Data Science: state of the art

Valentin Kuznetsov, Cornell University

SOSC 2018

Who Am I?

- Theoretical Physicists (neutrino oscillations) at Irkutsk Univ & JINR
- Particle Physicists (tracking, silicon detectors) at CERN
- PhD in Physics (theory + experiment) at JINR
- Computing in HEP at JINR, CERN, Fermilab, Cornell
    - HEP experiments: NOMAD, D0, Cleo-c, CMS
- Data Scientists (Univ. of Washington) at Cornell University
    - data management, data discovery, services
    - BigData, Analytics model, Machine Learning

Introduction

DATA

Data Scientist: The Sexiest Job of the 21st Century

by Thomas H. Davenport and D.J. Patil

FROM THE OCTOBER 2012 ISSUE

SUMMARY | SAVE | SHARE | 5 COMMENT | TEXT SIZE | PRINT | $8.95 BUY COPIES

When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren't seeking out connections with the people who were already on the site at

WHAT TO READ NEXT

Big Data: The Management Revolution

5 Essential Principles for Understanding Analytics

Data Scientists Don't Scale

VIEW MORE FROM THE

October 2012 Issue

Josh Wills
@josh_wills

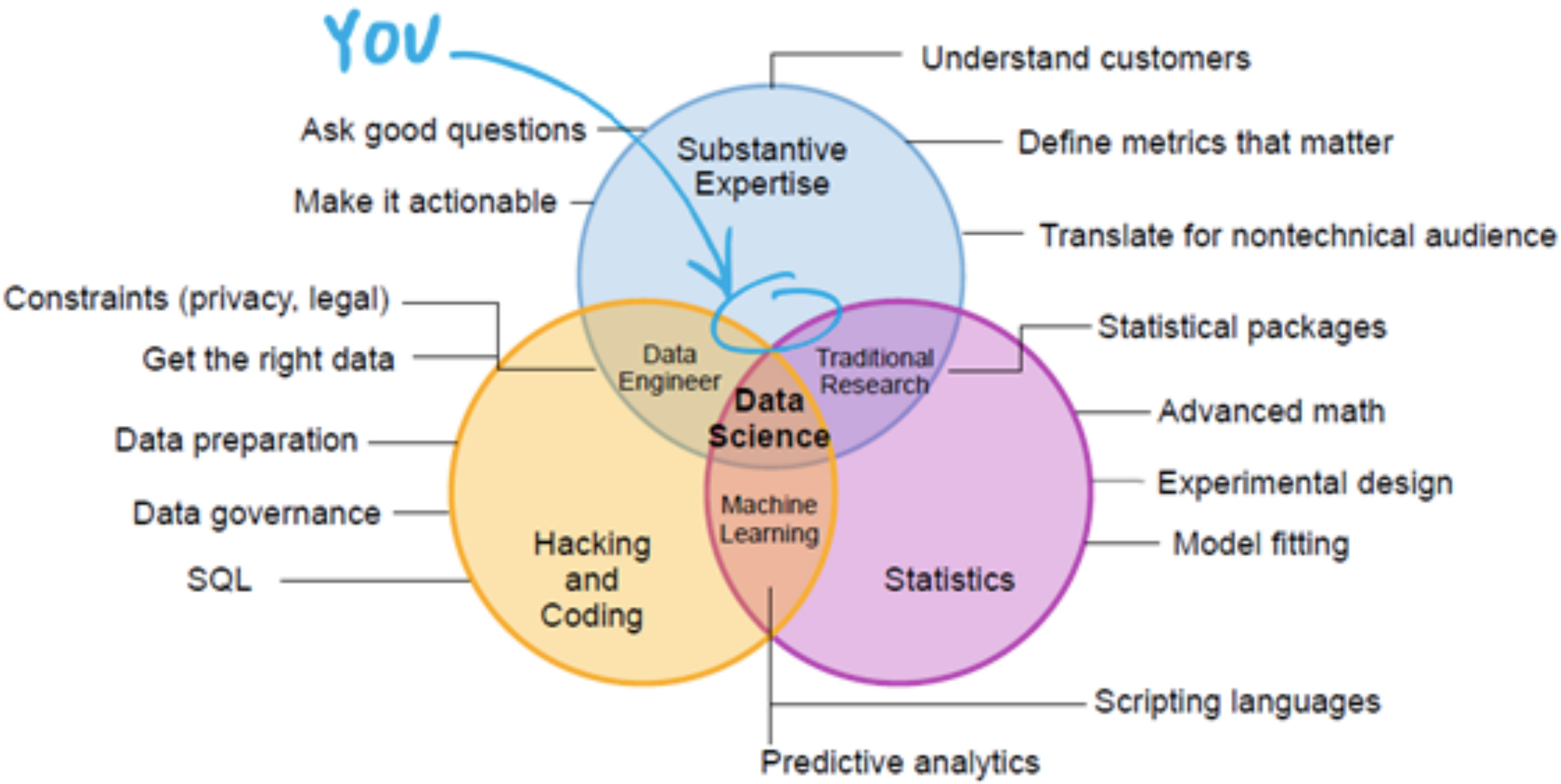
Following

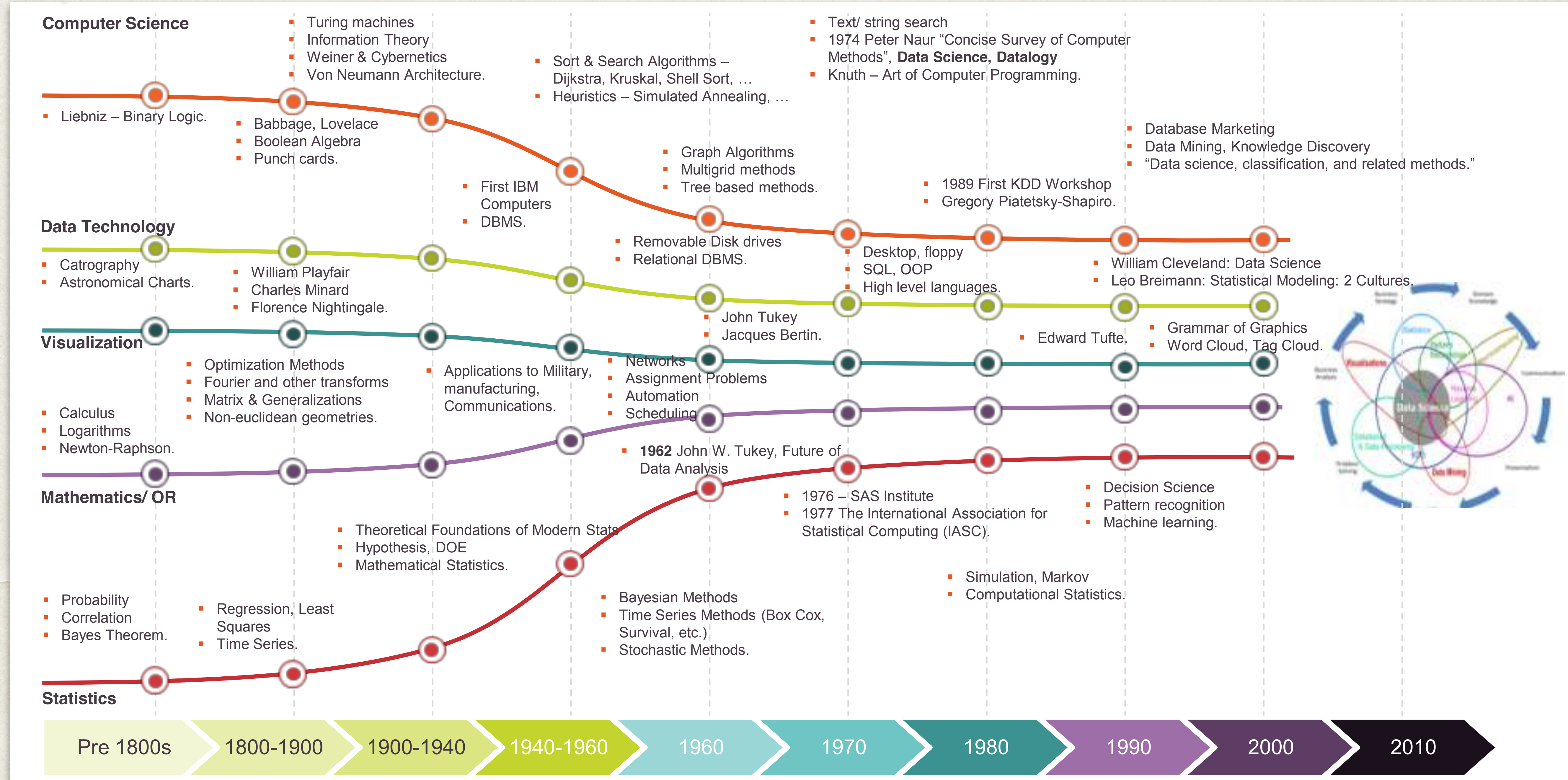
Data Scientist (n.): Person who is better at statistics than any software engineer and better at software engineering than any statistician.

RETWEETS 1,381 LIKES 857

6:55 PM - 3 May 2012

Data Science Is Multidisciplinary




Computer Science
Liebniz – Binary Logic.
Turing machines
Information Theory
Weiner & Cybernetics
Von Neumann Architecture.
Babbage, Lovelace
Boolean Algebra
Punch cards.
Sort & Search Algorithms – Dijkstra, Kruskal, Shell Sort, …
Heuristics – Simulated Annealing, …
First IBM Computers
DBMS.
Graph Algorithms
Multigrid methods
Tree based methods.
Text/ string search
1974 Peter Naur “Concise Survey of Computer Methods”, Data Science, Datalogy
Knuth – Art of Computer Programming.
1989 First KDD Workshop
Gregory Piatetsky-Shapiro.
Database Marketing
Data Mining, Knowledge Discovery
“Data science, classification, and related methods.”
Data Technology
Catrography
Astronomical Charts.
William Playfair
Charles Minard
Florence Nightingale.
Removable Disk drives
Relational DBMS.
Desktop, floppy
SQL, OOP
High level languages.
William Cleveland: Data Science
Leo Breimann: Statistical Modeling: 2 Cultures.
Visualization
John Tukey
Jacques Bertin.
Edward Tufte.
Grammar of Graphics
Word Cloud, Tag Cloud.
Optimization Methods
Fourier and other transforms
Matrix & Generalizations
Non-euclidean geometries.
Applications to Military, manufacturing, Communications.
Networks
Assignment Problems
Automation
Scheduling
Calculus
Logarithms
Newton-Raphson.
Mathematics/ OR
1962 John W. Tukey, Future of Data Analysis
1976 – SAS Institute
1977 The International Association for Statistical Computing (IASC).
Decision Science
Pattern recognition
Machine learning.
Theoretical Foundations of Modern Stats
Hypothesis, DOE
Mathematical Statistics.
Simulation, Markov
Computational Statistics.
Probability
Correlation
Bayes Theorem.
Regression, Least Squares
Time Series.
Bayesian Methods
Time Series Methods (Box Cox, Survival, etc.)
Stochastic Methods.
Statistics
Pre 1800s
1800-1900
1900-1940
1940-1960
1960
1970
1980
1990
2000
2010

BIG DATA & AI LANDSCAPE 2018

V1 – Last updated 6/19/2018

 mattturck.com/bigdata2018

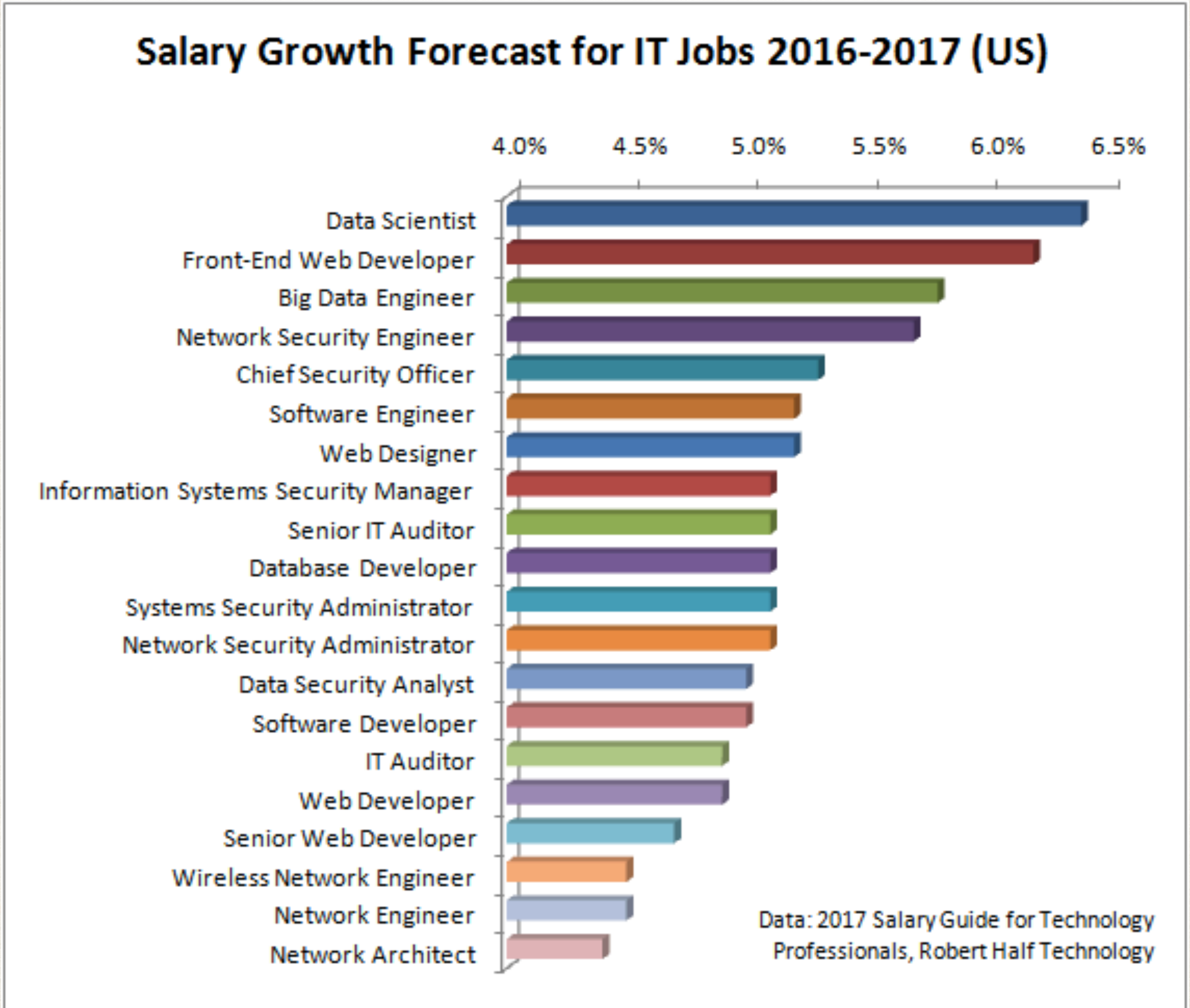
FIRSTMARK
EARLY STAGE VENTURE CAPITAL

You need to know (my bare minimum)

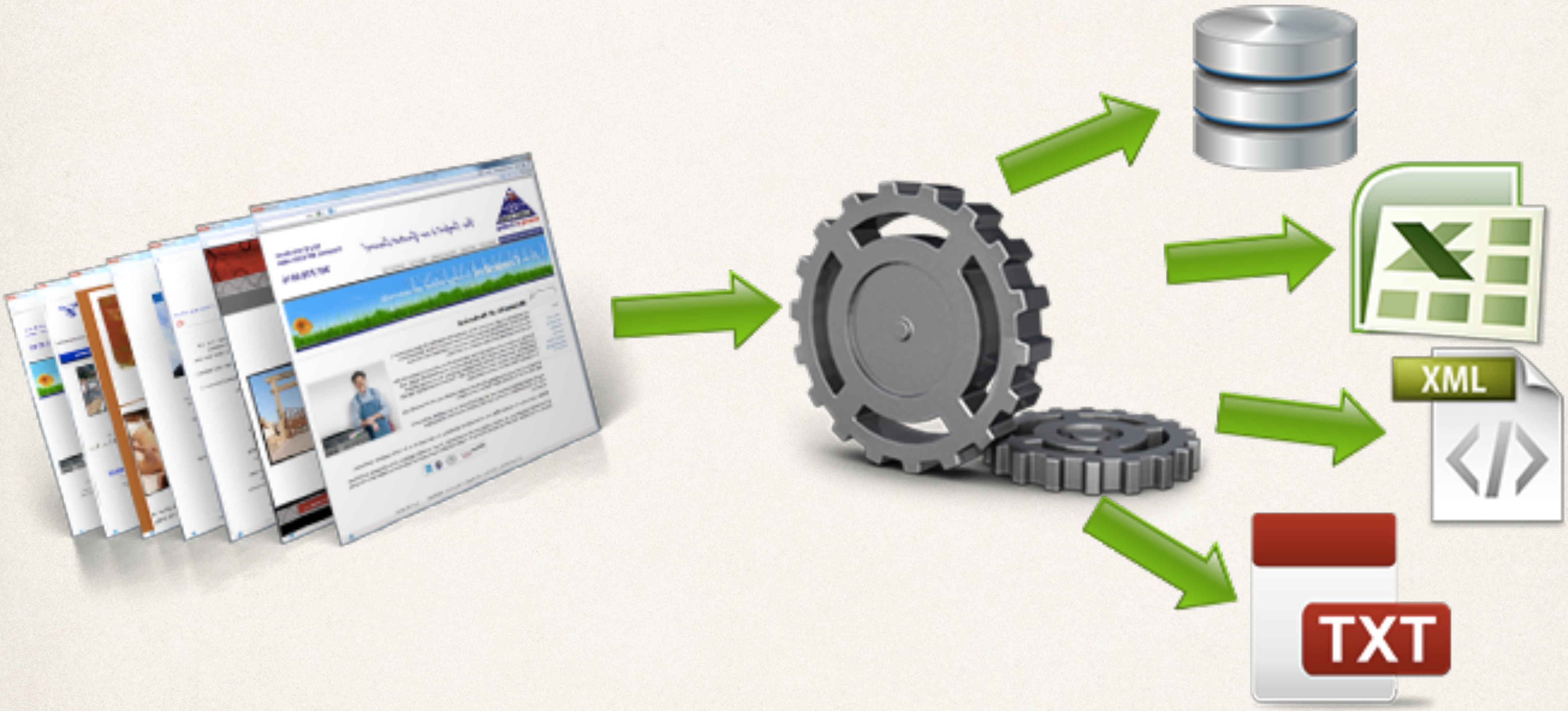
- Math, statistics, algorithms, be able to read scientific paper
- Programming languages: C/C++, Python, R, Go, etc.
- Shell scripting and unix tools: bash, sed, awk, etc.
- How to build/install packages/tools
  - from source code: make, autoconf, environment, tar, etc.
  - from package management tools: rpm, yum, apt, dpkg, pip, anaconda, and/or build your favorite Linux distribution
- Versioning tools: git, gitlab, bitbucket, etc.
- Compilers, linkers, structure of libraries, object files, etc.
- Statistical and visualization tools: R, MatLab, Pandas, NumPy, SciPy, matplotlib, etc.
- ML tools: Scikit-Learn, R, TensorFlow, Keras, xgboost, etc.

