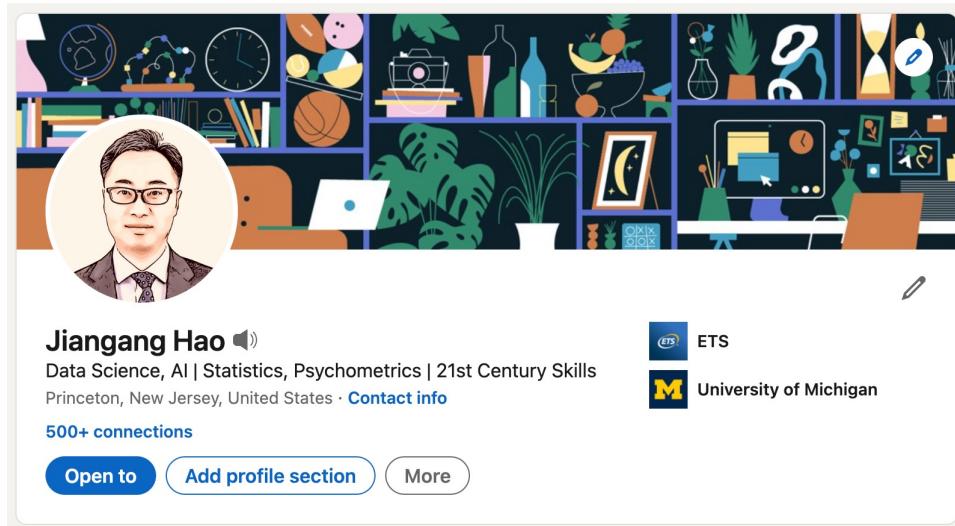




Machine Learning, Natural Language Processing, and their Application in Educational Assessment

Jiangang Hao
Psychometric and Data Science Modeling
Educational Testing Service

About me

A screenshot of a LinkedIn profile page for Jiangang Hao. The profile picture shows a man with glasses and a suit. The background is a colorful collage of various objects like books, plants, and technology. Below the profile picture, the name "Jiangang Hao" is followed by a speaker icon. The bio reads: "Data Science, AI | Statistics, Psychometrics | 21st Century Skills". It also mentions "Princeton, New Jersey, United States" and a link to "Contact info". The "Connections" section shows "500+ connections" with buttons for "Open to", "Add profile section", and "More". Logos for ETS and University of Michigan are present at the bottom right.

- Data Science
- AI/Machine Learning/NLP/Statistics
- Collaborative Problem Solving
- 21st Century Skills
- Game/simulation-based Assessments
- Astrophysics & Cosmology

<https://www.linkedin.com/in/jiangang-hao/>

Psychometric and Data Science Modeling group @ ETS



Learning Goals of this Workshop

- Learn the promises of data for digital assessments with examples
- Get a high-level understanding, with examples, of
 - Machine learning
 - Natural language processing
- Know where to find resources for further learning

This workshop will focus on Python-based tools
Lecture + demo/hands-on exercise



Promises and Challenges of Digital Learning and Assessments

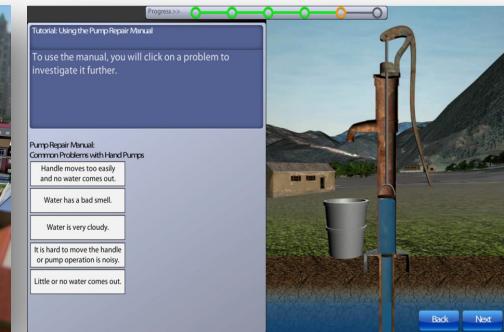
Impact of Digital Technology

- Digital technology completely changed the way of learning and assessment
 - Online learning, games, simulations, virtual reality, AI, etc.
 - Drop-in-from-sky assessment is not enough
 - Performance/competency-based assessments become feasible
- Data are the key to materialize the promises of digital technology
 - New data require new techniques to process
 - New methods are need to make sense of the complex data

