9d

Machine Learning Taxonomy

- Semisupervised (reinforcement) learning
  - Develop models using partially labeled data (mix of labeled and unlabeled data)





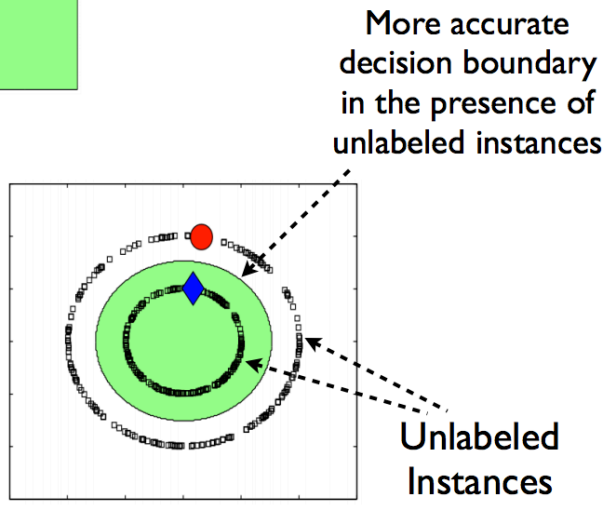


Image source: https://datawhatnow.com/pseudo-labeling-semi-supervised-learning/
https://towardsdatascience.com/coding-deep-learning-for-beginners-types-of-machine-learning-b9e651e1ed9d

Machine Learning Taxonomy

- Semisupervised (reinforcement) learning
  - Develop models using partially labeled data (mix of labeled and unlabeled data)

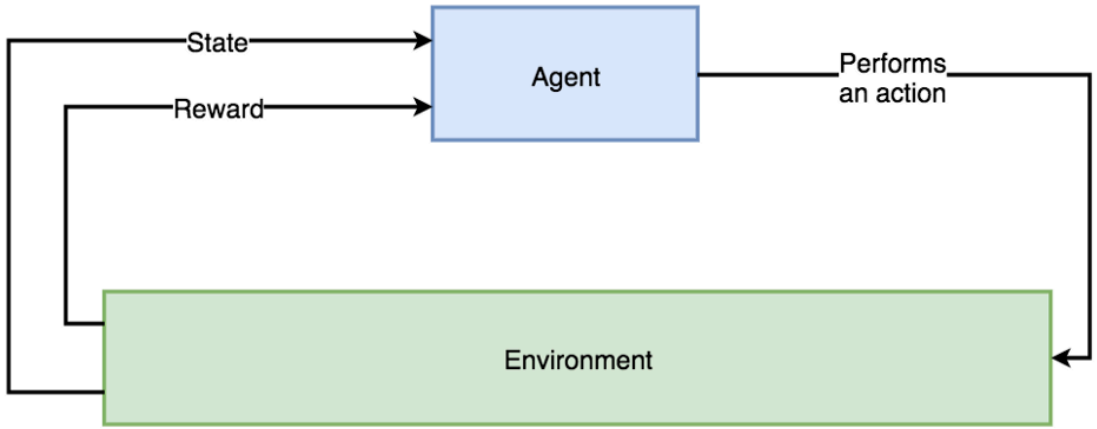
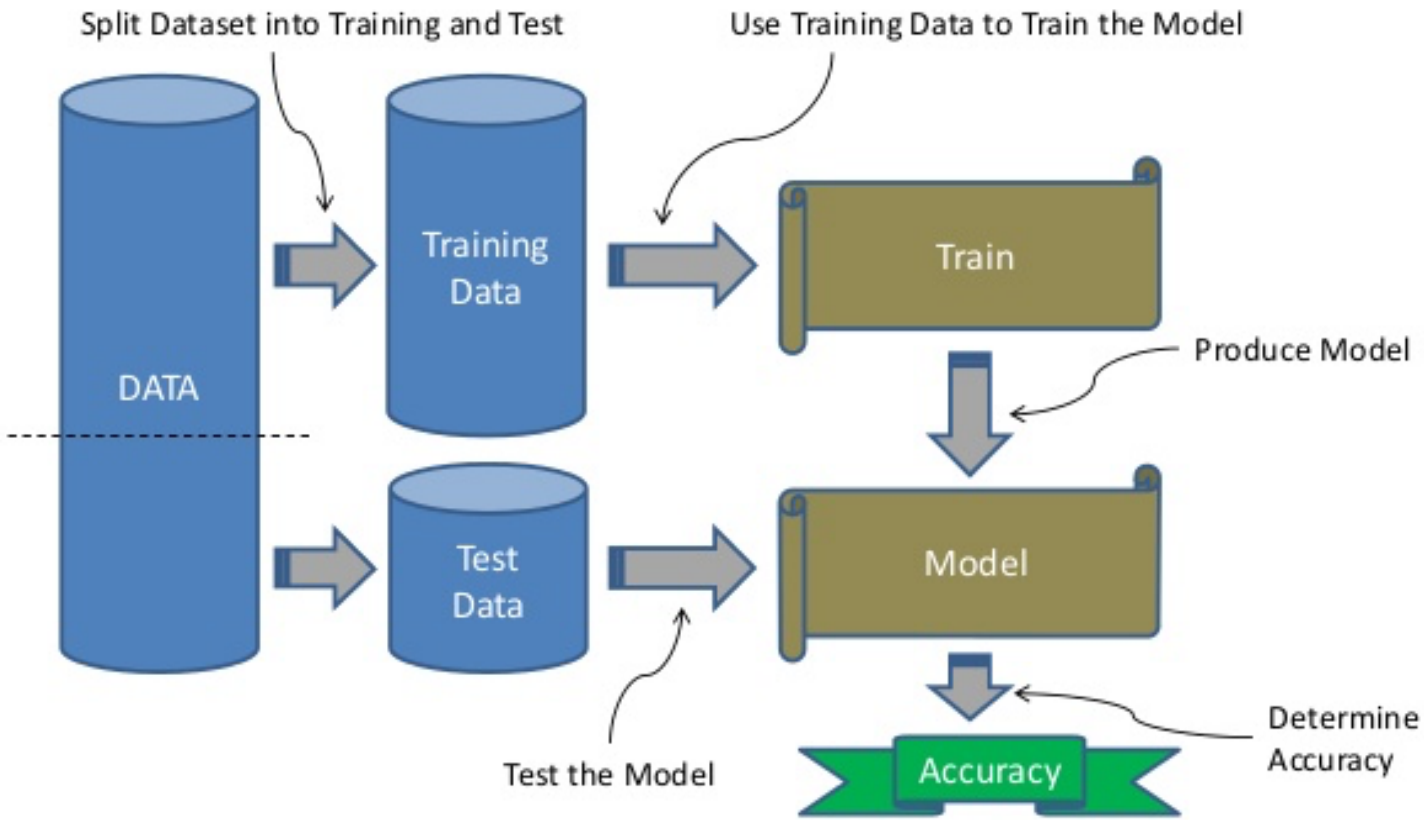