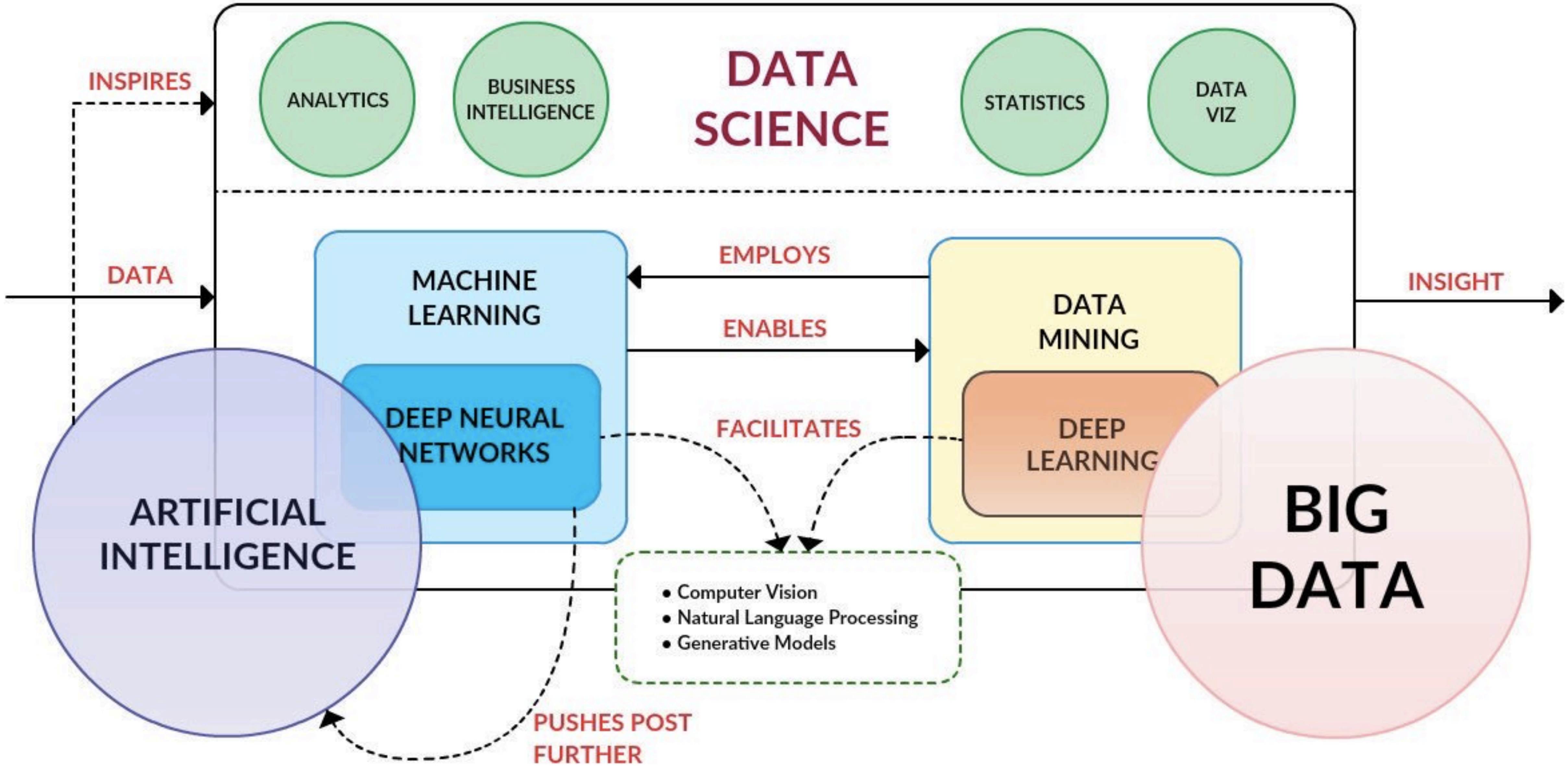
You need to know, cont'd

- Platforms: AWS, Azure, Google Cloud, etc.
- BigData tools: Hadoop, Spark, HDFS, HDF5, etc.
- Databases: ORACLE, MySQL, SQLite, NoSQL, GraphDB, MongoDB, CouchDB, etc.
- Monitoring: ElasticSearch, Kibana, Grafana, Prometheus, etc.
- Streaming: Spark, Kafka, Storm, etc.
- Collaboration: Jupyter, Zeppelin, Anaconda, SWAN, etc.
- Search: ElasticSearch, Lucene, Solr, etc.
- Lexical analysis & NLP: lexer, tokenizer, scanner, etc.
- Read, write, and ask questions about everything


Salary Growth Forecast for IT Jobs 2016-2017 (US)
4.0%
4.5%
5.0%
5.5%
6.0%
6.5%
Data Scientist
Front-End Web Developer
Big Data Engineer
Network Security Engineer
Chief Security Officer
Software Engineer
Web Designer
Information Systems Security Manager
Senior IT Auditor
Database Developer
Systems Security Administrator
Network Security Administrator
Data Security Analyst
Software Developer
IT Auditor
Web Developer
Senior Web Developer
Wireless Network Engineer
Network Engineer
Network Architect
Data: 2017 Salary Guide for Technology Professionals, Robert Half Technology

Problem statement


XML
TXT


INSPIRES
DATA
ANALYTICS
BUSINESS INTELLIGENCE
DATA SCIENCE
STATISTICS
DATA VIZ
MACHINE LEARNING
DEEP NEURAL NETWORKS
EMPLOYS
ENABLES
FACILITATES
DATA MINING
DEEP LEARNING
INSIGHT
ARTIFICIAL INTELLIGENCE
BIG DATA
• Computer Vision
• Natural Language Processing
• Generative Models
PUSHES POST FURTHER


Category
Customer
Spill
Injury
Transport
Equipment
Security
Divergence
Complaint
Cause
Material
Procedure
Design
Training
Management
External
Equipment
Personnel
Month
Year / Month
2007
2008
2009
Count
Site
Weston
Bolton
Shirley
Lincoln
Maynard
Acton
Concord
Hudson
Trend
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Severity
Near Miss
Medium
Major
Critical
Cause
Design
Equipment
External
Managem..
Material
Personnel
Procedure
Training
Severity
Critical
Major
Medium
Near Miss
Count
517.0
Category
(All)
Complaint
Customer
Divergence
Equipment
Injury
Security
Spill
Transport
Site
(All)
Acton
Bolton
Concord
Hudson
Lincoln
Maynard
Shirley
Weston
Month
(All)
Year
(All)
2007
2008
2009
Status
(All)
Closed
Open

Data, Algorithms, Techniques

Engineering Effort for Effective ML

- From “Hidden Technical Debt in Machine Learning Systems”, D. Sculley at al. (Google), paper at NIPS 2015

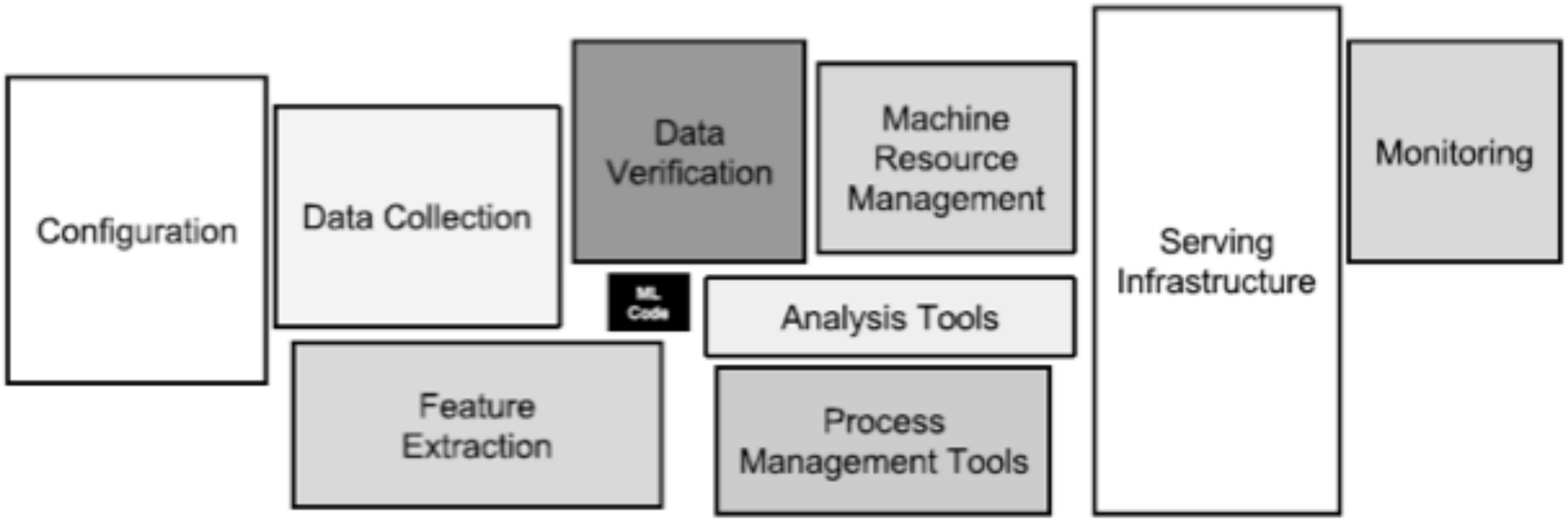


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Data pre-processing

- Most of the time will be spend in this step
- Data clean-up, data transformation, feature engineering
  - data transformation
    - scaling and normalization
    - encoding, aggregation features, log-transformation (to remove outliers)
  - data visualization, exploration
  - data augmentation, imputing, bucketing, binning, feature interactions
  - dimensionality reduction
- **Your programming skills will be required here: R, Python, Databases, etc.**

Types of data

- Data
  - Categorical
  - Numerical
    - Discrete
    - Continuous

Categorical

Examples:
- Marital status
- Eye color
- Political party

Discrete

Examples:
- # children
- month of the year

Continuous

Examples:
- weight
- house prices

Data transformation

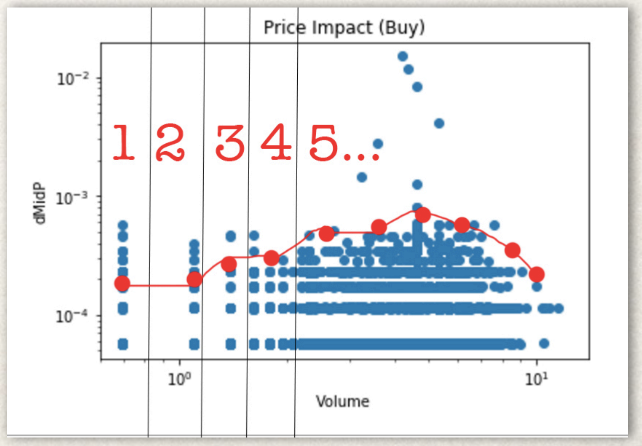
- Data transformation and aggregation: log, sum of values
- Scaling: a technique to scale data to a given range [0,1] or any other range

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- Normalization/Standardization: a technique to scale data to mean with zero and and unit-variance

$$x' = \frac{x - \bar{x}}{\sigma}$$

- Augmentation: a technique to create additional data based on input sample which slightly differ from it, e.g. image rotation, flip, scale, crop, etc.
- Bucketing/Binning: a technique to place similar values into buckets/bins

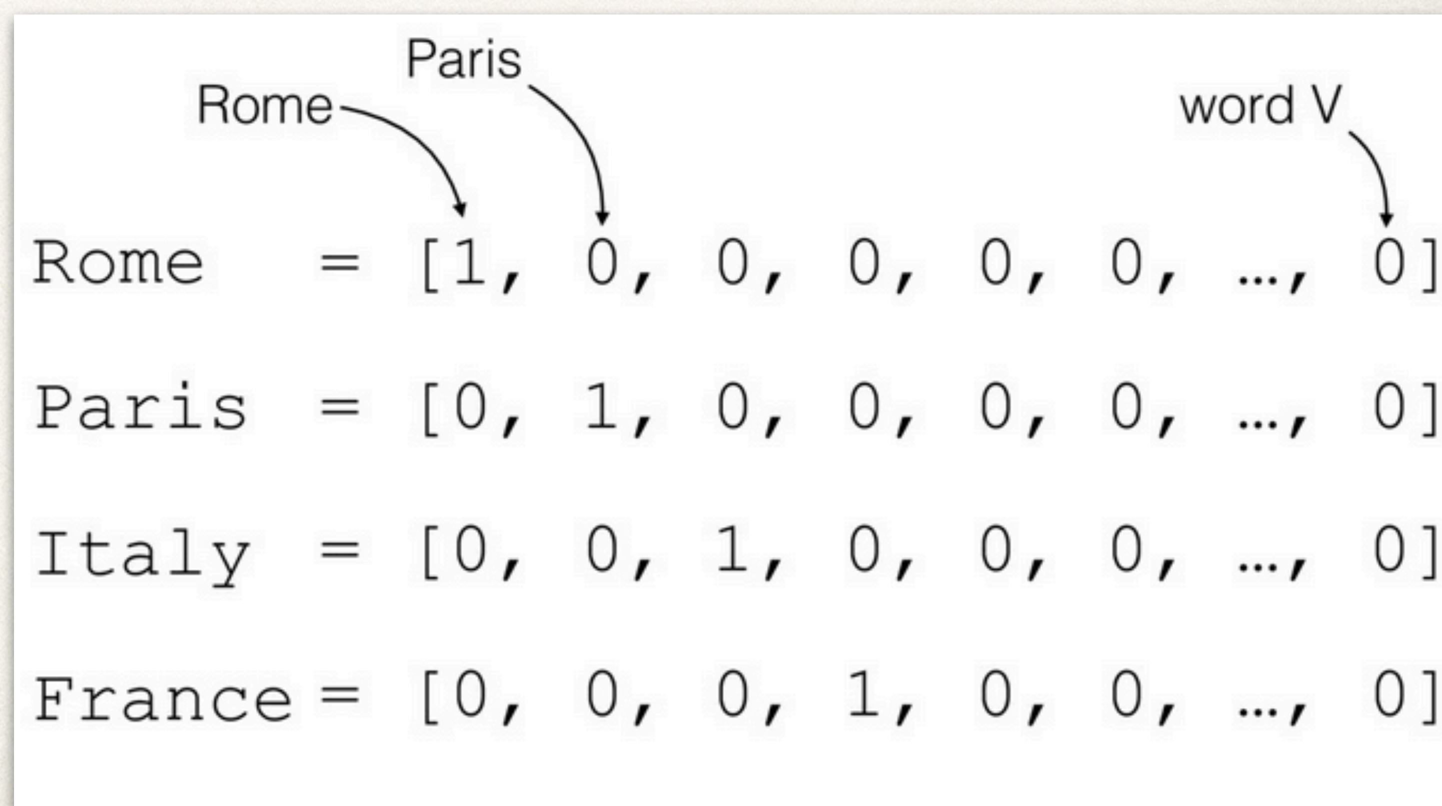


One-hot-encoding

[Ref]

- It is a technique to handle “categorical” data
- It represents categorical column as vector of words
- You need to define word vector for full set of data (train + test datasets)
- Issues with NULL or missing data
  - delete rows with missing data
  - impute data for missing values

“One-Hot” refers to a state in electrical engineering where all of the bits in a circuit are 0, except a single bit with a value of 1 (it is said to be “hot”).