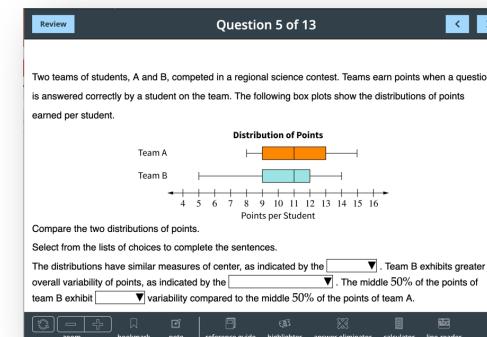
Game-based



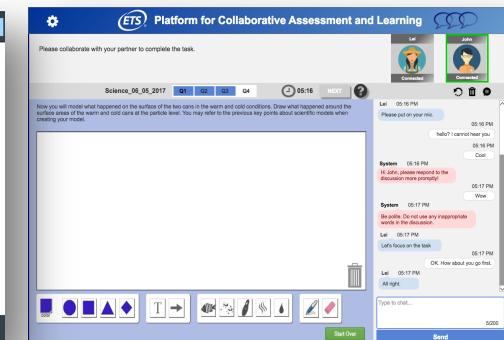
Simulation-based



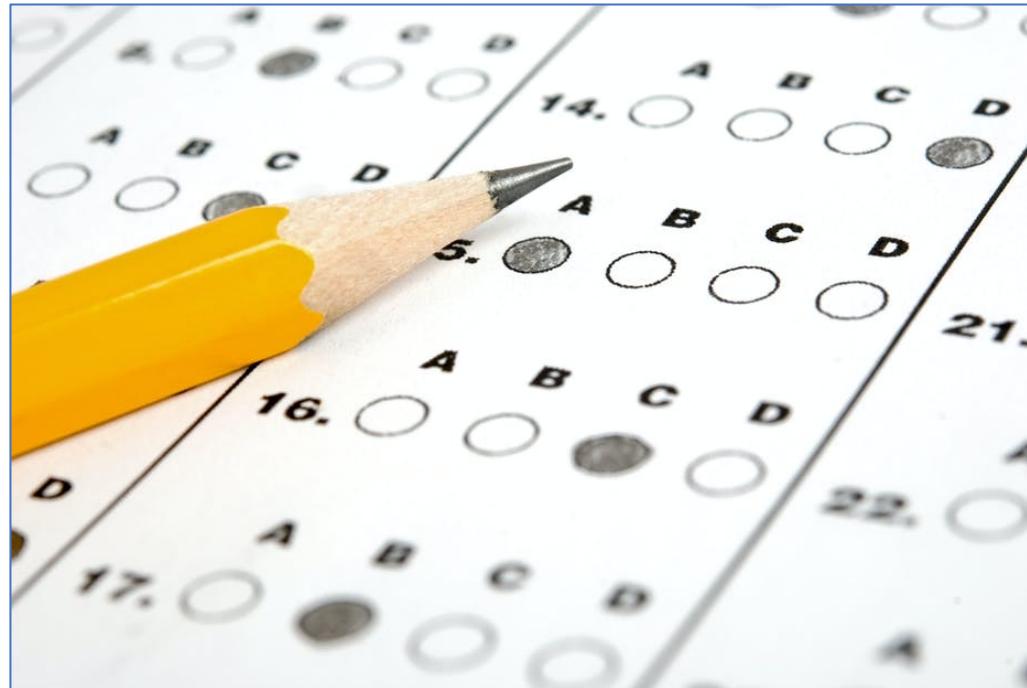
Technology-enhanced



Inter-personal



A Mindset Shift

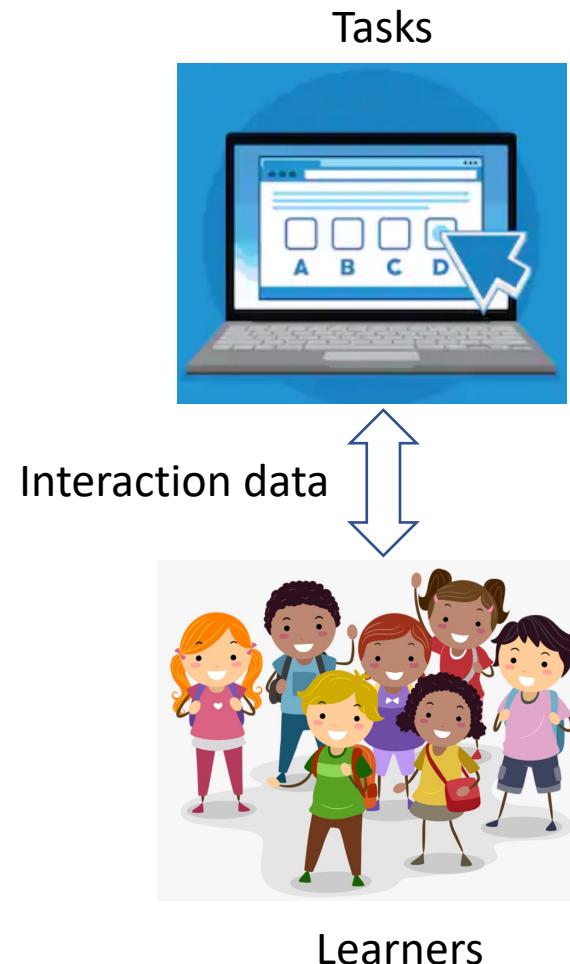


A compromise due to technology limitation of pre-digital age

Don't miss the elephant in the room



Multimodal Process Data in Assessments



Capture fine-grained interaction information between learners and learning/assessment tasks, allowing better understanding of the learners and the tasks