Image source: https://towardsdatascience.com/coding-deep-learning-for-beginners-types-of-machine-learning-b9e651e1ed9d

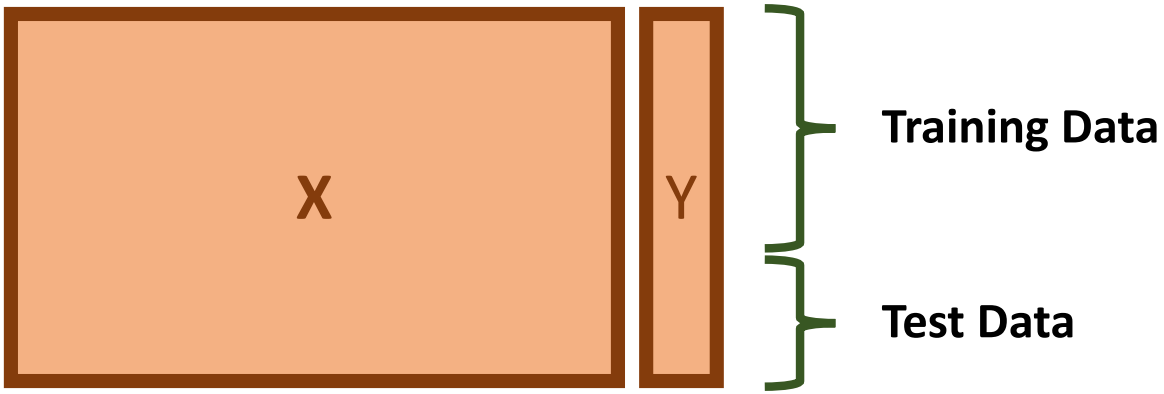
Training and Test Data

- Training data: a set of observations used to generate data models
- Test data: a set of observations used to evaluate trained data models