Leave-one-out encoding

Ref

- Use mean of all values within the same category except given row
- Add random noise
- Replace categorical value with leave-one-out times noise
- The test categorical values always represented as mean and no noise
- This technique may complement one-hot encoding

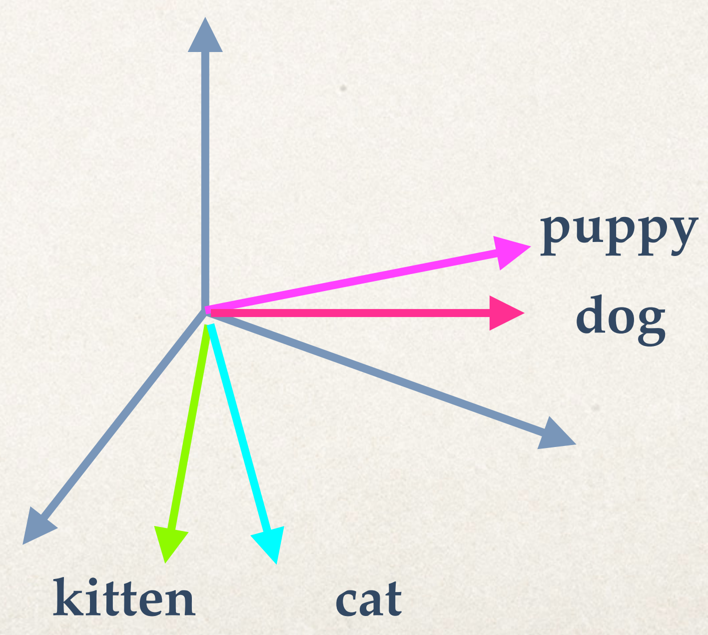
Split	UserID	Y	mean_y	random	newID
Train	A1	0	0.667	1.05	0.70035
Train	A1	1	0.333	0.97	0.32301
Train	A1	1	0.333	0.98	0.32634
Train	A1	0	0.667	1.02	0.68034
Test	A1	-	0.5	1	0.5
Test	A1	-	0.5	1	0.5
Train	A2	0			

Word embedding

Ref

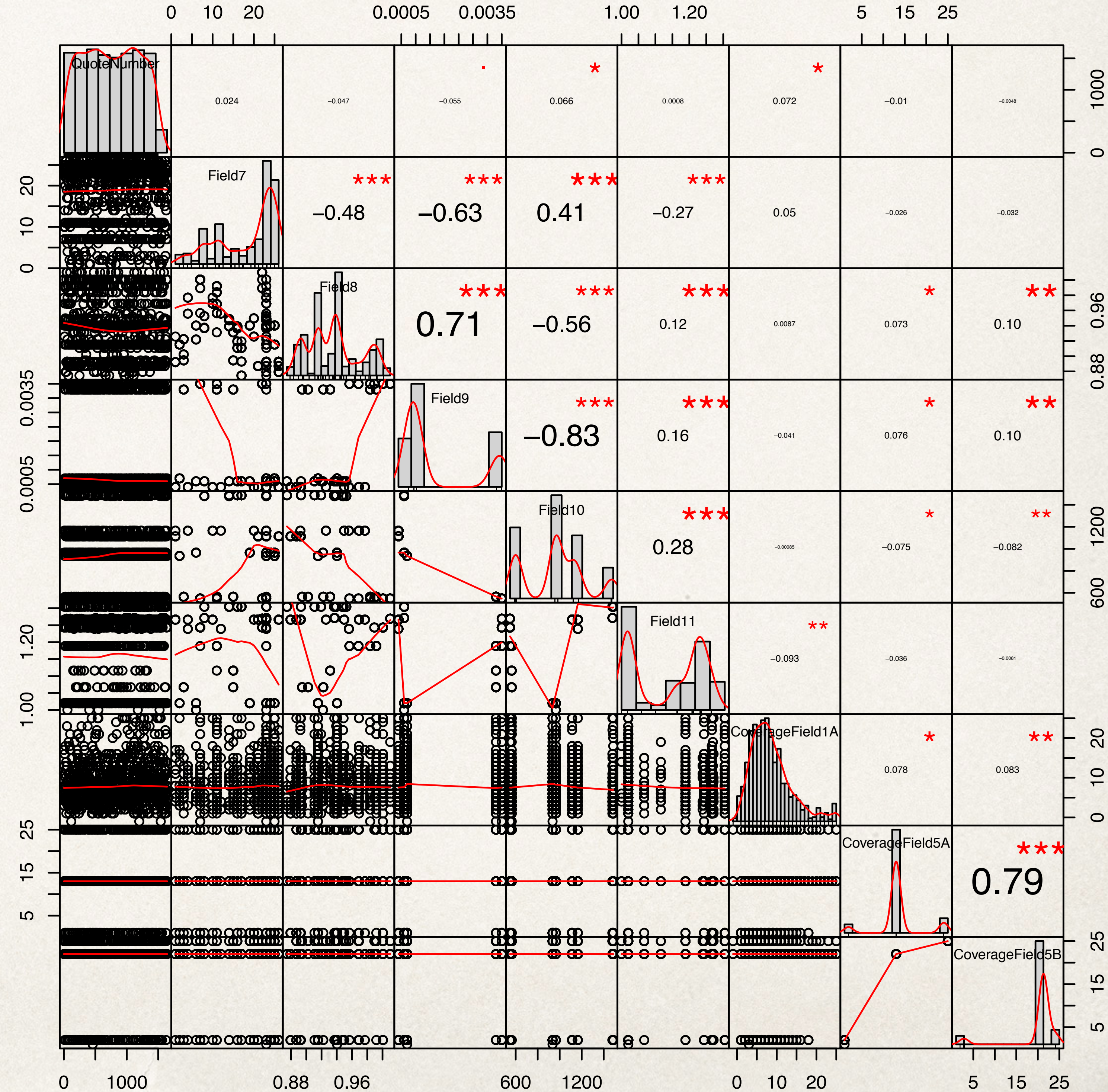
- A way to capture multi-dimensional relationships between categories
  - e.g. Sun and Sat may have similar effect while other days may be treated independently
  - you define a dimension of word vector up-front
  - it projects categorical variables into another phase space, e.g. days may be sunny or rainy, season or off season; all of these features are hidden from original data representation
- Use NN or other ML algorithms to train the model to find best representation of embedded variables

puppy	[0.9, 1.0, 0.0]
dog	[1.0, 0.2, 0.0]
kitten	[0.0, 1.0, 0.9]
cat	[0.0, 0.2, 1.0]

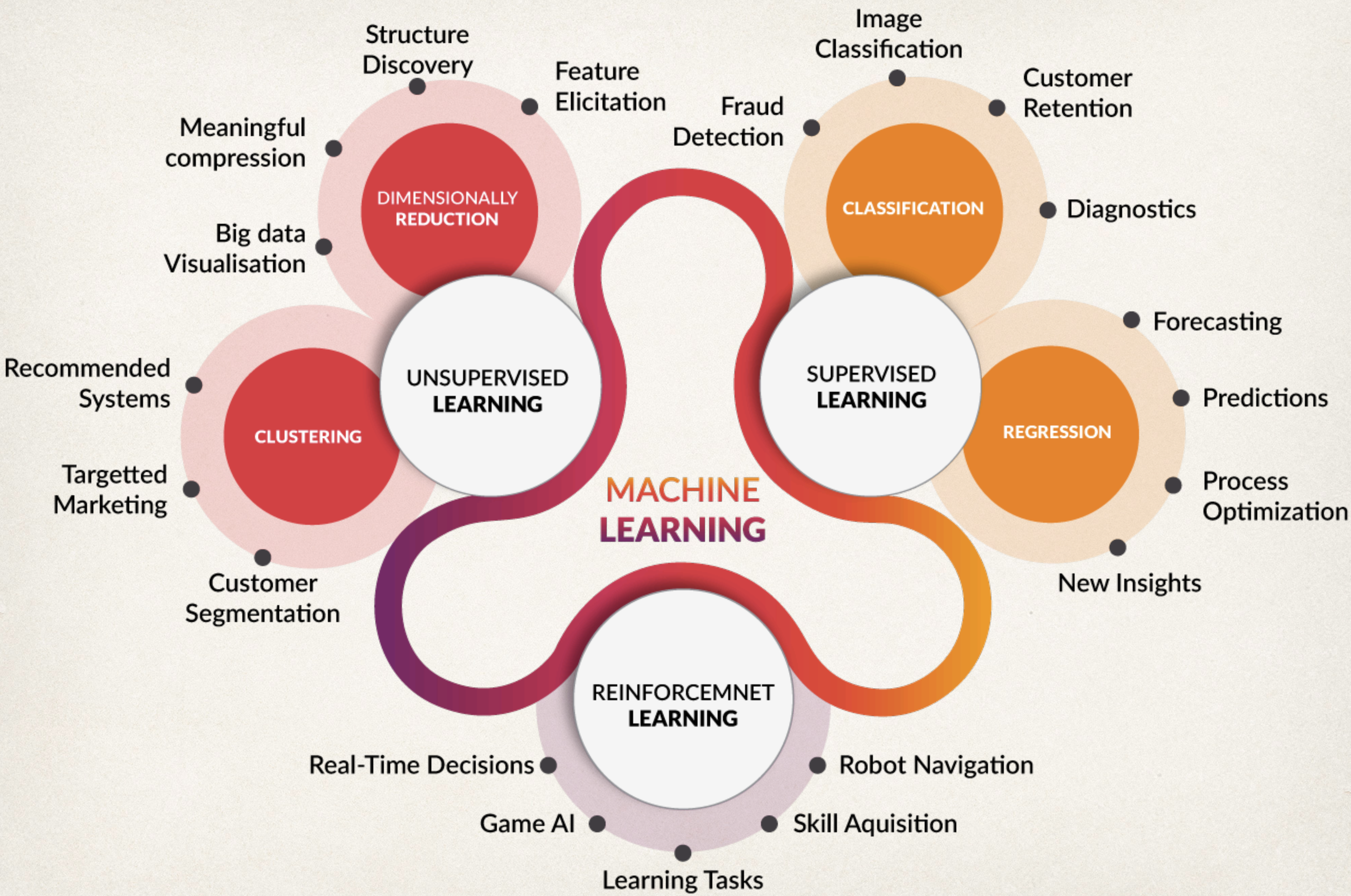


Data visualization

- Graphical representation may reveal important features of the data
    - find correlations, identify range, etc.
- Identify features which may require transformations, e.g. see outliers or skewness in data
- It helps to identify a strategy how to deal with different features

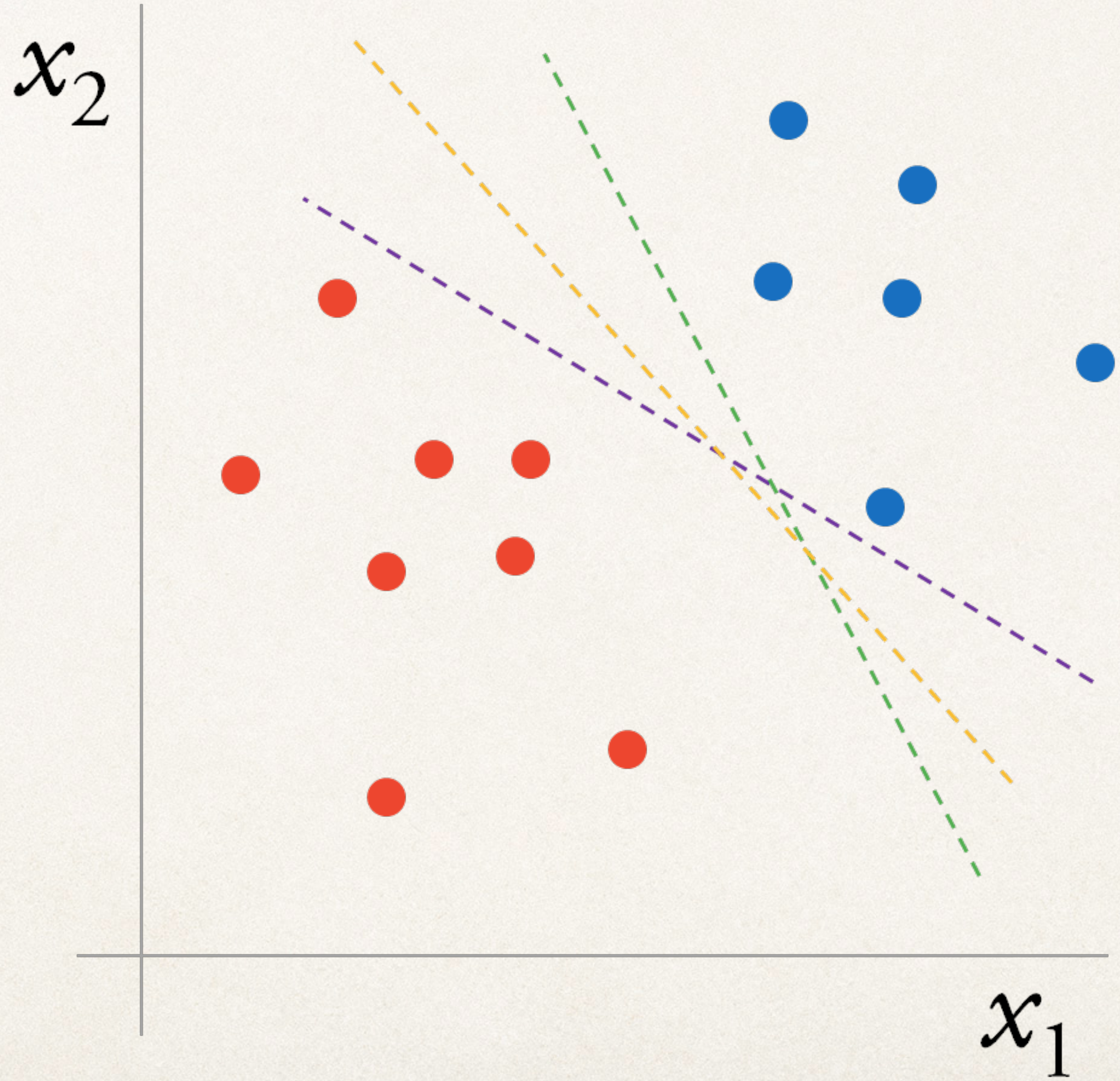


Data Science

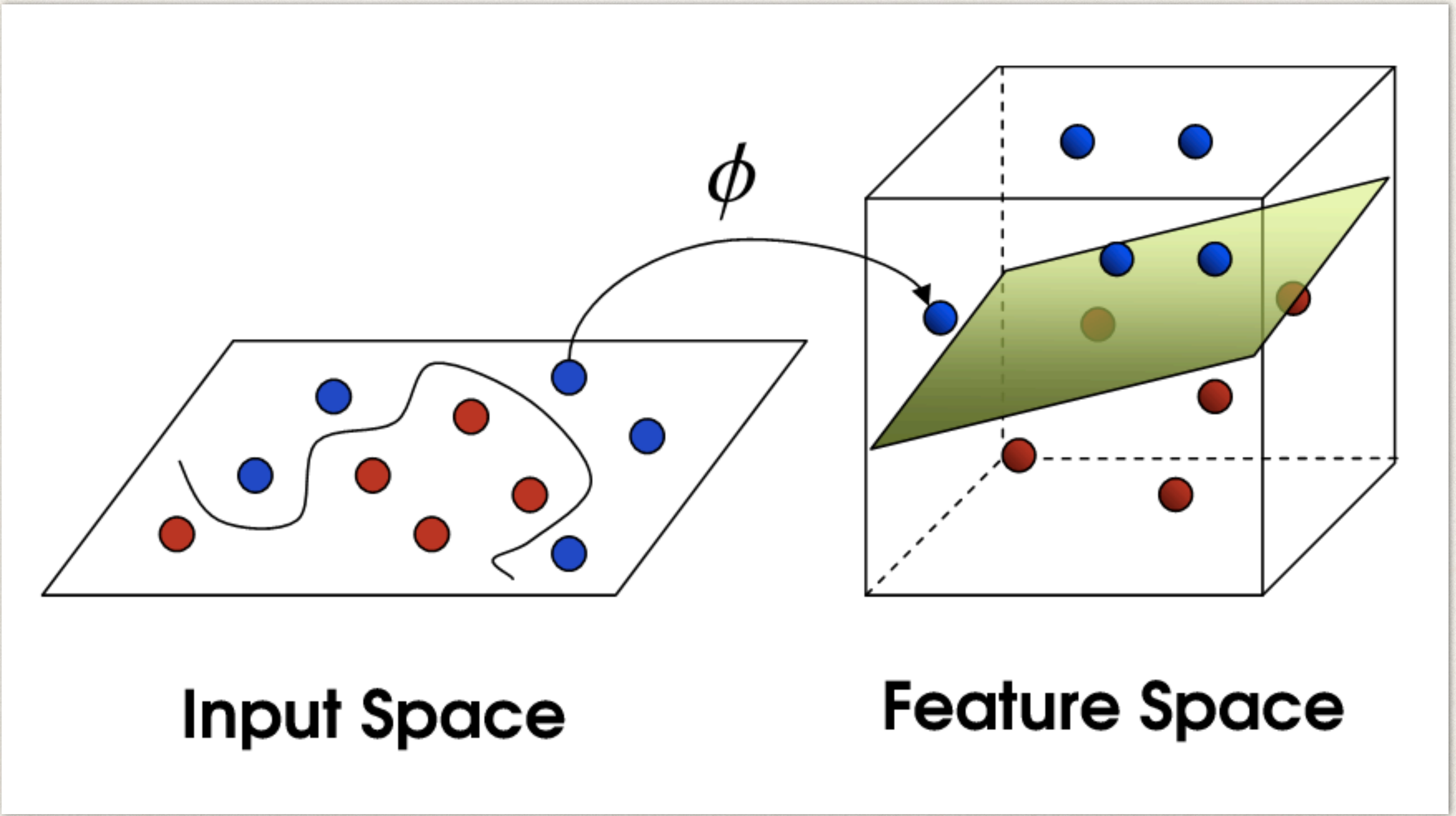

MACHINE
LEARNING
UNSUPERVISED
LEARNING
DIMENSIONALLY
REDUCTION
Structure
Discovery
Feature
Elicitation
Meaningful
compression
Big data
Visualisation
CLUSTERING
Recommended
Systems
Targetted
Marketing
Customer
Segmentation
SUPERVISED
LEARNING
CLASSIFICATION
Image
Classification
Customer
Retention
Fraud
Detection
Diagnostics
REGRESSION
Forecasting
Predictions
Process
Optimization
New Insights
REINFORCEMNET
LEARNING
Real-Time Decisions
Robot Navigation
Game AI
Skill Aquisition
Learning Tasks