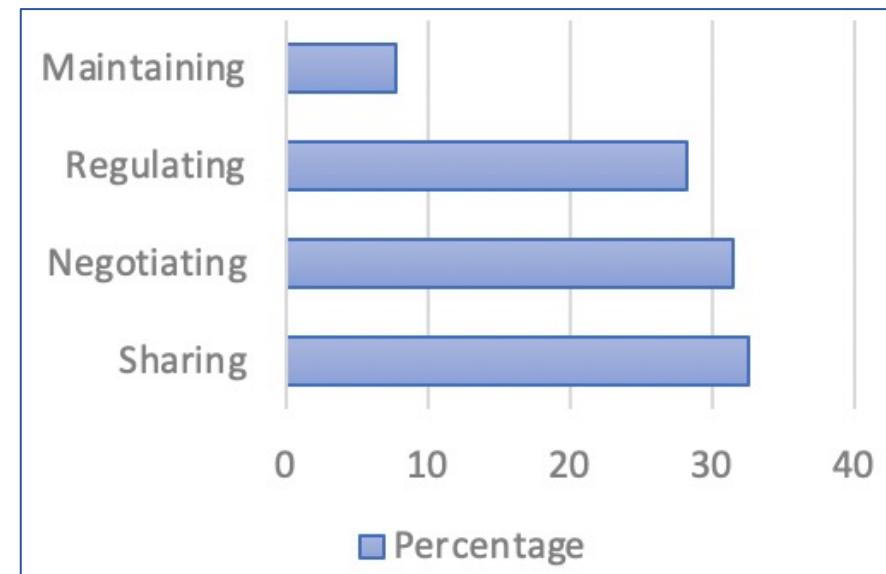
Main uses:

1. Providing evidence for new constructs
2. Providing direct support to the psychometric operations
3. Informing designs of items and delivery platforms
4. Pushing the development of new psychometric and statistical models
5. Uncover new psychology and cognitive behaviors
6. Providing information to support test security and remote testing
7. Creating more informative reporting to feedback various stakeholders.
8. Providing information on group difference and shedding new light on fairness



Use Case 1 Example: Providing Evidence for New Constructs

The screenshot shows the ETS CBAL platform interface. At the top, it says "Platform for Collaborative Assessment and Learning". Below that, there's a "Text Chat Box" section showing a conversation between users LIN and Jiangang. LIN is connected, and Jiangang is also connected. The main area displays "Question 4 of 7" about condensation. It includes a simulation of a can in a humid environment with particles around it, a temperature control slider set at 70°F, and a "play" button. The question asks about particle speed relative to the can's surface. There are three options: "The speed of the water particles decreases.", "The speed of the water particles increases.", and "The speed of the water particles remains the same." The third option is selected. Below the question, there's another part about the can being warmer than the air, with similar particle speed questions and a selected answer.

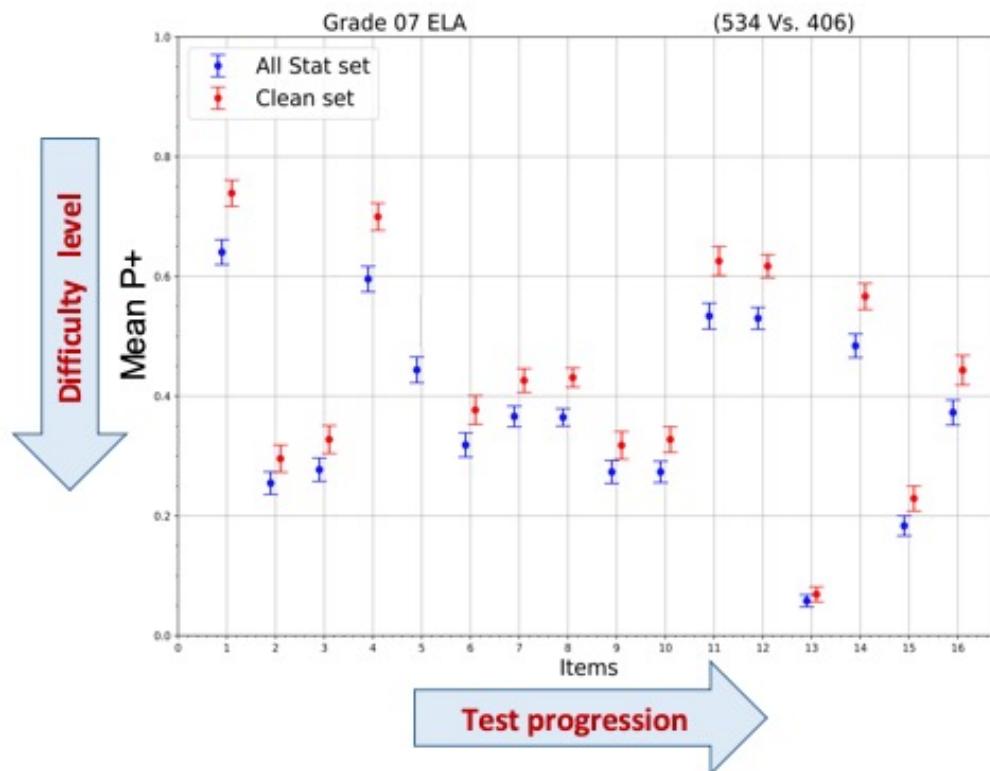


Hao, Liu, von Davier, Lederer, et al., 2017

Measuring collaborative problem solving based on the collaboration process data

Use Case 2 Example: Support Psychometric Operations

Better item calibration



Definitions:

- Clean set: students who complete the assessment in a single session
- All Stat set: students who complete the assessment in a single session and in multiple sessions

Findings

- Different samples lead to different item parameter estimates
- The difference of item mean $P+$ is **not** affected much by the item location in the test
- Easier items are more affected

Based on data from ETS Winsight assessment



Use Case 3 Example: Informing Design

- For each item type, there should be an optimal design that allows us to get more information than others.
- Considering the reading item, the design on the left allows us to know the time student spend on reading passage and responding question

NAEP

The screenshot shows a reading passage from a story about an innkeeper and a merchant. The passage is divided into two columns. The right column contains the first part of the story, and the left column contains the second part. Below the passage, there is a large text input area for responses, indicated by a blue arrow icon. The URL in the address bar is <https://cotw.naepims.org/app/student/studentShell.html#/blocks/grade4/red/1715RE4W05CLID30EX/toolbarOn/01252018>.

Winsight

The screenshot shows a reading passage titled "From Sea to Shining Sea" by Lori Mortensen. The passage describes a family's summer vacation. To the right of the passage is a list of six multiple-choice questions. The first question asks which details show the narrator is unhappy about her family's vacation. The options include statements from the passage like "But I don't even want to think about it." and "Good-bye, summer vacation. Good-bye, cannonballs at the lake. Good-bye to hanging out and doing absolutely nothing." The URL in the address bar is <https://ws.nextera.questarai.com/tds/#/practice>.

Process data provide an opportunity for us to summarize the optimal design for different item types to get the most information.



Use Case 5 Example: Uncover new psychological and cognitive behaviors

Highlighter tool

knee, the stainless steel spoon in my hand.

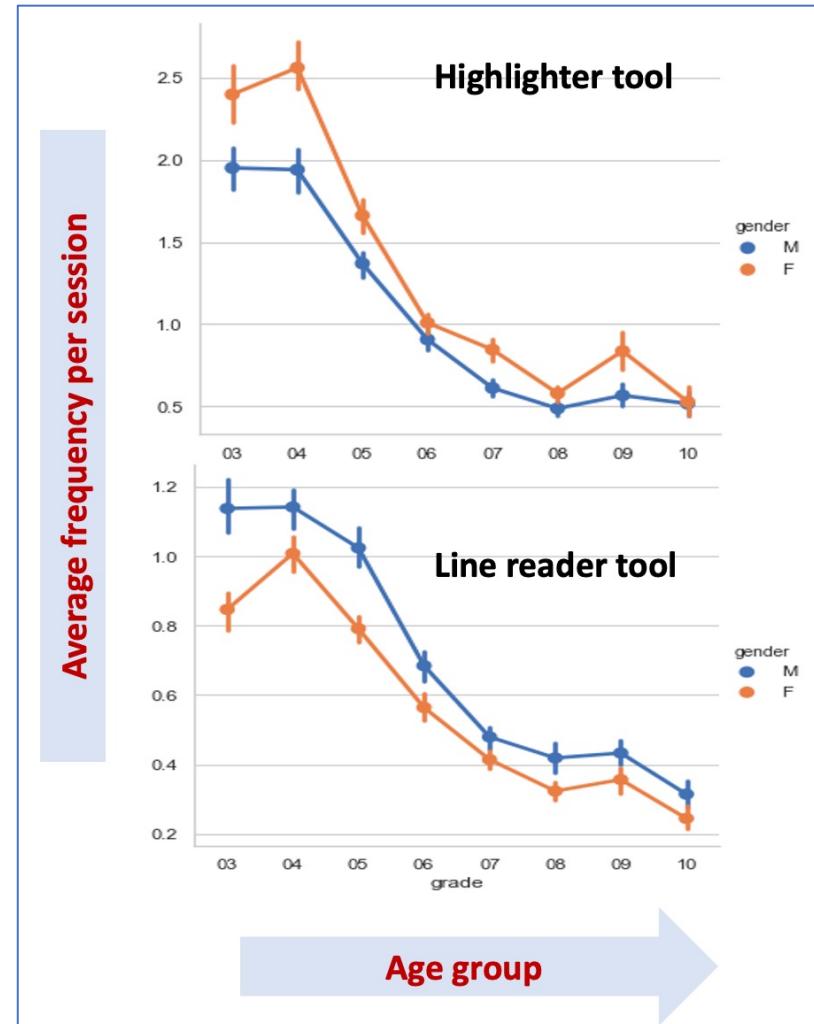
2 Violet the goat had gotten sick, suddenly and alarmingly. One day she was browsing the pasture, playing head-but with the other goats, and the next day she could barely stand. Her eyes twitched, independent of her control, in a way that was both comical and horrifying. She seemed to have seizures; when she tried to walk, her legs disobeyed her and she fell heavily and awkwardly, legs stiff. We thought she was dying. The vet said she had a thiamine deficiency and gave us a bewildering array of shots and pastes to administer to her three times a day for ten days. She hated every shot—not that I could

Line reader tool

three times a day for ten days. She hated every shot—not that I could blame her. The needles were thick and two inches long. Her shoulders must have grown sore from the repeated pokes. The yogurt was our own idea, a probiotic to encourage the essential bacteria in her gut, the original source of the missing thiamine.