Data for Supervised Learning

- Data comes as a matrix
  - A row corresponds to an observation or a data instance
  - $\mathbf{X} = (X_1, \ldots, X_d)$: input attributes or features
  - Y: output, class, or label



k-Nearest Neighbors (kNN)

- *Birds of a feather flock together (**類類相從**)*
  - Things that are similar tend to associate with each other
    - Insight: Tundra swans (Cygnus columbianus) and Northern Pintail ducks (Anas acuta)

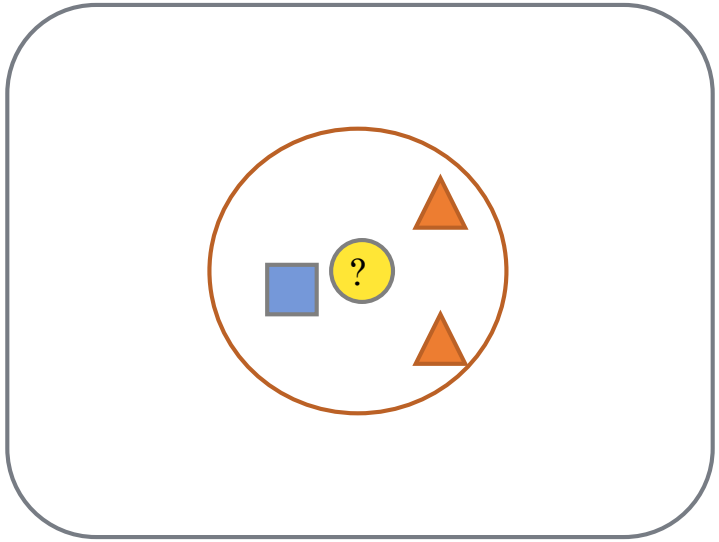
Image source: https://www.canadiannaturephotographer.com/david_lilly7.html

k-Nearest Neighbors (kNN)

- A Simplest form of making predictions
  - *Here we are focusing on classification*; but the same idea could be applied to regression problems
- Instance-based approach
  - To make a decision, retrieve *k* closest training examples in the feature space
  - *A.k.a*, lazy learning approach; all computations are deferred until the prediction time

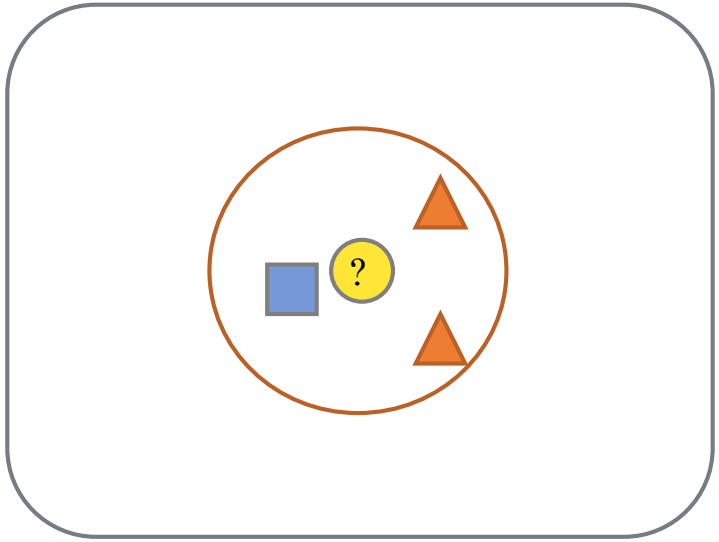
k-Nearest Neighbors (kNN)

- Illustration (*k*=3)



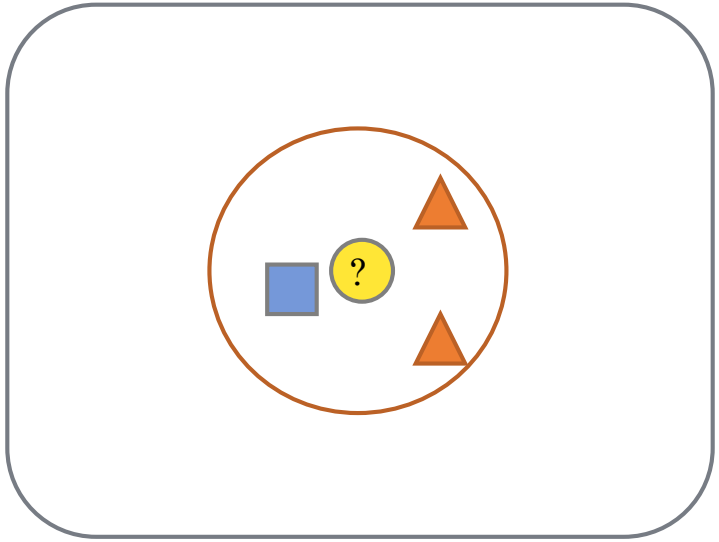
k-Nearest Neighbors (kNN)