Classification

Businesses
who
target
customers
good vs bad,
stay or leave

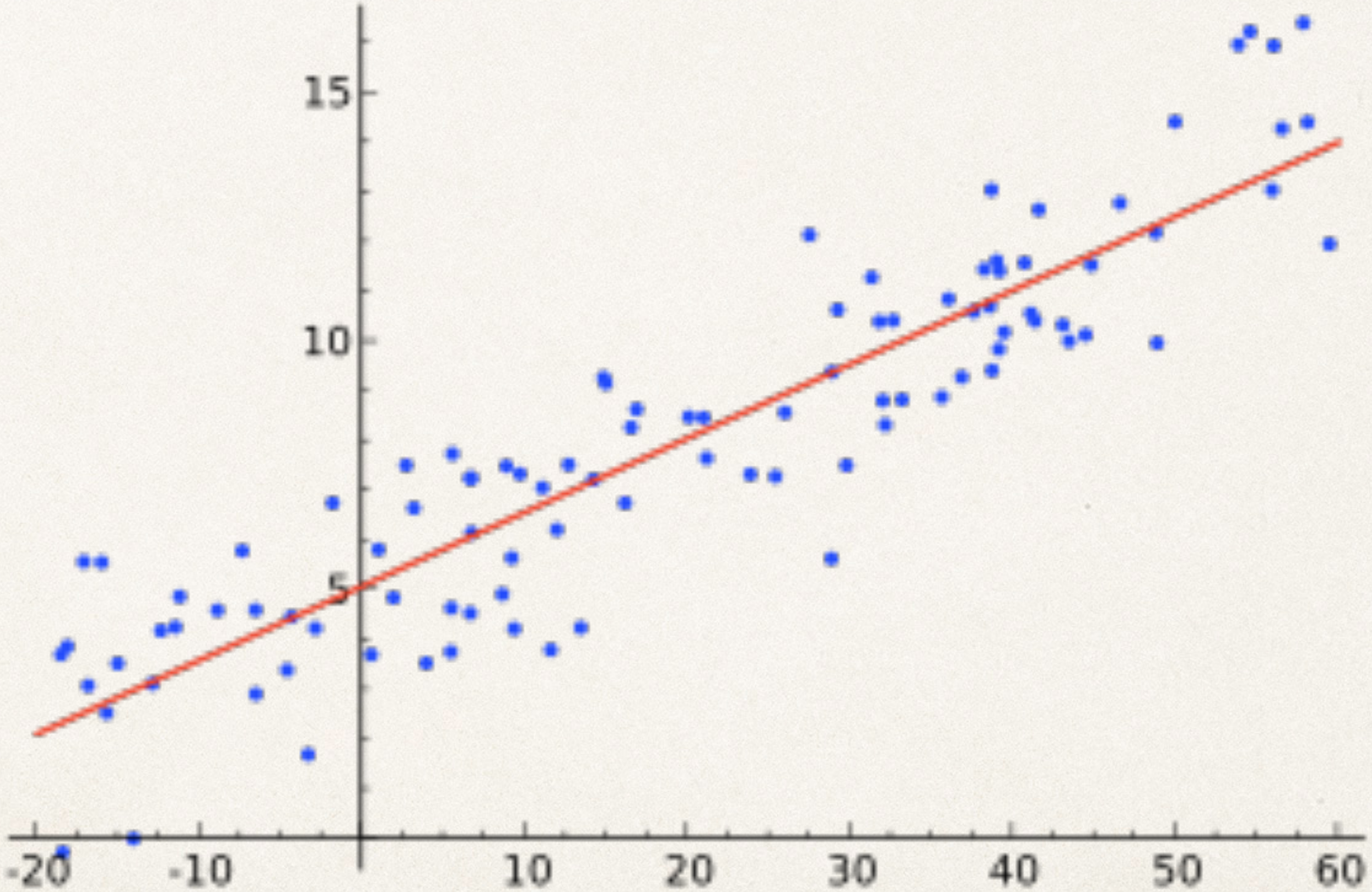


Feature space



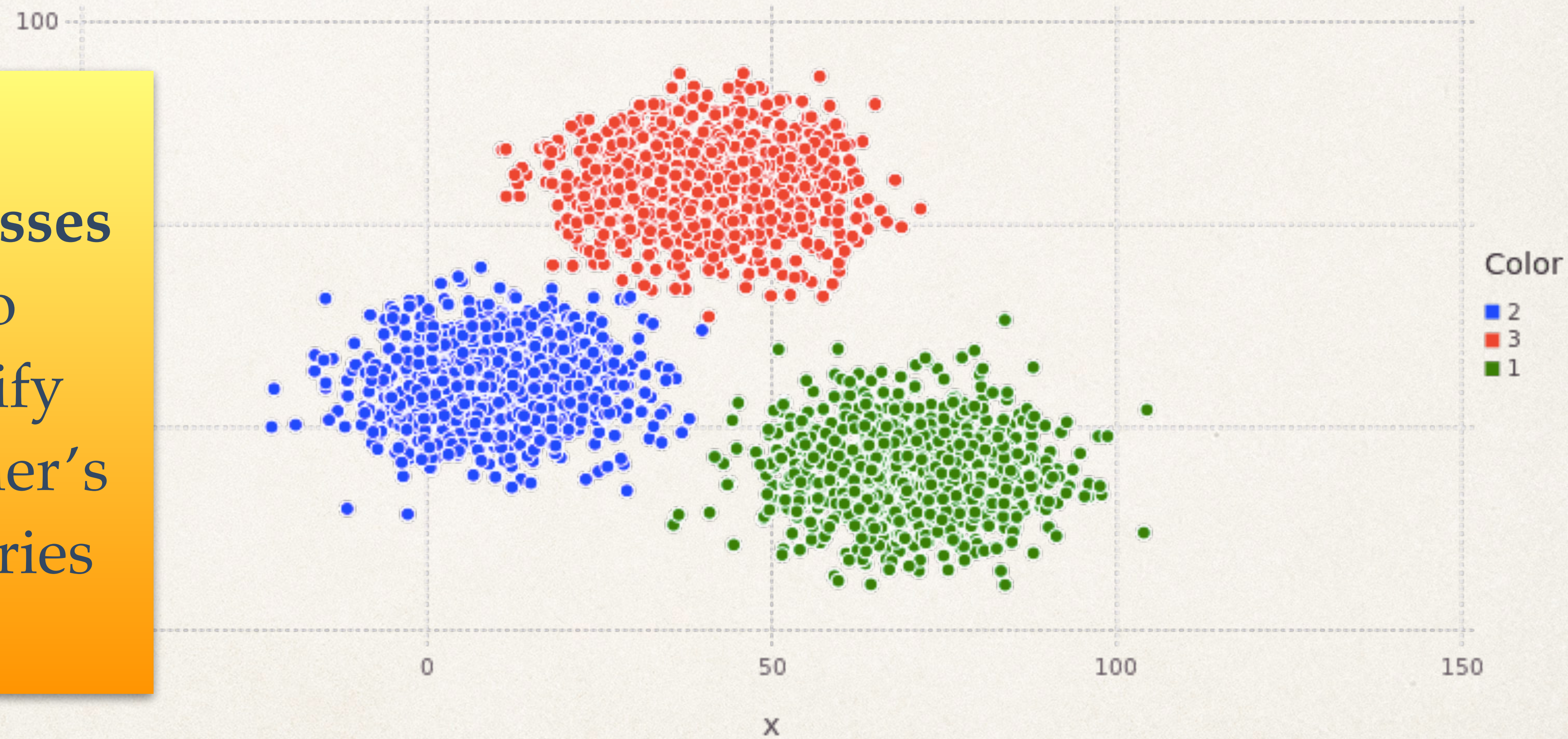
Regression

Businesses who predict customer's behavior, e.g. house prices,



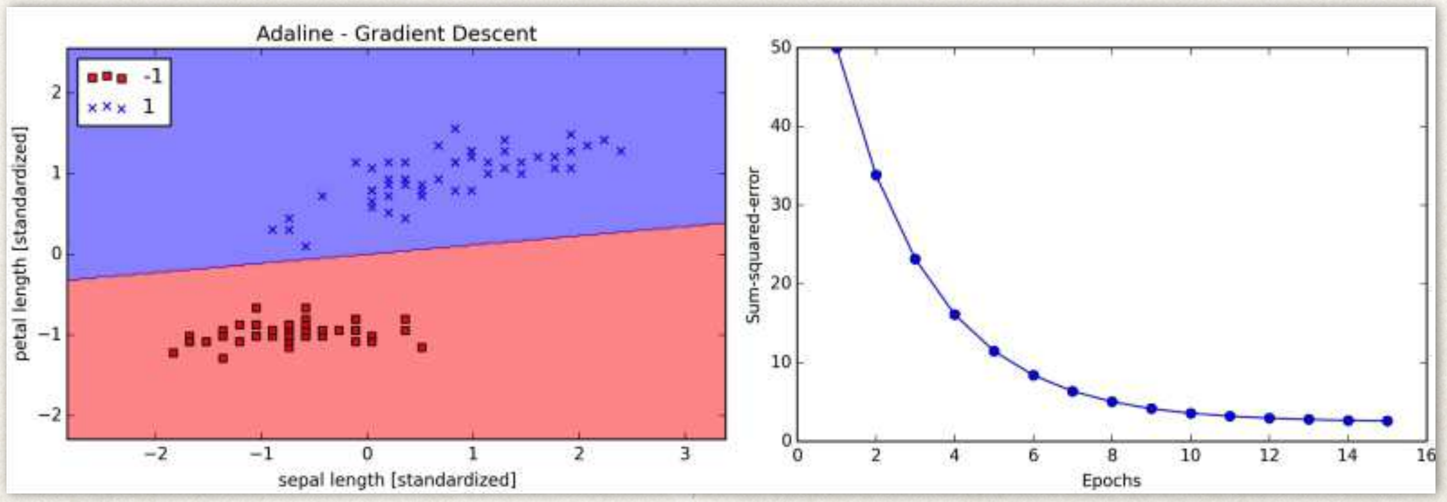
Clustering

Businesses who identify customer's categories



ML algorithm

- Inputs: X, e.g. timestamp, price, color, size, etc.
- Features: $\mathbb{X}$, transformed inputs
- Labels: y (stay vs leave)
- Weights: **W** (matrix)
- Activation function: ϕ (step function, e.g. sigmoid)
- Predictions: $z = \phi(\mathbf{W}^T \mathbb{X})$ yields (-1,1)
- Cost function: J(**W**), e.g. $\sum(y_i - z_i)^2/2$
- Algorithm: minimizes cost function & find best separation



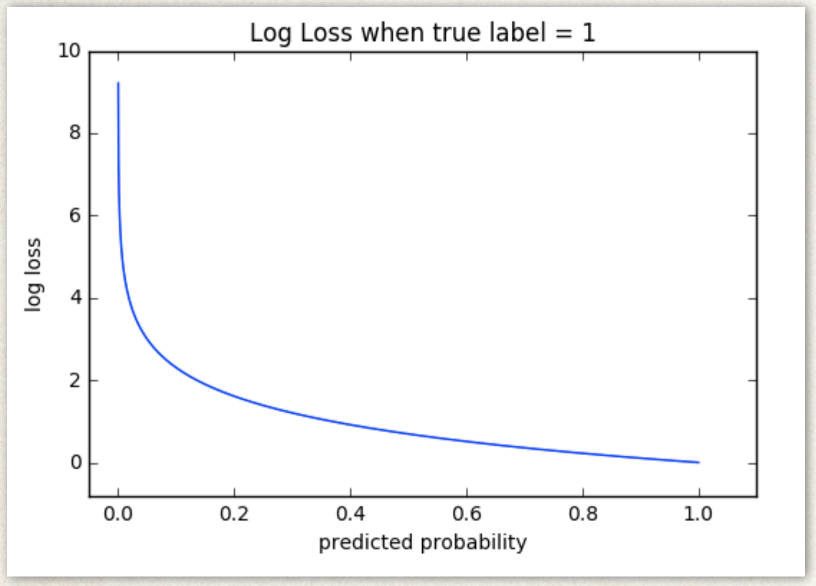
Loss functions

[Ref 1]

[Ref 2]

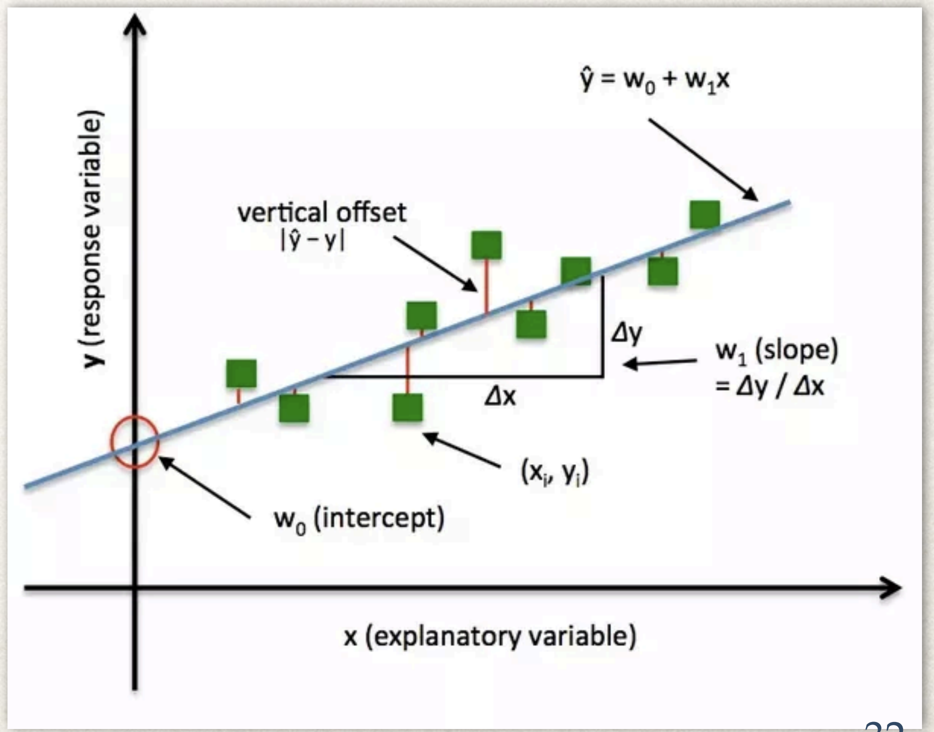
Cross-Entropy Loss

$$-(y\log(p) + (1-y)\log(1-p))$$



Classification	Regression
Log Loss	MSE
Focal Loss	MAE
Relative Entropy	Huber Loss
Exponential Loss	Log cosh Loss
Hinge Loss	Quantile Loss

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$



Regularization

Ref 1

Ref 2

- One of the major aspects of training the model is overfitting, when ML model tries too hard to capture the noise in your training dataset
- **Regularization** term is an addition to loss function which helps generalize the model. It helps to learn simpler model, induce models to be sparse, introduce group structure into learning problem

$$\min_f \sum_{i=1}^{n} V(f(x_i), y_i) + \lambda R(f)$$

  - **L1** or **Lasso regularization** adds penalty which is a sums of the absolute values of weights

$$Min(\sum_{i=1}^{n}(y_i - w_i x_i)^2 + p\sum_{i=1}^{n}|w_i|)$$

MSE+L1

  - **L2** or **Ridge regularization** adds penalty which is a sums of the squared values of weights

$$Min(\sum_{i=1}^{n}(y_i - w_i x_i)^2 + p\sum_{i=1}^{n}(w_i)^2)$$

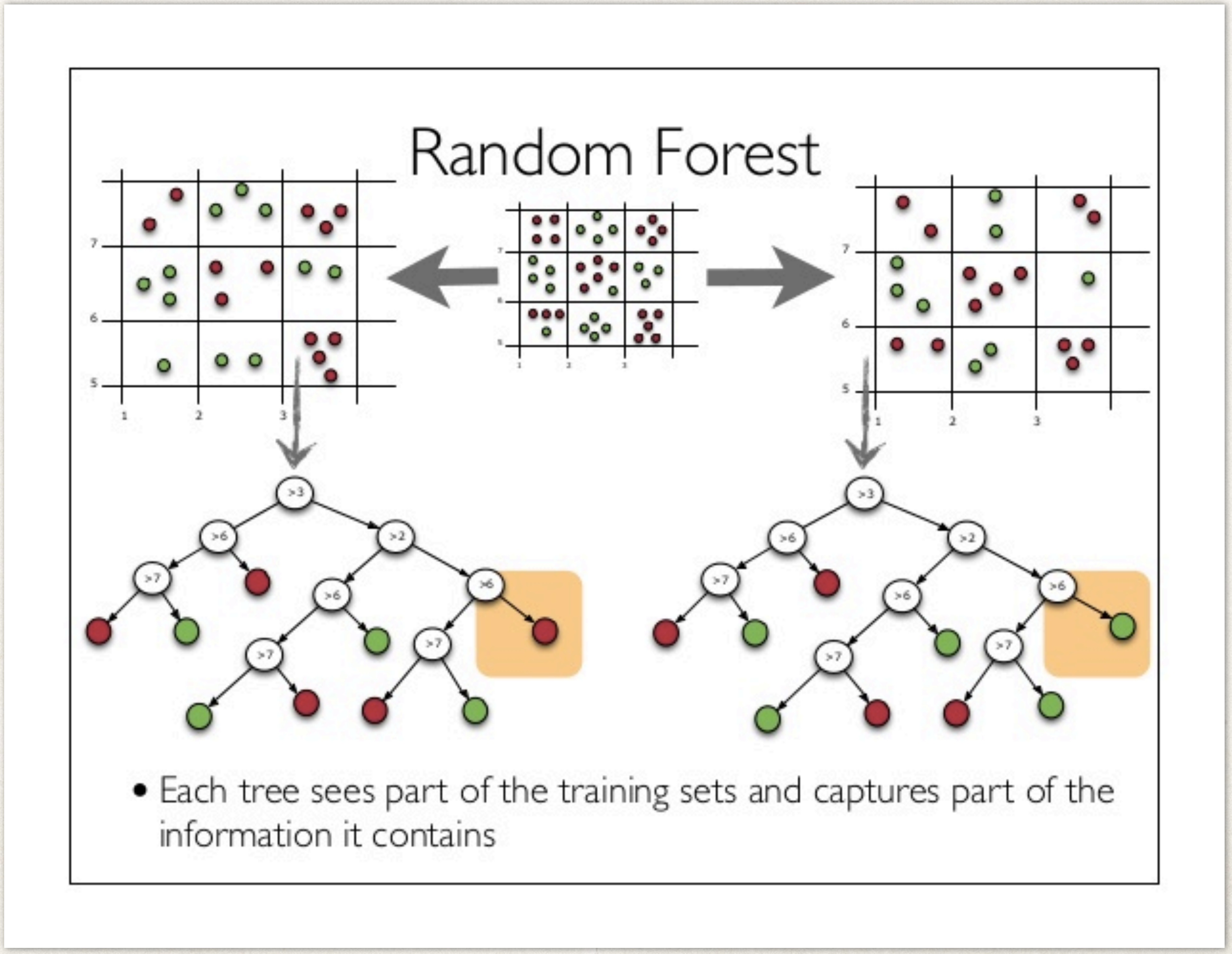
MSE+L2

- **Dropout** is a term introduced in NN context where hidden nodes are dropped randomly and allow model to generalize better
- **Early Stopping** is time regularization technique which stop training based on given criteria

TYPE	NAME	DESCRIPTION	ADVANTAGES	DISADVANTAGES
Linear	Linear regression	**The "best fit" line** through all data points. Predictions are numerical.	Easy to understand -- you clearly see what the biggest drivers of the model are.	X Sometimes too simple to capture complex relationships between variables. X Tendency for the model to "overfit".
Linear	Logistic regression	The adaptation of **linear regression** to problems of classification (e.g., yes/no questions, groups, etc.)	Also easy to understand.	X Sometimes too simple to capture complex relationships between variables. X Tendency for the model to "overfit".
Tree-based	Decision tree	A graph that uses a **branching method** to match all possible outcomes of a decision.	Easy to understand and implement.	X Not often used on its own for prediction because it's also often too simple and not powerful enough for complex data.
Tree-based	Random Forest	Takes the average of many decision trees, each of which is made with a sample of the data. Each tree is weaker than a full decision tree, but **by combining them we get better overall performance.**	A sort of "wisdom of the crowd". Tends to result in very high quality models. Fast to train.	X Can be slow to output predictions relative to other algorithms. X Not easy to understand predictions.
Tree-based	Gradient Boosting	Uses even weaker decision trees, that are increasingly **focused on "hard" examples.**	High-performing.	X A small change in the feature set or training set can create radical changes in the model. X Not easy to understand predictions.
Neural networks	Neural networks	Mimics the behavior of the brain. Neural networks are interconnected neurons that pass messages to each other. Deep learning uses several **layers of neural networks put one after the other.**	Can handle extremely complex tasks - no other algorithm comes close in image recognition.	X Very, very slow to train, because they have so many layers. Require a lot of power. X Almost impossible to understand predictions.