3 Now, as Martin, Violet's owner, held her face between his hands, I

feeding a baby—a really big hairy smelly baby with horns. Violet—named for the dark, almost purple shade of her head—wrinkled her nose, then decided that she absolutely loved yogurt. She grabbed the



Observations:

- Girls tend to use more highlighter tool
- Boys tend to use more line reader tool
- Decreasing trend of tool usage

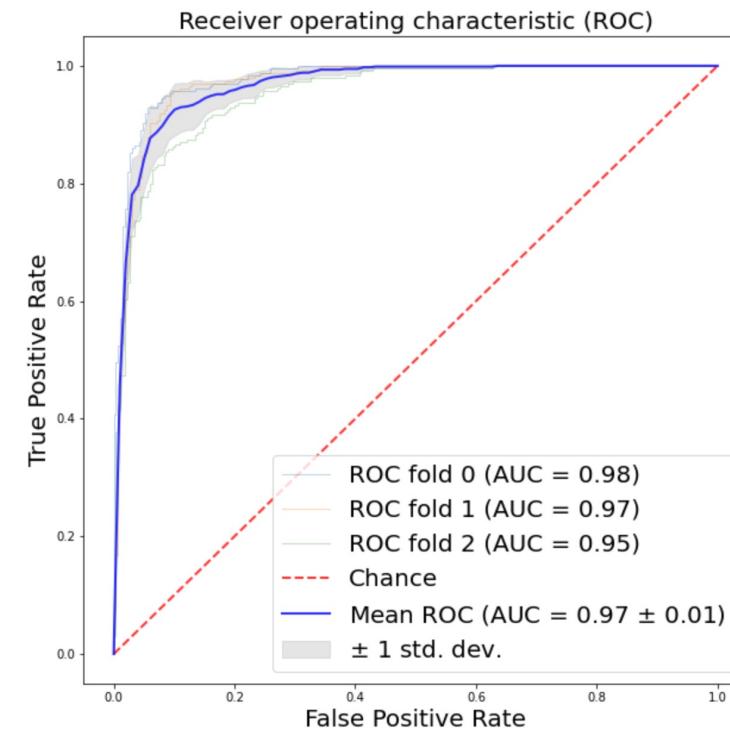
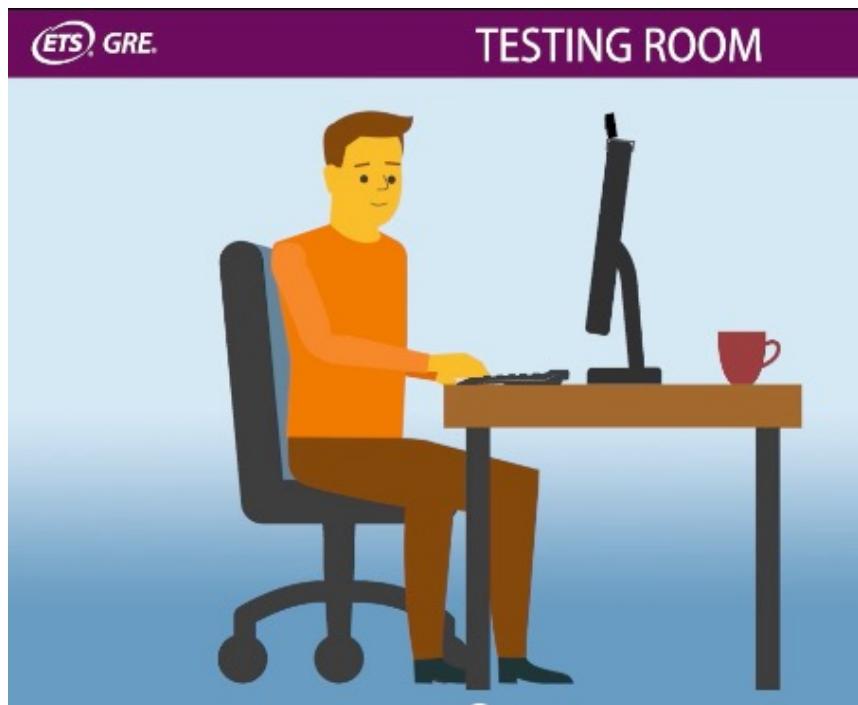
Key message:

- Analytics helps to identify subgroup difference to allow tailored strategy
- In business settings, marketing and sales strategy