- Questions
  - How to define closeness?
  - How to decide *k*?
  - How to make predictions?
  - Are all instances considered equally important?



k-Nearest Neighbors (kNN)

- Requires 4 things:
  - Distance metric is determined according to the data type; *E.g.*, Euclidean, Manhattan, Hamming, …
  - *k* is determined empirically
  - To make a prediction, use a majority vote
  - Instance weighting could be used when making predictions



k-Nearest Neighbors (kNN)

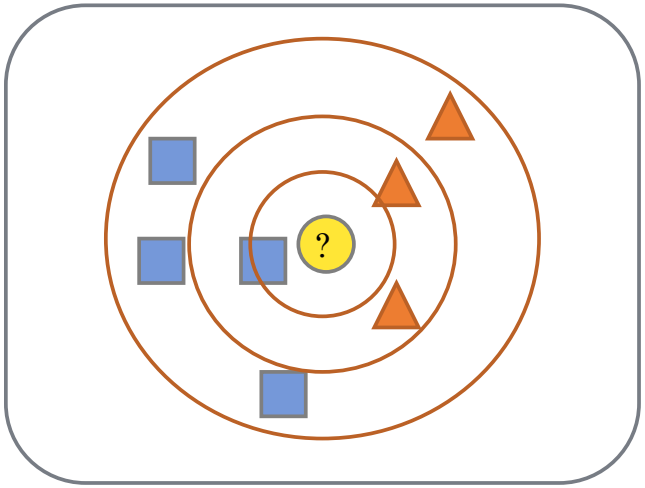
- Requires 4 things:
  - Distance metric is determined according to the data type; *E.g.*, Euclidean, Manhattan, Hamming, …
  - *k* is determined empirically
  - To make a prediction, use a majority vote
  - Instance weighting could be used when making predictions

- To classify a test (unseen) instance:
  - Compute the distances to other training data instances
  - Identify *k* nearest neighbors (*k* closest data points to the test instance)
  - Determine the class label of the testing instance using the class labels of nearest neighbors (majority voting)

k-Nearest Neighbors (kNN)

- If *k* is too small: sensitive to noisy data instances
  too large: neighborhood may include instances from other classes
- Choose an odd value for *k*, to eliminate the ties

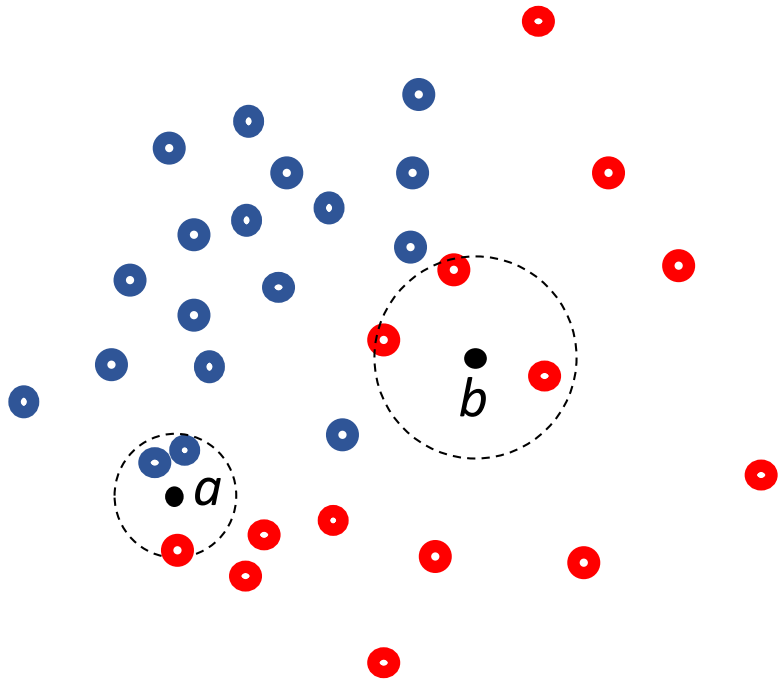
- Example: Determining *k*



- *k = 1: the testing instance belongs to the square class*
- *k = 3: the testing instance belongs to the triangle class*
- *k = 7: the testing instance belongs to the square class*