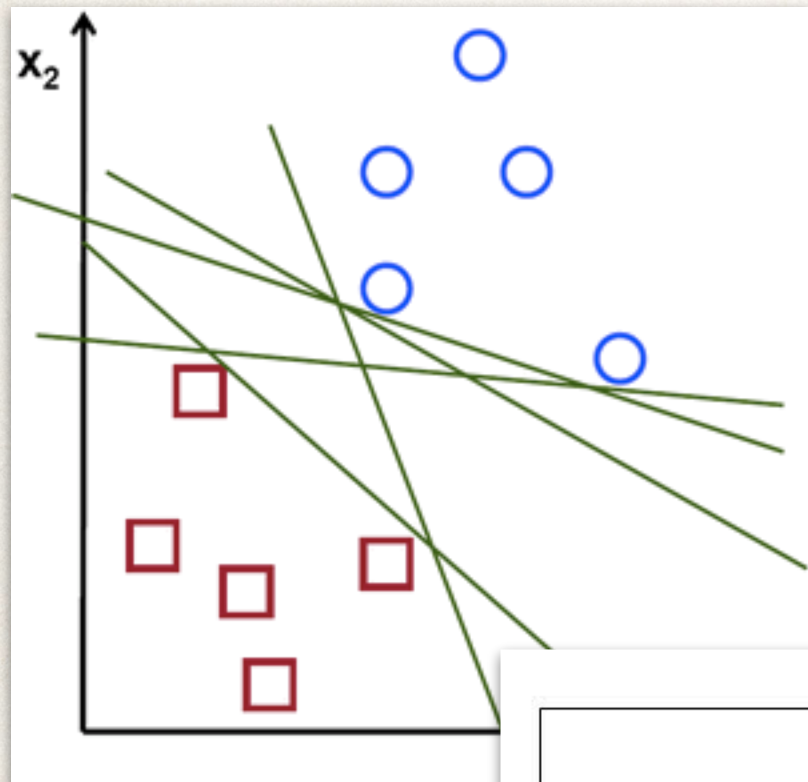
Random Forest

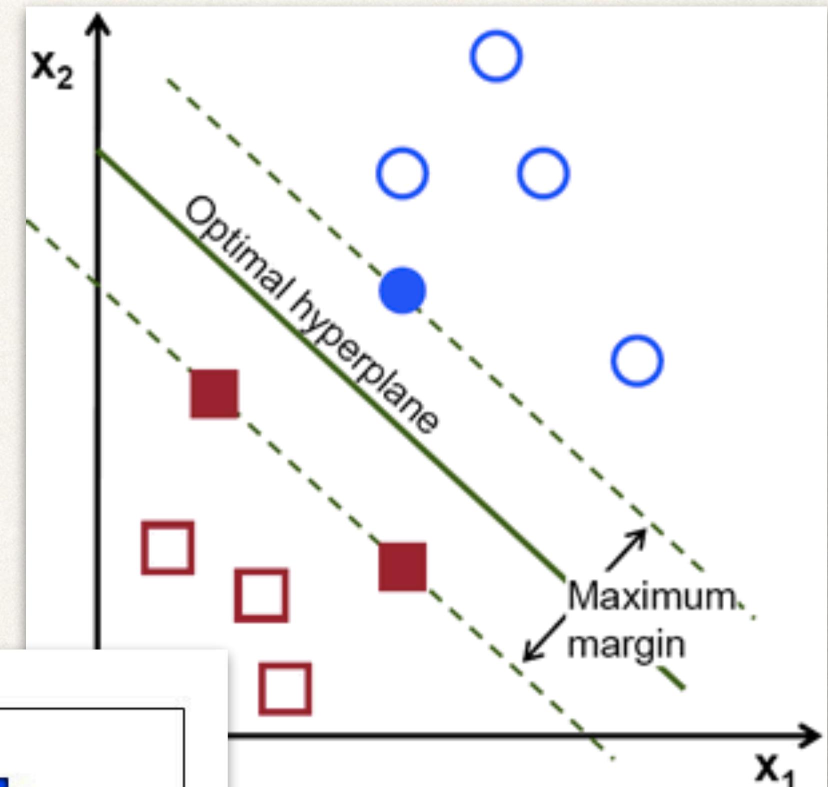
Ref

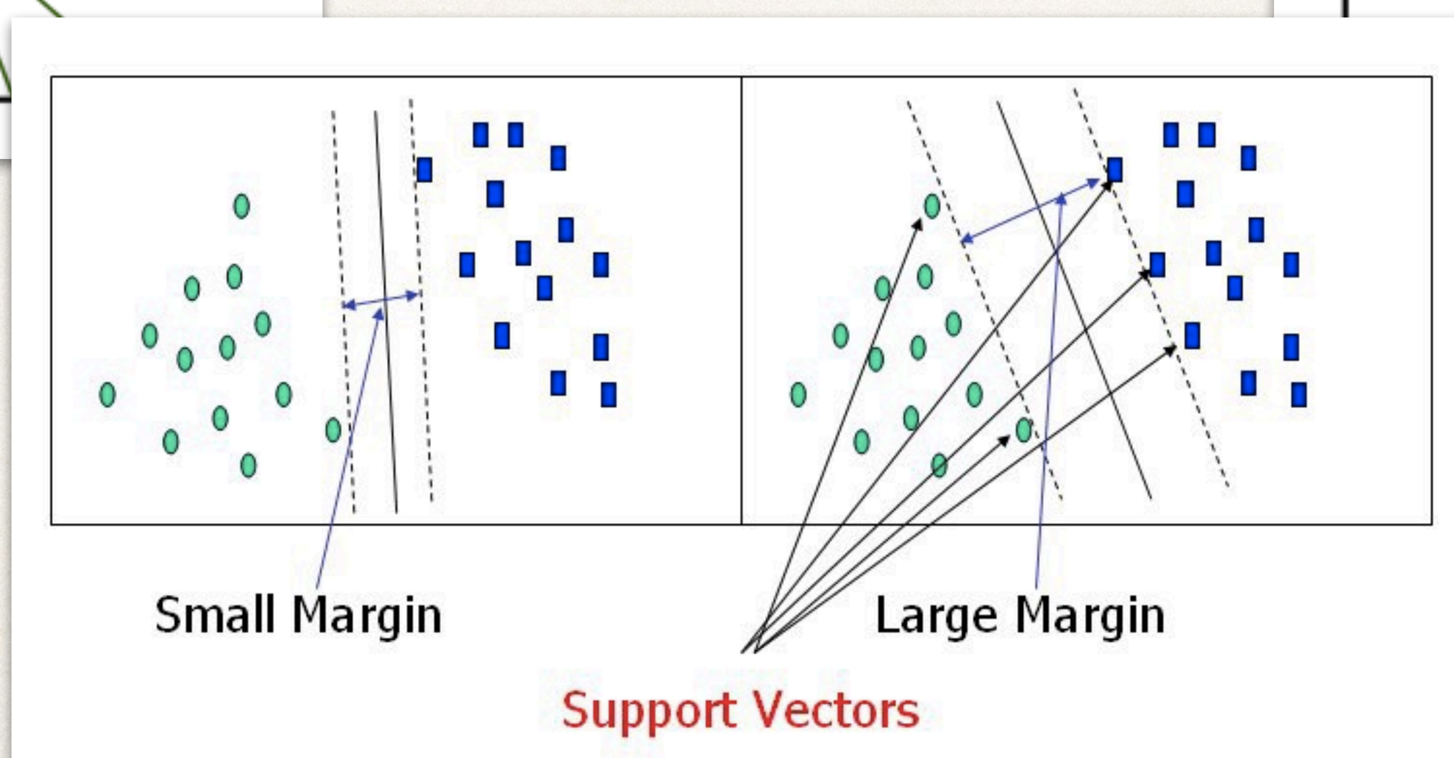


Support Vector Machines

[Ref]



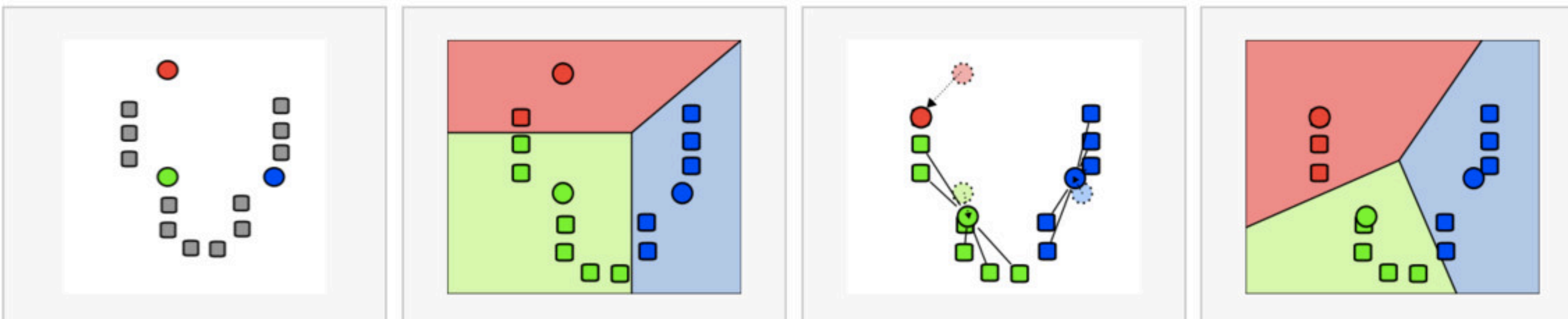




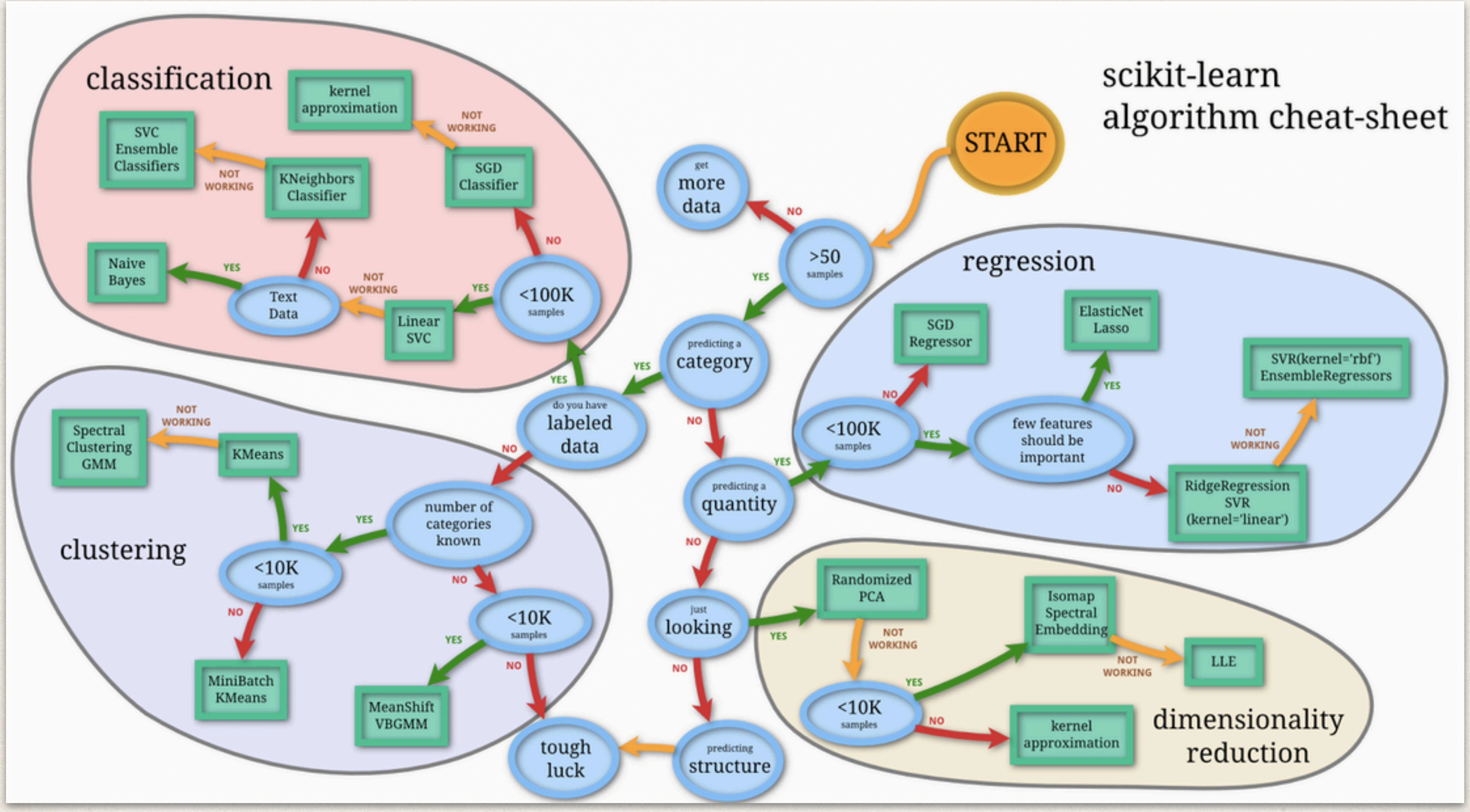
K-means clustering

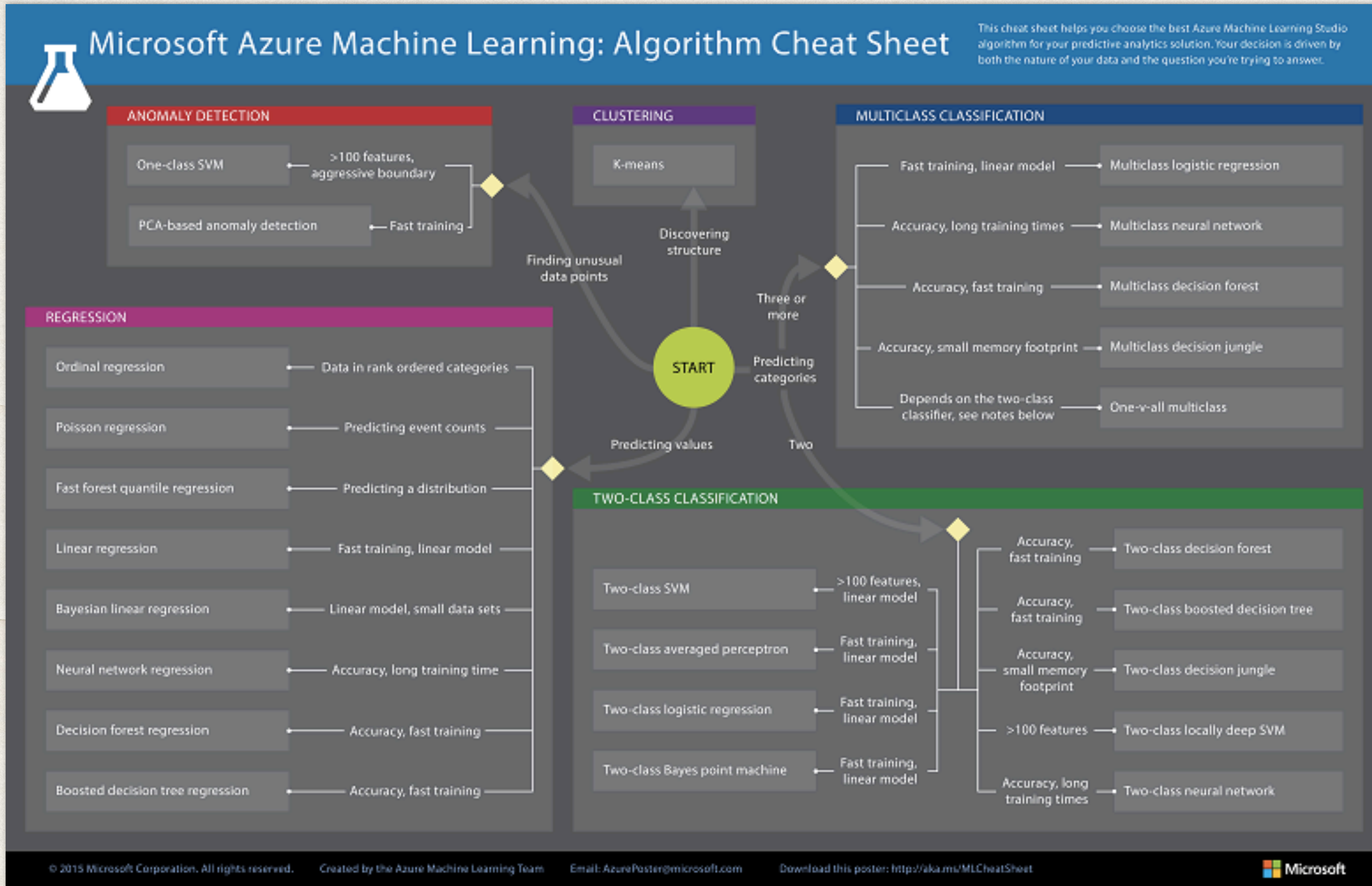
Ref

Demonstration of the standard algorithm



1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).
2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.
3. The centroid of each of the *k* clusters becomes the new mean.
4. Steps 2 and 3 are repeated until convergence has been reached.