Use Case 6 Example: Providing Information to Support Test Security

Detecting remote desktop access in remote testing based on clickstream data



Hao & Li, 2021



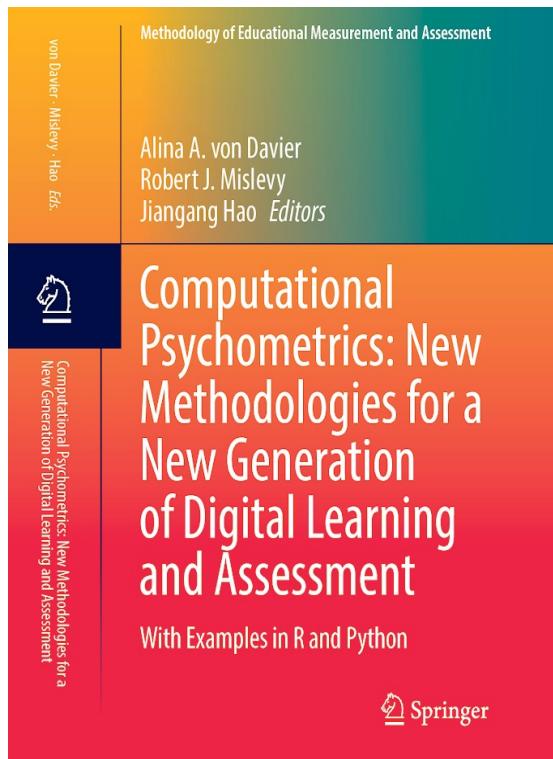
Practical Challenges



- Who is going to do the work and where are we going to find them?
- New skills are needed
 - Design systems by leveraging the new digital affordances
 - Data science
 - Machine learning
 - Natural language processing
- How much should a psychometric researcher learn?
- Where to learn these skills?

A New Book

Computational Psychometrics is a term introduced by Alina von Davier in 2015 to describe an interdisciplinary field that fuses theory-based psychometric principles and data-driven computational methods from **data science, machine learning, natural language processing, and other quantitative disciplines** to handle the large-scale and high-dimensional data from digital learning and assessment.



<https://bookauthority.org/books/new-educational-assessment-books?t=14nb4c&s=award&book=3030743934>



Coverage

- Conceptualization
 - Next generation learning and assessment
 - Computational psychometrics
 - Virtual performance-based assessment
 - Adaptive learning
- Methodology
 - Concepts and models from psychometrics
 - Bayesian inference
 - Data science
 - Machine learning
 - Time series and stochastic process
 - Social network analysis
 - NLP - text mining and automated scoring
- Code examples

https://github.com/jgbrainstorm/computational_psychometrics

1	Introduction to Computational Psychometrics: Towards a Principled Integration of Data Science and Machine Learning Techniques into Psychometrics	1
	Alina A. von Davier, Robert J. Mislevy, and Jiangang Hao	
Part I Conceptualization		
2	Next Generation Learning and Assessment: What, Why and How ...	9
	Robert J. Mislevy	
3	Computational Psychometrics: A Framework for Estimating Learners' Knowledge, Skills and Abilities from Learning and Assessments Systems	25
	Alina A. von Davier, Kristen DiCerbo, and Josine Verhagen	
4	Virtual Performance-Based Assessments	45
	Jessica Andrews-Todd, Robert J. Mislevy, Michelle LaMar, and Sebastiaan de Clerk	
5	Knowledge Inference Models Used in Adaptive Learning	61
	Maria Ofelia Z. San Pedro and Ryan S. Baker	
Part II Methodology		
6	Concepts and Models from Psychometrics	81
	Robert J. Mislevy and Maria Bolsinova	
7	Bayesian Inference in Large-Scale Computational Psychometrics ...	109
	Gunter Maris, Timo Bechger, and Maarten Marsman	
8	A Data Science Perspective on Computational Psychometrics	133
	Jiangang Hao and Robert J. Mislevy	
9	Supervised Machine Learning	159
	Jiangang Hao	
10	Unsupervised Machine Learning	173
	Pak Chung Wong	
11	Advances in AI and Machine Learning for Education Research.....	195
	Yuchi Huang and Saad M. Khan	
12	Time Series and Stochastic Processes	209
	Peter Halpin, Lu Ou, and Michelle LaMar	
13	Social Networks Analysis	231
	Mengxiao Zhu	
14	Text Mining and Automated Scoring	245
	Michael Flor and Jiangang Hao	



Why We Need It?

- Intrinsic need to expand the existing psychometric methodologies to include new methods from, e.g., data science, machine learning, natural language processing and other quantitative disciplines, to address the new challenges of learning and assessment in the digital age. When many new methods are included, introducing a new term to encompass these new features will be more convenient and effective for communication.
- Practical challenges of preparing the workforce.
 - Applicants from psychometrics programs do not have the needed data science/machine learning skills (and mindsets) to process and model complex data from digital tasks
 - Applicants with data science/machine learning skills from other disciplines, such as computer science, generally know very little about the core values of psychometrics.
 - Hiring people who do not know the core values of the substantive area poses a big retention challenge for organizations, as they may quickly move on if they find they are not interested in the area at all after a few months.
 - It is imperative to prioritize a set of new methodologies and integrate them with the core values of psychometrics in a principled manner to help prepare a stable workforce for digital learning and assessment in the future.
- Bridge people from other quantitative disciplines (such as computer science, applied mathematics, physics, and others) to digital learning and assessment.
 - Providing a concise coverage of psychometrics' established values and methods could help them better understand how to apply their skills to join forces to promote learning and assessment in a digital age.