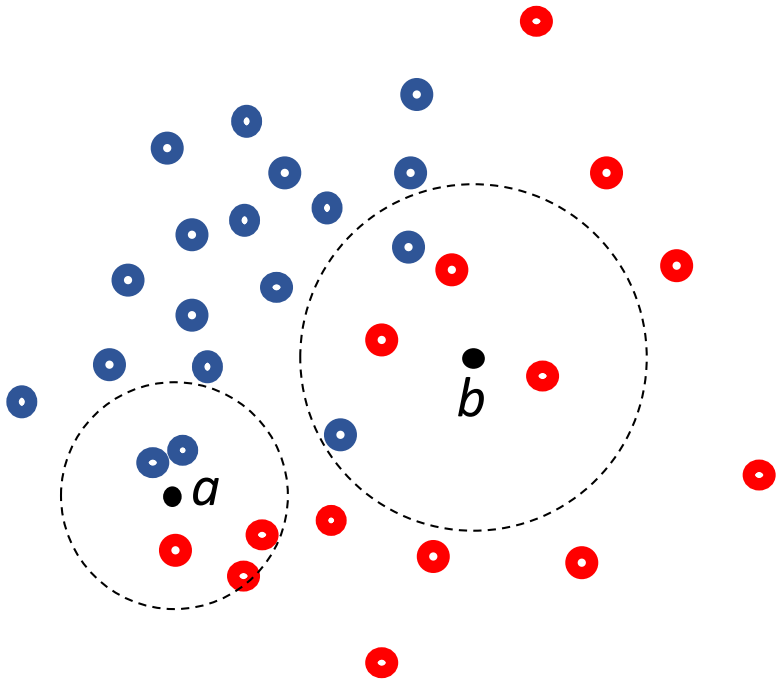
Exercise

- k = 3; Classes = {red, blue}



Exercise

- k = 5; Classes = {red, blue}



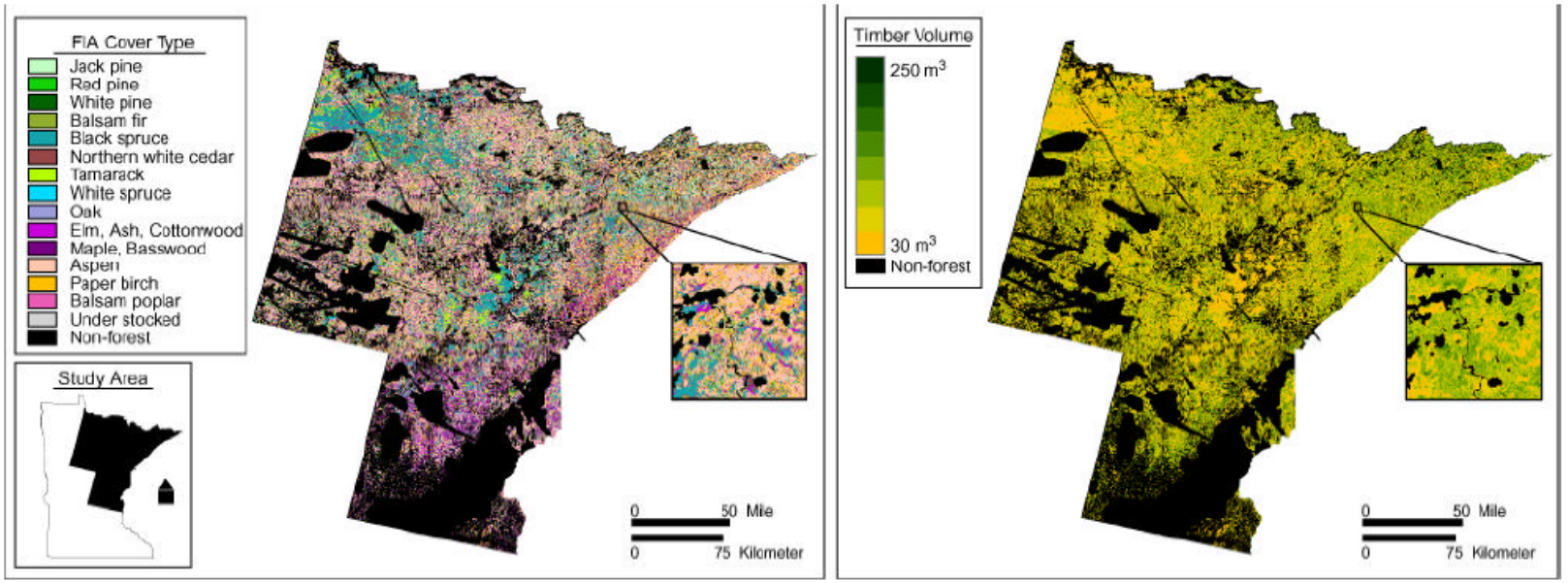
k-Nearest Neighbors (kNN)