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



Microsoft Azure Machine Learning: Algorithm Cheat Sheet
This cheat sheet helps you choose the best Azure Machine Learning Studio algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.
ANOMALY DETECTION
One-class SVM
>100 features, aggressive boundary
PCA-based anomaly detection
Fast training
CLUSTERING
K-means
Discovering structure
Finding unusual data points
START
Three or more
Predicting categories
Predicting values
Two
MULTICLASS CLASSIFICATION
Fast training, linear model
Multiclass logistic regression
Accuracy, long training times
Multiclass neural network
Accuracy, fast training
Multiclass decision forest
Accuracy, small memory footprint
Multiclass decision jungle
Depends on the two-class classifier, see notes below
One-v-all multiclass
REGRESSION
Ordinal regression
Data in rank ordered categories
Poisson regression
Predicting event counts
Fast forest quantile regression
Predicting a distribution
Linear regression
Fast training, linear model
Bayesian linear regression
Linear model, small data sets
Neural network regression
Accuracy, long training time
Decision forest regression
Accuracy, fast training
Boosted decision tree regression
Accuracy, fast training
TWO-CLASS CLASSIFICATION
Two-class SVM
>100 features, linear model
Two-class averaged perceptron
Fast training, linear model
Two-class logistic regression
Fast training, linear model
Two-class Bayes point machine
Fast training, linear model
Accuracy, fast training
Two-class decision forest
Accuracy, fast training
Two-class boosted decision tree
Accuracy, small memory footprint
Two-class decision jungle
>100 features
Two-class locally deep SVM
Accuracy, long training times
Two-class neural network
© 2015 Microsoft Corporation. All rights reserved.
Created by the Azure Machine Learning Team
Email: AzurePoster@microsoft.com
Download this poster: http://aka.ms/MLCheatSheet
Microsoft

Engineering challenges of 21st century

[Ref]

- Advance personalized learning
- Make solar energy economical
- Enhance virtual reality
- Reverse-engineer the brain
- Engineer better medicine
- Advance Health informatics
- Restore and improve urban infrastructure
- Provide access to clean water
- Secure Cyberspace
- Prevent Nuclear terror
- Manage the Nitrogen cycle
- Develop carbon sequestration methods
- Engineer tools of scientific discovery

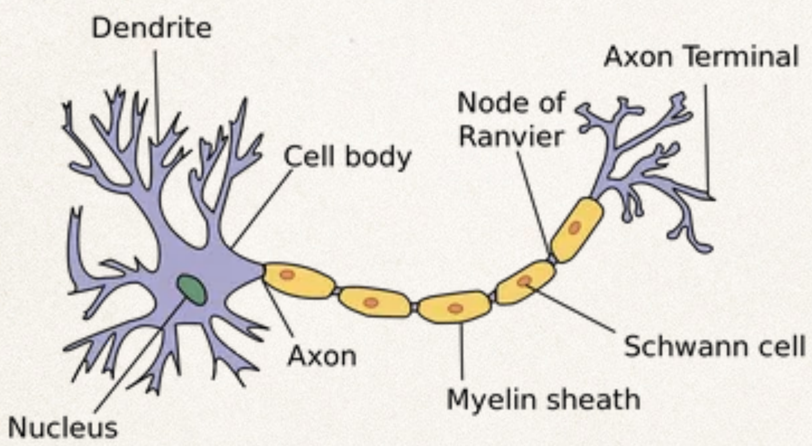
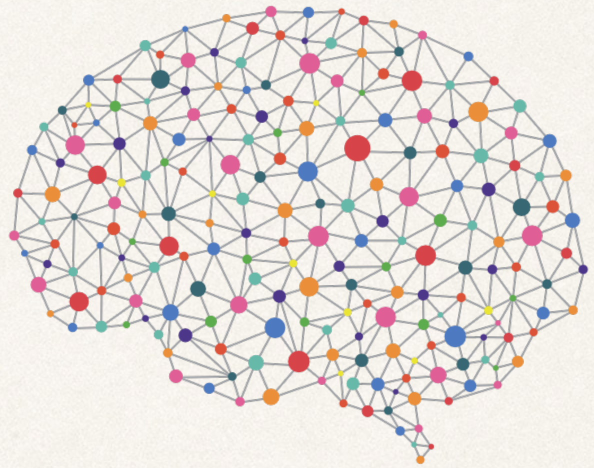
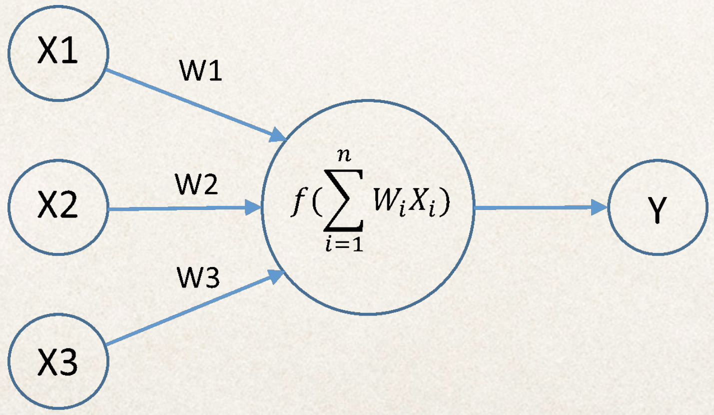
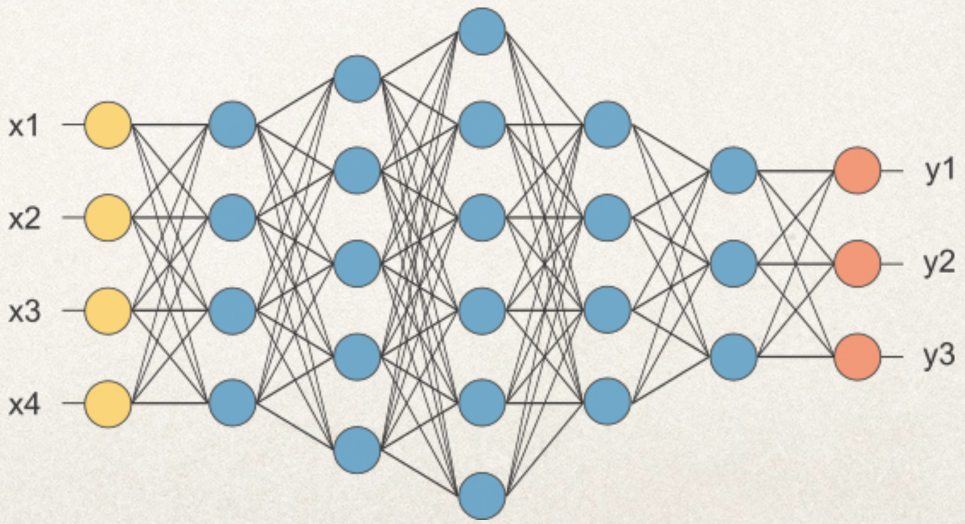
Neural Networks & Deep Learning

Deep Learning is a subfield of ML concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.

— Joson Brownlee (Machine Learning Mastery)

Neural Networks

Ref

	Neuron	Network
Biological		
Artificial		

Where NN/DL is used already

Life

- image recognitions
- language translation
- audio transcripts

Business

- returning customers
- house value predictions
- credit risks assessments

Medicine

- diabetic retinopathy
- patience admission to hospitals
- early cancer detection

Science

- physics, jet identification
- chemistry, predicting properties of molecules
- natural science, whales detection

Robotics

- self-driving cars
- robot movements
- end-to-end robotic control

Computers & IT

- data placement
- network optimization
- process scheduling

and, much more: games, manufacturing, mobile, social media, etc.

Where ML/DL can be used

Anywhere We're Using Heuristics To Make a Decision!

Compilers: instruction scheduling, register allocation, loop nest parallelization strategies, …

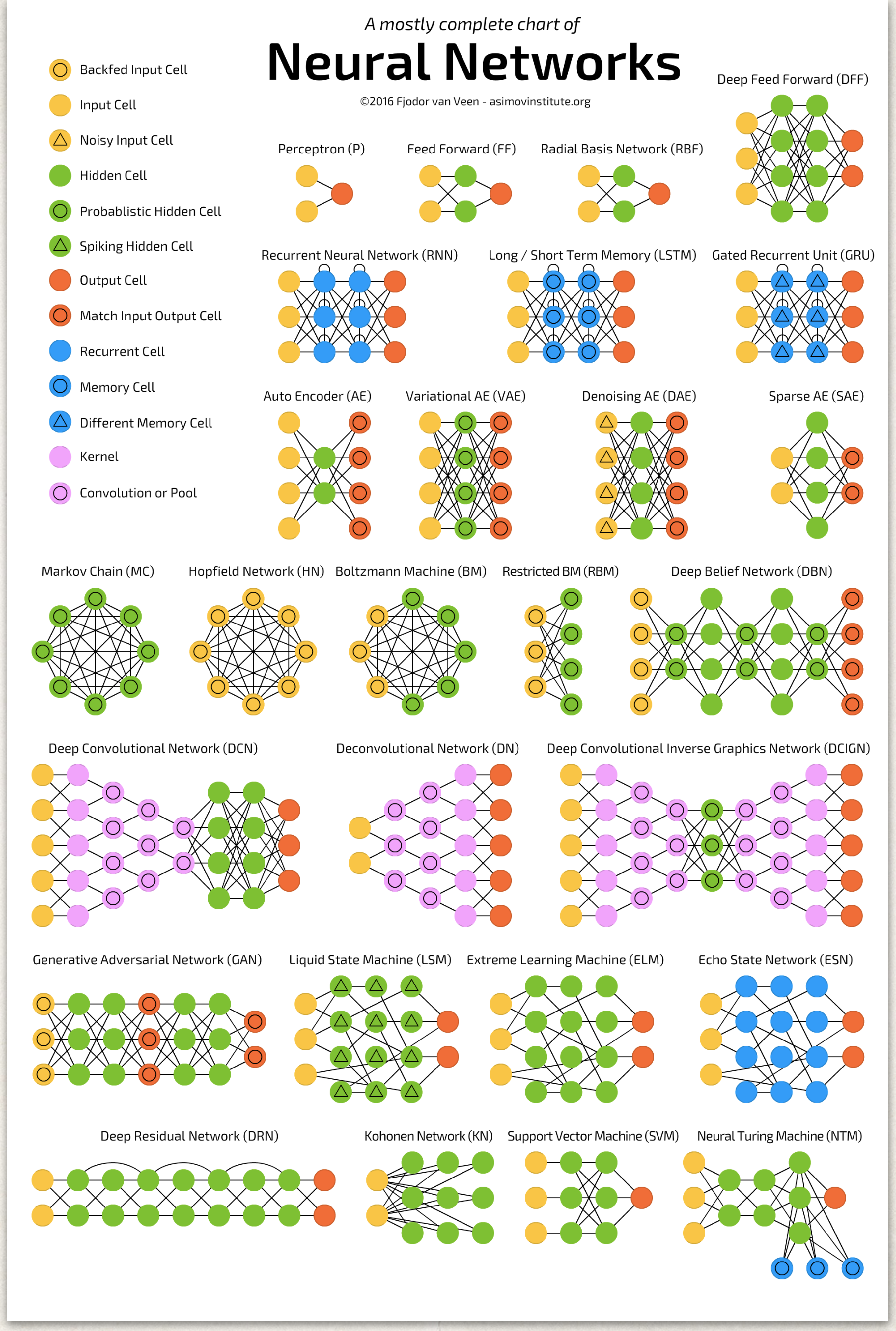
Networking: TCP window size decisions, backoff for retransmits, data compression, ...

Operating systems: process scheduling, buffer cache insertion/replacement, file system prefetching, …

Job scheduling systems: which tasks/VMs to co-locate on same machine, which tasks to pre-empt, ...

ASIC design: physical circuit layout, test case selection, …

Jeff Dean, Google Brain Team



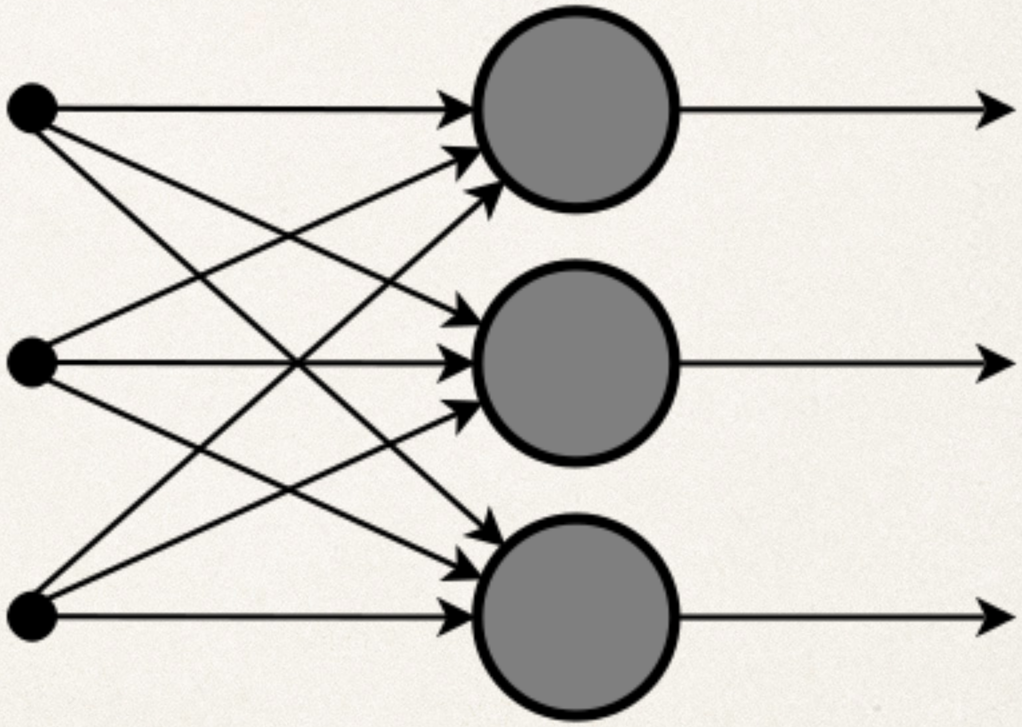
Ref 1

Ref 2

Ref 3

Feed-forward Neural Network

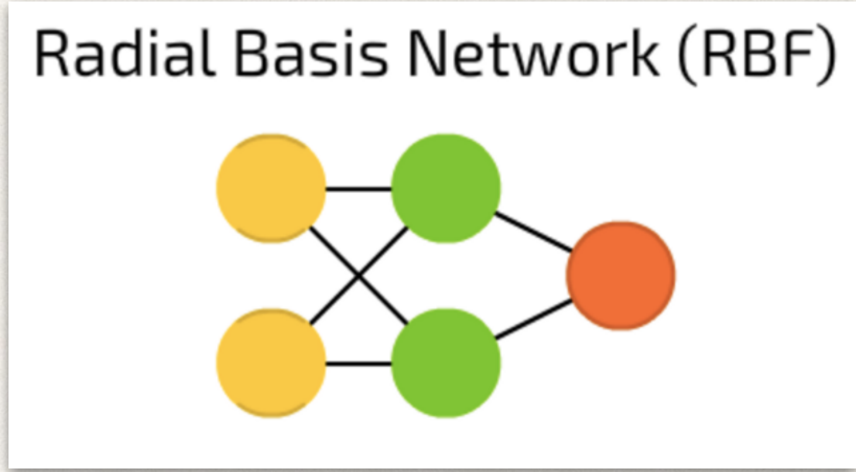
- Simplest form of ANN
- The data passes through input nodes and exit on the output nodes
- Easy to implement and combine with other type of ML algorithms
- Used in many ML tasks, from speech, image recognition to classification and computer vision

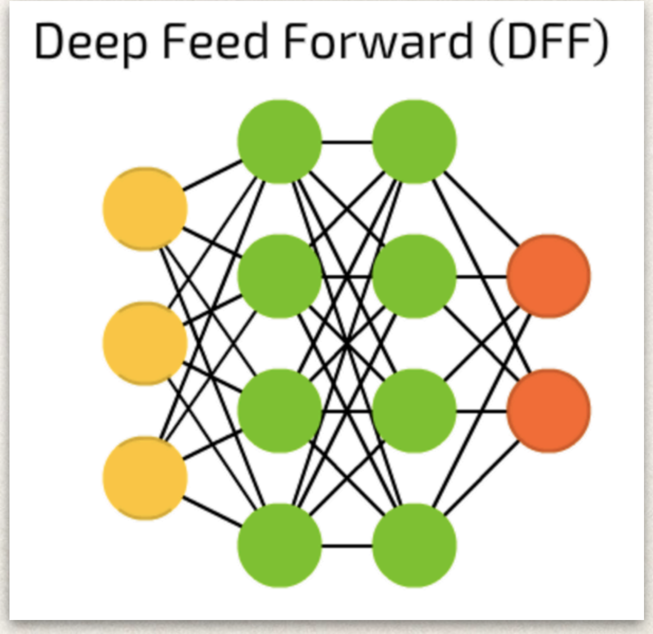


Radial Basis (RBF) and Deep Feed Forward (DFF) Networks

Ref

- RBF is feed-forward networks that uses radial basis function instead of logistic one
  - it is suitable to answer question as "how far are we from the target"
- DFF is a neural networks with more than one hidden layer
- Used in many ML tasks, e.g. classification and regression