Computational Psychometrics in Measurement

Bringing
Together the
Different
Aspects of
Measurement

1. Testing Specialists
2. Computational Psychometricians
3. Classroom Assessment Specialists
4. Critical Theorists & Philosophers of Science



Briggs, NCME Presidential Address, 2022



Prioritized Areas



- Machine learning/AI
 - Supervised and unsupervised learning
 - Some use cases
 - Software packages
- Natural Language Processing
 - Language models
 - Text representation and mining
 - Automated scoring
 - Deep learning-based models

Workshop

- Data science
 - Data science basics
 - Data wrangling and processing
 - Visualization and dashboarding

Homework



Machine Learning

- Machine learning basics
- Supervised learning
- Unsupervised learning
- Deep learning
- Software tools and packages

Chapter 9 Supervised Machine Learning

Jiangang Hao



Abstract Machine learning refers to a set of methodologies that allow computers to “learn” the relationship among numerical representations of data. In this Chapter, we focus on an important branch of machine learning, supervised machine learning, and introduce three widely used supervised learning methods, the Support Vector Machine, Random forest, and Gradient Boosting Machine. Python codes examples are included to show how to use these methods in practice.

9.1 Introduction

As highlighted in Chap. 6, one of the core missions of psychometrics is to ensure the observed evidence from assessment or learning tasks can support the claims on the targeted constructs in a valid, reliable, fair and comparable way. In the area of educational assessment, traditional psychometrics has given most attention to developing systematic methodologies to accomplish this mission when the evidence can be easily mapped into some simple forms, such as dichotomous or polymotous scores. The regularity and simplicity of the data make statistical inference well suited for modeling the data. However, digitally based learning and assessment tasks generate much more complex data. The complexity of these data makes it difficult (or sometimes impossible) to represent the information in simple scores in order to apply well-established statistical modeling frameworks, such as the familiar Item Response Theory (IRT). So, there is an intrinsic need for additional methodologies to harness the more complex data from digitally based

The R or Python codes can be found at the GitHub repository of this book: https://github.com/jgbainstorm/computational_psychometrics

J. Hao (✉)
Educational Testing Service, Princeton, NJ, USA
e-mail: jhao@ets.org

Chapter 10 Unsupervised Machine Learning

Pak Chung Wong



Abstract The chapter introduces the concept of machine learning with an emphasis on unsupervised learning algorithms and applications. The discussion starts with a brief background on machine learning and then a high-level discussion on the differences between supervised and unsupervised learning algorithms. We present three categories of unsupervised machine learning techniques that include clustering, outlier detection, and dimension reduction; five prevailing unsupervised learning algorithms that include K-means, agglomerative clustering, DBSCAN, principal component analysis, and multidimensional scaling; and five Python programming examples that demonstrate the learning concepts and results using psychometric assessment data collected from an online collaborative problem-solving environment. This chapter demonstrates the potential of machine learning and highlights the opportunities it presents in psychometric research and development.

10.1 Introduction

Arthur Lee Samuel (Samuel, 1959), who coined the term “machine learning,” defined the then new discipline as a sub-field of Computer Science that gives “computers the ability to learn without being explicitly programmed.” Machine learning (Machine Learning, 2017) has since found its way into different data science communities, which include, among others, exploratory data analysis (Tukey, 1997), data mining (Fayyad et al., 1996), and visual analytics (Wong & Thomas, 2004). More recently, (Pedro Domingos, 2015) broke down the five tribes (or paradigms) of machine learning as symbolists, connectionists, evolutionaries,

The R or Python codes can be found at the GitHub repository of this book: https://github.com/jgbainstorm/computational_psychometrics