- Pros
  - Simple technique that is easily implemented
  - Extremely flexible classification scheme
    - One can easily come up with tweaks: distance metrics, weighting schemes
  - Building a model is inexpensive (at no cost)

- Cons
  - Predictions are relatively expensive
    - Requires distances to all other training data instances
    - More costly as the size of the training set grows
  - Accuracy can be degraded by the presence of noisy of irrelevant features

Examples: kNN

- Forest classification
  - Nationwide forest inventories by USDA Forest Service
  - Maps of tree coverages (https://rs.umn.edu/)
    - Field plot inventories have not been able to produce precise map data
    - Traditional satellite-based forest classifications are not detailed enough
    - Using kNN, precise county and local estimates to obtain more useful operational maps become available

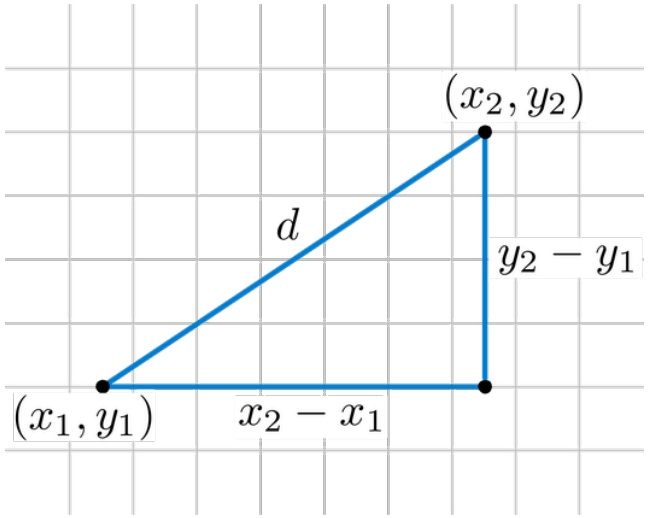


Examples: kNN

- Text categorization
  - Dept. of Computer Science and Engineering, Army HPC Research Center
  - Text categorization is the task of deciding whether a document belongs to a set of pre-specified classes of documents
  - Weight adjusted k-NN is used to improve the classification objective function
    - Each feature has an associated weight
      - A higher weight implies that this feature is more important in the classification task
    - A small subset of the vocabulary may be useful in categorizing documents

Data Normalization

- Consider Euclidean distance



$$E(x, y) = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2}$$

- Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
- Example
  - *Height of a person may vary from 140 cm to 200 cm*
  - *Weight of a person may vary from 40 kg to 140 kg*
  - *Income of a person may vary from KRW 20M to 500M*

Potential Improvement

- Prediction procedure (revisited)
  - To classify a test (unseen) instance:
    - Compute the distances to other training data instances
    - Identify *k* nearest neighbors (*k* closest data points to the test instance)
    - Determine the class label of the testing instance using the class labels of nearest neighbors (majority voting)

Bottleneck

  - Potential solutions
    - Make the distance estimation faster
      - https://mccormickml.com/2014/08/22/fast-euclidean-distance-calculation-with-matlab-code/
    - Take less number of instances into consideration
      - https://www.youtube.com/watch?v=Y4ZgLlDfKDg

References

- Peter Harrington, Machine Learning in Action. Manning Publications Co., 2012.
- John Guttag, Introduction to Computation and Programming Using Python with Application to Understanding Data. 2nd Ed. MIT Press, 2016.

real - world machine learning. (henrik brink)

machine learning (Tom Mitchell)

Thank you!

Machine Learning with Real World Data (Day 1)

홍참길, 한동대학교 전산전자공학부

July 16, 2019 @ 전산전자공학부 여름 학기 전공 캠프

charmgil@handong.edu

한동대학교 HANDONG GLOBAL UNIVERSITY