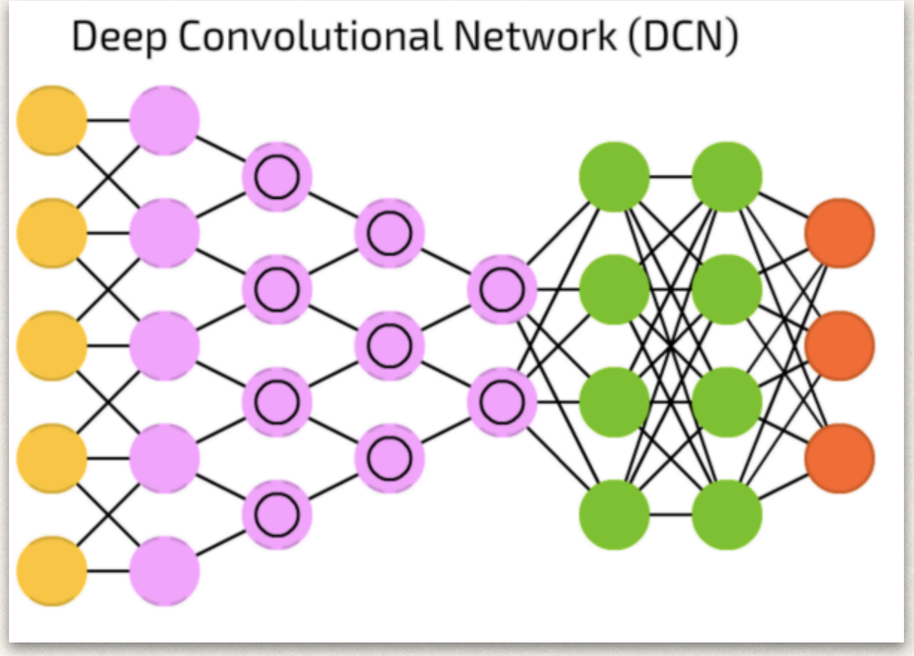


input cell hidden cell output cell

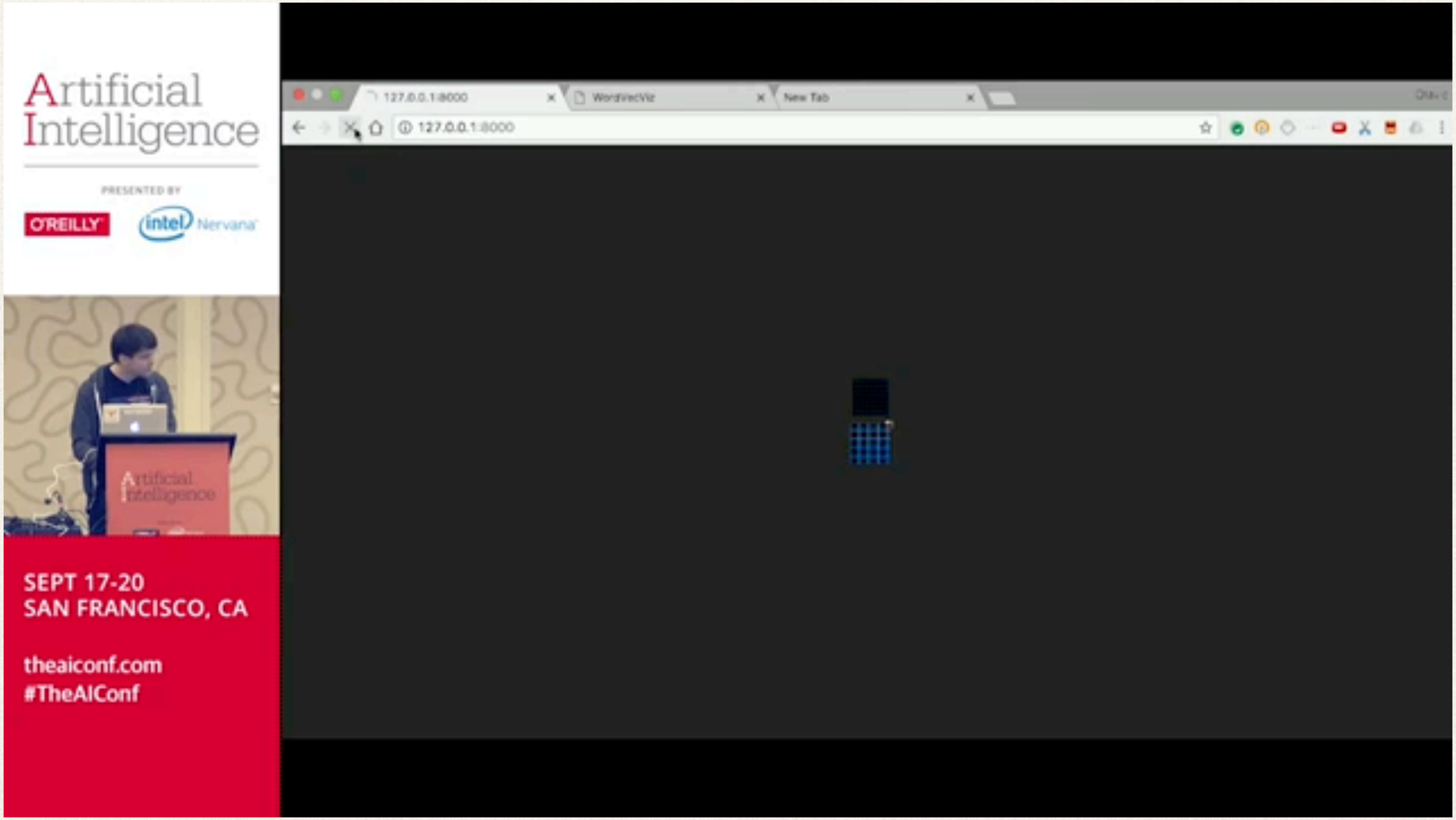
Convolutional NN: DCN

[Ref]

- NN which introduce two concepts: convolution to process input data and pooling to simplify it
  - use non-linear functions to reduce unnecessary features
- Successfully used for image classifications



Visualization of CDN

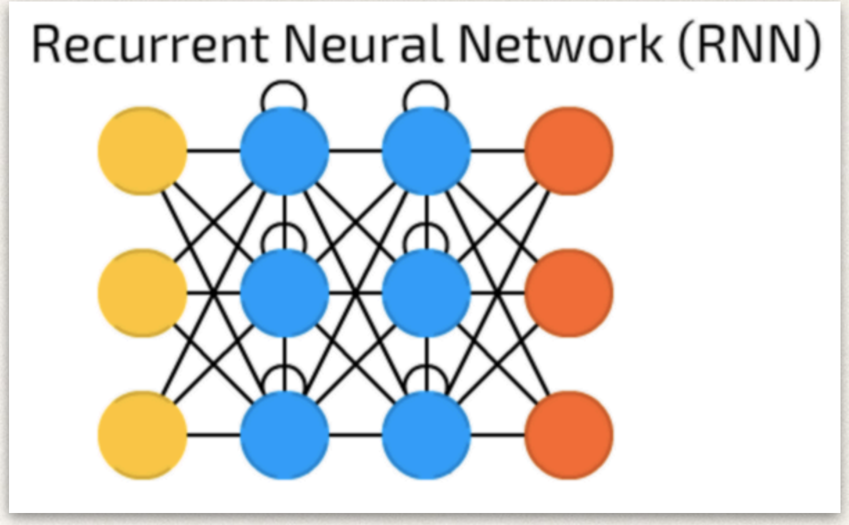


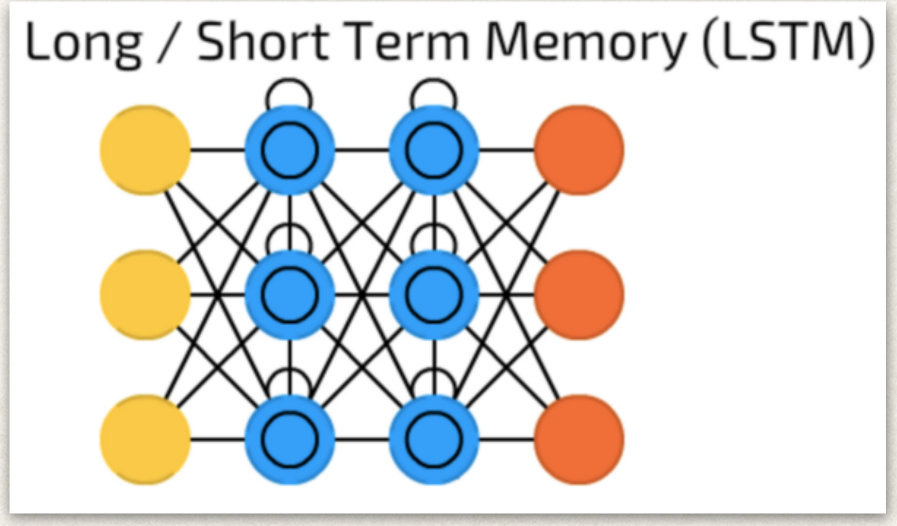
https://www.youtube.com/watch?v=Oqm9vsf_hvU

Recurrent Neural Networks (RNN), LSTM and GRU

Ref

- A FNN with Recurrent Cells: a hidden cell which received its own output with fixed delay
  - context is important, decision from past iterations can influence current state
  - a word can be analyzed only in context of previous words or sentences
- LSTM introduces concept of memory cell
  - "keep in mind" previous info, e.g. something that happen many frames ago
- GRUs are LSTMs with different gating
- Successfully used in text and speech recognitions



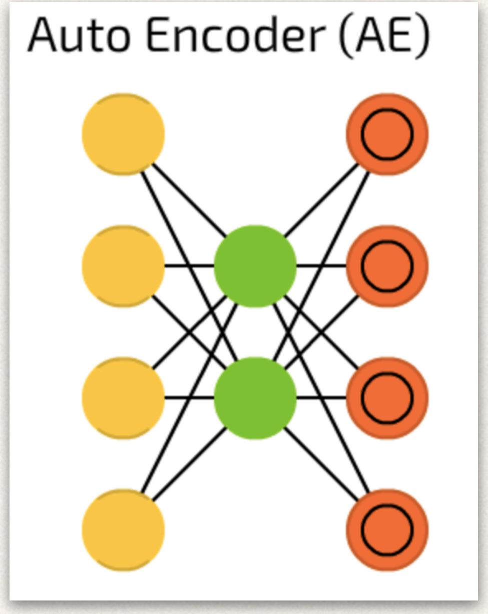


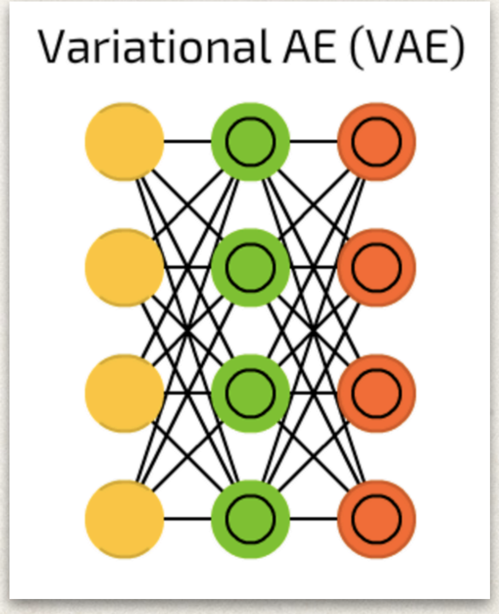
input cell output cell recurrent cell memory cell

Autoencoders: AE, VAE, DAE, SAE

Ref1, Ref2

- Autoencoders is special NN which find smaller representation of given input and search for common patterns
  - how can we generalize the data
  - It used for classification, clustering and feature compression
- VAEs compress probabilities instead of features
  - how strong is connection between two events
- DAE (De-noising AE) adds noise to input data and generalize it better
- SAE (Sparse AE) reveals some hidden grouping patterns in data, number of hidden cells more then input
- Used for data compression and dimensionality reduction



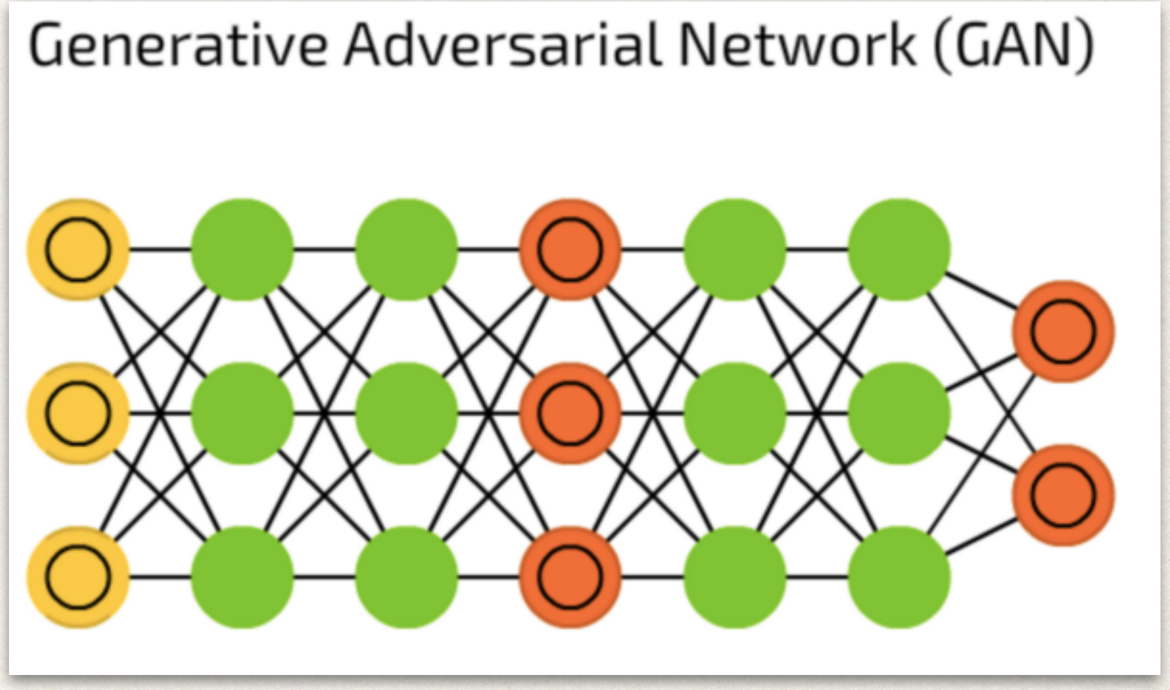


input cell   probabilistic hidden cell   match input output cell

Generative Adversarial Networks: GANs

Ref 1 Ref 2

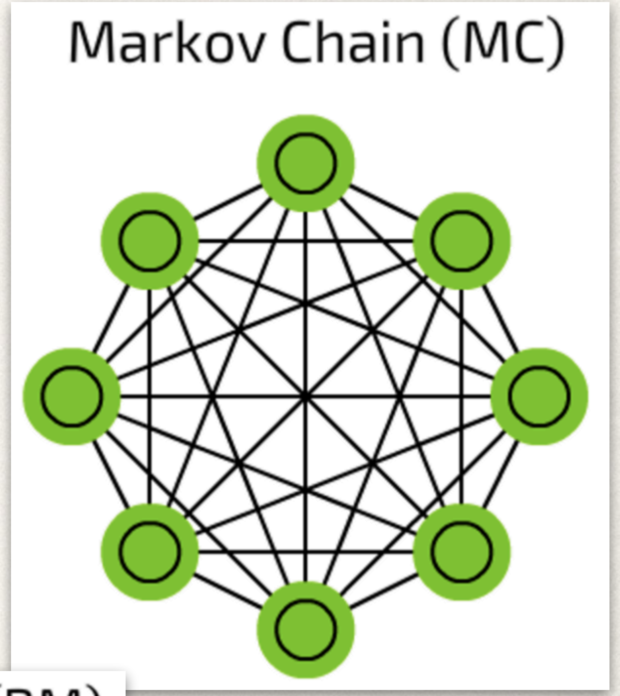
- GANs represents a huge family of double networks that are composed from generator and discriminator
    - generator generates an input according to given distribution
    - discriminator discriminates it based on our sample output
- Can be used to generate samples of data without prior knowledge about the data
- Used in modeling and generating high dimensional data

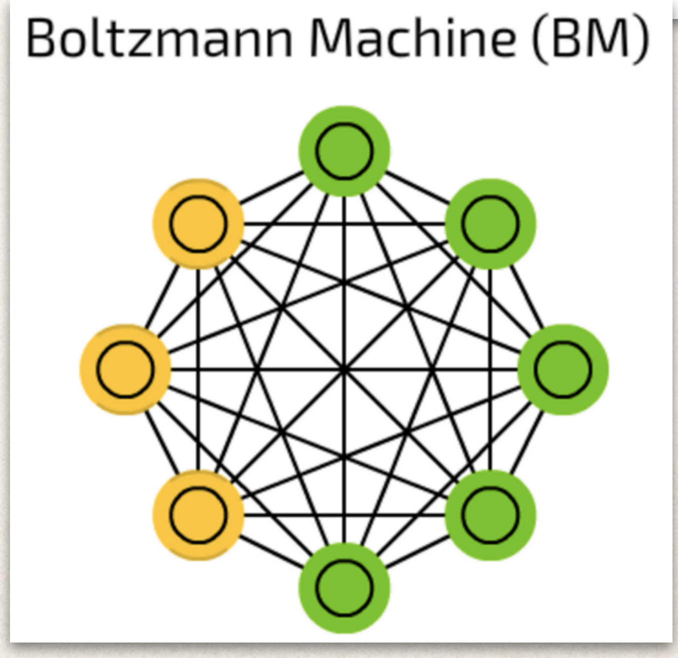


Graph NN: MC, HN, BM

Ref1, Ref2, Ref3

- graph networks deals with edges which have probabilities
    - after word hello there is word dear with probability of P1 and word you with probability P2
- Hopfield Network (HN) are trained on limited set of samples to reproduce full set
- Boltzman Machine (BM) networks are similar to HN where some cells are marked as input and remain hidden
- Used for feature detection and extractions





input cell **probabilistic hidden cell**

Data Science recipe

- Understand your data: preprocessing, cleaning, augmentation, one-hot-encoding
- Categorize the problem: classification, regression, clustering, dimensionality reduction
- Choose the language and toolkit: R, Python, Hadoop+Spark, ML providers
- Choose the right technique: trees, bagging, stacking, boosting, (rank | weight) averaging
- Start coding using your favorite ML framework and visualization tools

Techniques

[Ref 1]

[Ref 2]

[Ref 3]

Ensembles

All models are wrong, but some are useful (George Box)

Sometimes intentionally built weak models are good blending candidates

- Bagging
  - building multiple models (typically of the same type) from different subsamples of the training dataset
- Boosting
  - building multiple models (typically of the same type) each of which learns to fix the predictions errors of a prior model in the chain
- Stacking
  - building multiple models (typically of the different types) and supervisor model that learns how to best combine the predictions of the primary model
- Weighting | Blending
  - combine multiple models into single prediction using different weight functions

Diversity is a key: use different un-correlated models, e.g. GBM, RF, SVM, NN

better to fight over-fitting

better to get lower errors

Bagging vs Boosting

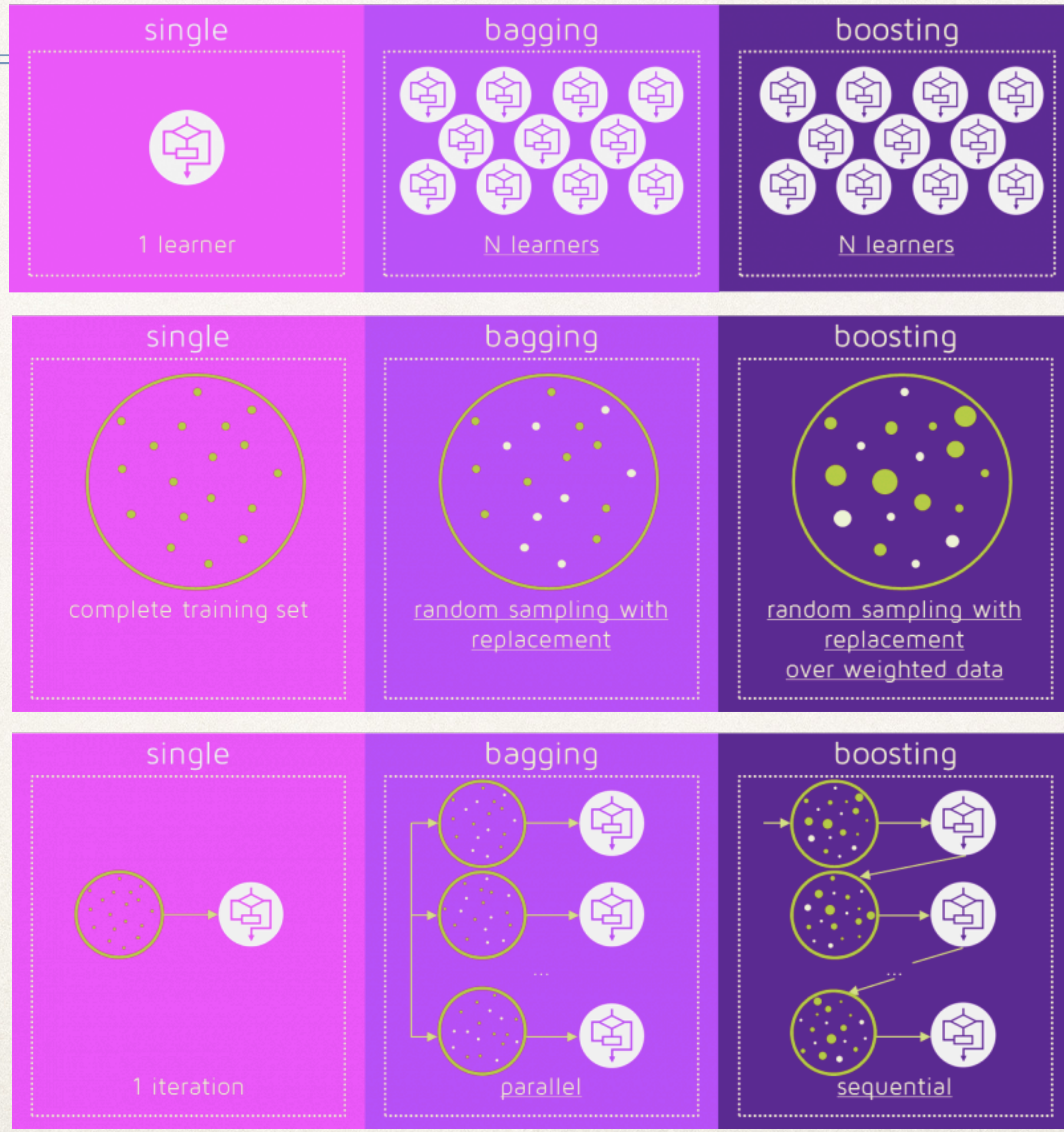
Ref

Similarities

Both are ensemble methods to get N learners from one

Generate several training sets by random sampling

Make final decision by averaging N learners or taking majority of them



Differences

build independently for Bagging, and Boosting tries to add new models that do well where previous models fail

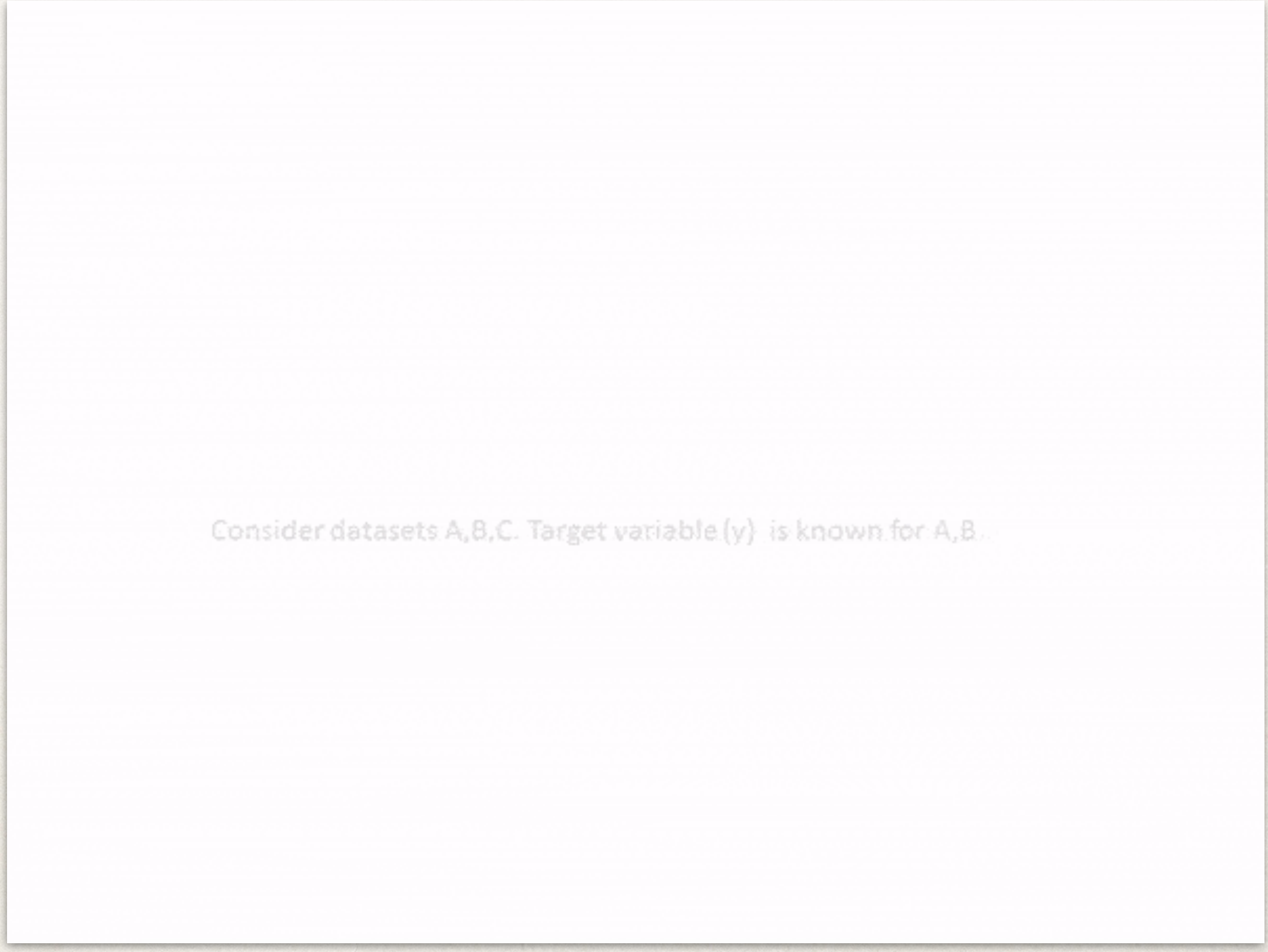
Boosting weights the data to scale in favor of most difficult cases

Bagging: equally weighted average
Boosting: weighted average, more weight to those who perform better on training set

Stacking

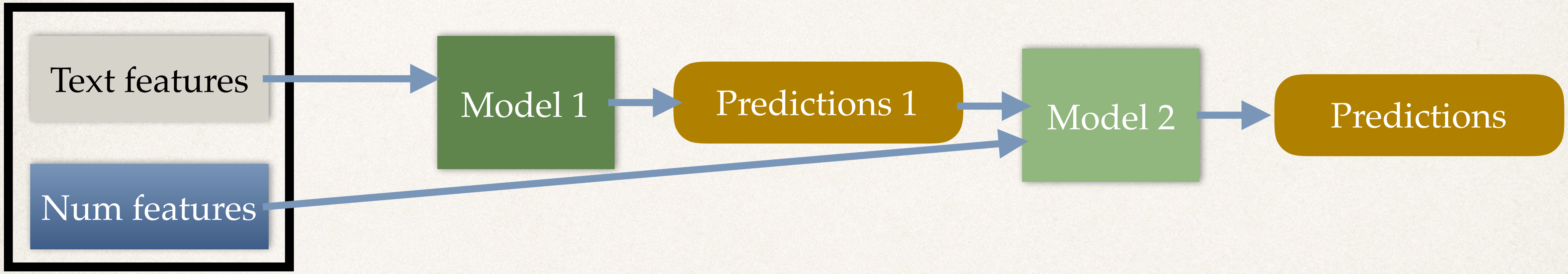
[Ref](#)

- Stacking (also called meta-ensembling) is a model ensembling technique used to combine information from multiple predictive models to generate a new model
- Usually outperform individual models used in ensemble, e.g. GBM+RF+NN
- Most effective when base models are independent
- May be applied at multiple level, e.g. stacking first set, then second set, etc.



Technical tricks

- Use one set of features (text) for simple model 1, and use numerical features and model1 prediction for model 2, etc.



- Use chained models: build stand-alone model for G, then used in next model, e.g. F=>G=>B=>A
- Feature engineering:
  - one-hot-encoding, leave-one-out, word embedding and add them to original data set
  - split days into years, months, dates and threat them as categorical variables
  - aggregate values, e.g. sum all numerical values in a row and/or use its mean/median
  - handle missing values, e.g. apply mean across column or even apply additional training to find their values

Tools and frameworks

Classification
Regression
Clustering
Dimensionality reduction
Model selection
Preprocessing

DataFrame
data.table
ggplot
xgboost
NeuralNetwork
Trees, Bagging

ML for “standard” use-cases

- In most cases you may rely on R or Python eco-system. In Python scikit-learn is de-facto standard, in R all ML tools are available through 3rd party packages via install.packages(<pkg>)
- Majority of DataScientists in kaggle competition use xgboost, the distributed gradient boosting library (both R and Python APIs are available) based on parallel tree boosting algorithm (aka GBDR, GBM)
- Less known libraries are:
  - Weka is Waikato Environment for Knowledge Analysis is a suite of machine learning software written in Java, developed at the University of Waikato, New Zealand (GUI environment)
  - StackNet is a computational, scalable and analytical Meta modeling framework (developed by top-level kaggle competitor Kaza-Nova and used in many competition to won first places). Written in Java and uses uses Wolpert's stacked generalization to improve accuracy of ML models. The network is built iteratively one layer at a time (using stacked generalization), each of which uses the final target as its target.
  - h2o Open Source Fast Scalable Machine Learning Platform For Smarter Applications (Deep Learning, Gradient Boosting, Random Forest, Generalized Linear Modeling (Logistic Regression, Elastic Net), K-Means, PCA, Stacked Ensembles, Automatic Machine Learning (AutoML)

Neural network frameworks

- Torch is an open source machine learning library, a scientific computing framework, and a script language based on the Lua programming language.
- Theano is a numerical computation library for Python that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. In Theano, computations are expressed using a NumPy-esque syntax and compiled to run efficiently on either CPU or GPU architectures.
- Caffe is a deep learning framework (C++ and Python) made with expression, speed, and modularity in mind.
- TensorFlow is an open-source software library (C++, Python, Go) for data-flow programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.
- PyTorch is a deep learning framework for fast, flexible experimentation. It is Tensors and Dynamic neural networks in Python with strong GPU acceleration.
- Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano
- Apache MXNet framework (Python and R) is a modern deep learning framework
- onnx.ai is an Open Neural Network exchange format which allows to import and export Neural Network models from/to different frameworks

Visualization of Neural Networks

- TensorFlow playground: provides an intuitive web based interface to train Neural Networks for a given dataset
- ConvNetJS is a Javascript library for training Deep Learning models (Neural Networks) entirely in your browser
- LSTMVis - visual analysis for Recurrent Neural Networks
- Netron is a visualizer for Deep Learning and machine learning models
- Ann-visualizer, is a python library for visualizing Artificial Neural Networks
- Keras-vis is a high-level toolkit for visualizing and debugging your trained keras neural net models
- VisualDL is an open-source cross-framework web dashboard that richly visualizes the performance and data flowing through your neural network training

ML for Big Data

- Some datasets can't be trained with standard ML tools since they are too big to fit into memory, therefore you can't use "standard" tools like scikit-learn or R
- Gradient Boosting Algorithm (GBM) is a ML technique which produces a prediction model in a form of ensemble of weak prediction models, typically decision trees
    - Boosting is an ensemble technique in which the predictors are not made independently, but sequentially. Therefore a large dataset can be learned in "chunks" with GBM
- Vowpal Wabbit is online learning algorithm designed to deal with tera-features datasets
- Spark ML Big Data platform (MLlib), Spark is a technique to deal and process large datasets using Hadoop platform which now has a set of ML algorithms available as a part of platform

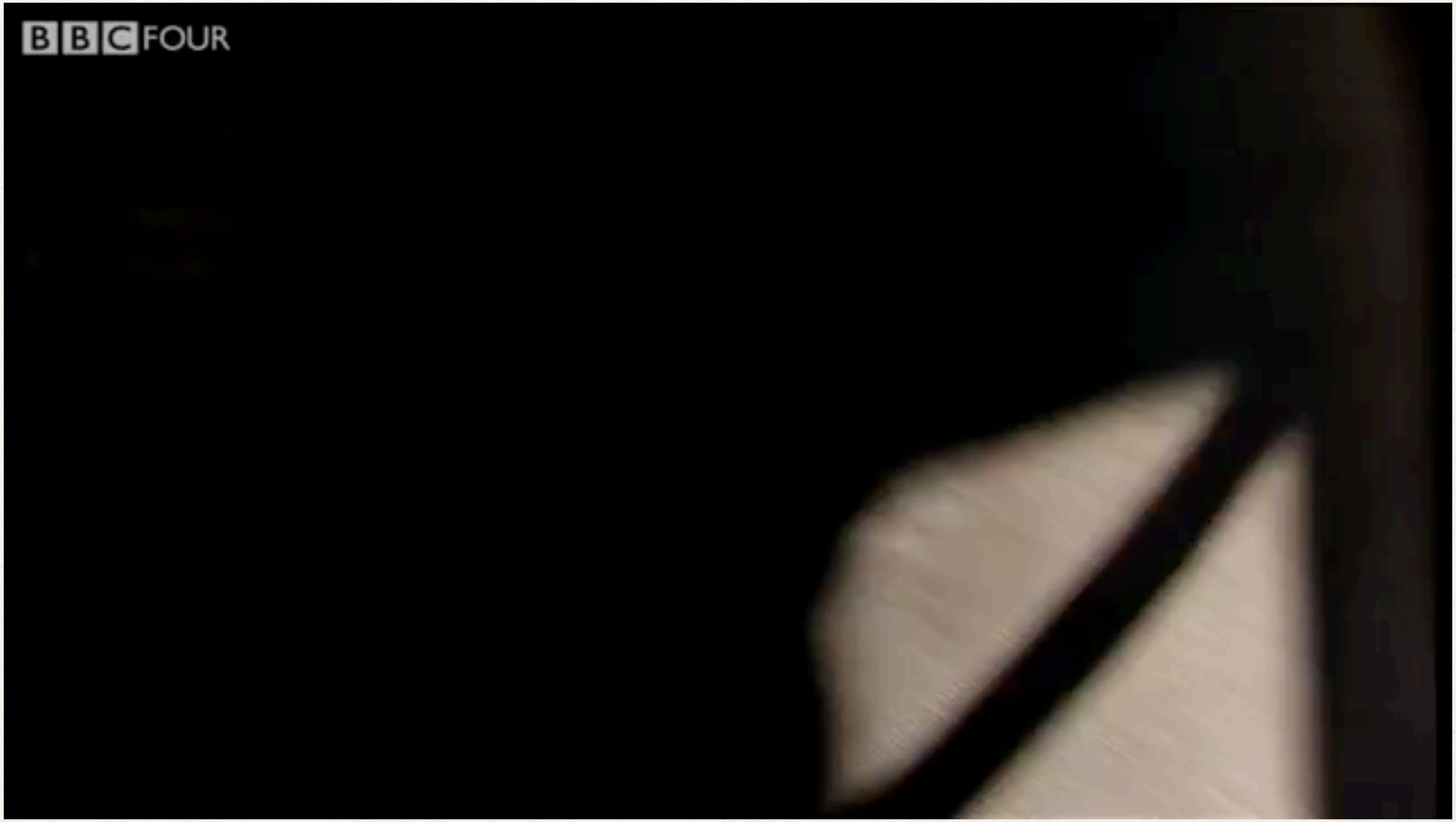
Courses

- kaggle.com is a place to do data science projects, it is your **ULTIMATE** source of knowledge in DataScience, ML, DL and AI
- fast.ai provides cutting edge about deep learning
- Google TensorFlow Development Summit new ideas and practical implication of TF
- Machine Learning A-Z: Hands on Python & R In Data Science covers machine learning workflows
- Scala and Spark for Big Data and Machine Learning covers Big Data technology
- Building Neural Network from scratch: github and blog
- Machine Learning courses ranked by user reviews

Resources

- [How to get started with ML]
- [Choosing the right ML algorithm]
- [Colah's blog]
- [Stacking Made Easy]
- [Gradient Descend Optimization]
- [ML, Python and Math Cheat Sheets]
- [Data Science interview questions]
- [Neural Network zoo]
- [Large Scale Deep-Learning with TensorFlow]
- [Learning Machine Learning]
- [Cheat Sheet for AI, ML, NN, BigData]
- [Salary history and career path of a Data Scientist]

The Story



https://www.youtube.com/watch?feature=player_embedded&v=jbkSRLYSojo