P. C. Wong (✉)
University of Iowa, Iowa City, IA, USA

Chapter 11 Advances in AI and Machine Learning for Education Research

Yuchi Huang and Saad M. Khan

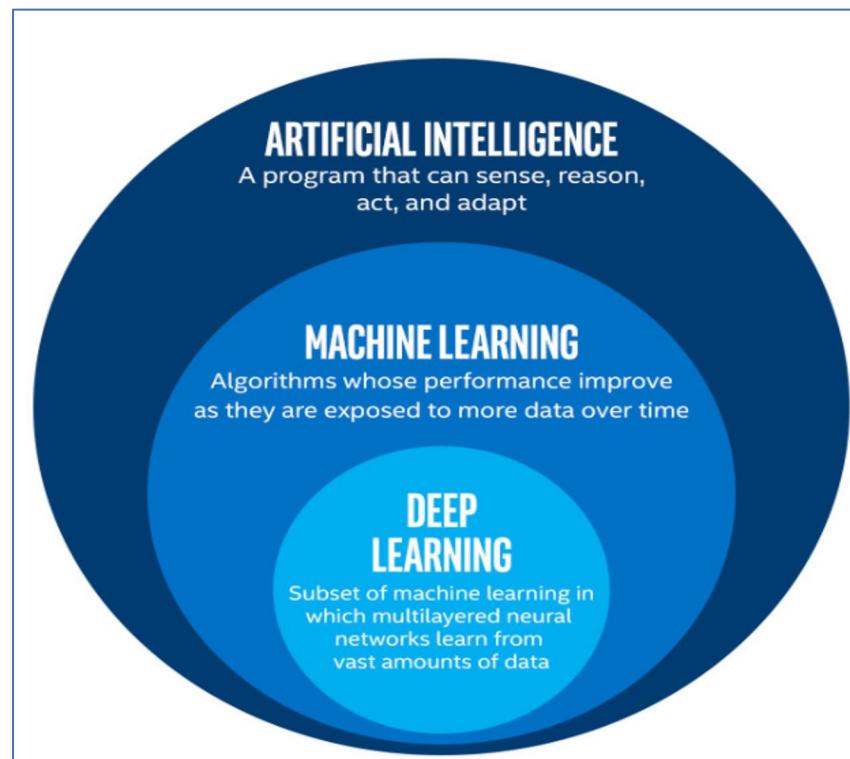


Machine Learning Basics



Machine Learning

A term coined by Arthur Samuel in 1959, refers to a set of methodologies that allow computers to “learn” the relationship among numerical representations of data without explicit instructions by human experts



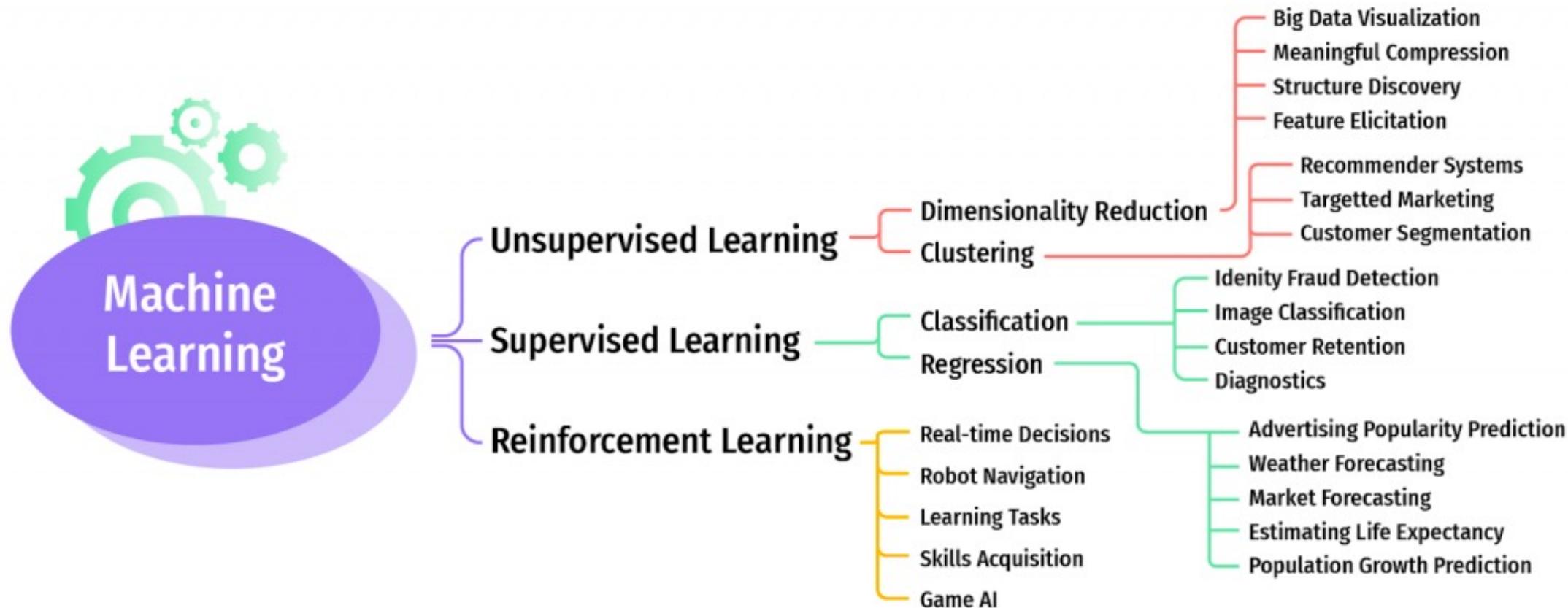
Mat Velloso
@matvelloso

Difference between machine learning and AI:
If it is written in Python, it's probably machine learning
If it is written in PowerPoint, it's probably AI

22/11/18, 5:25 PM

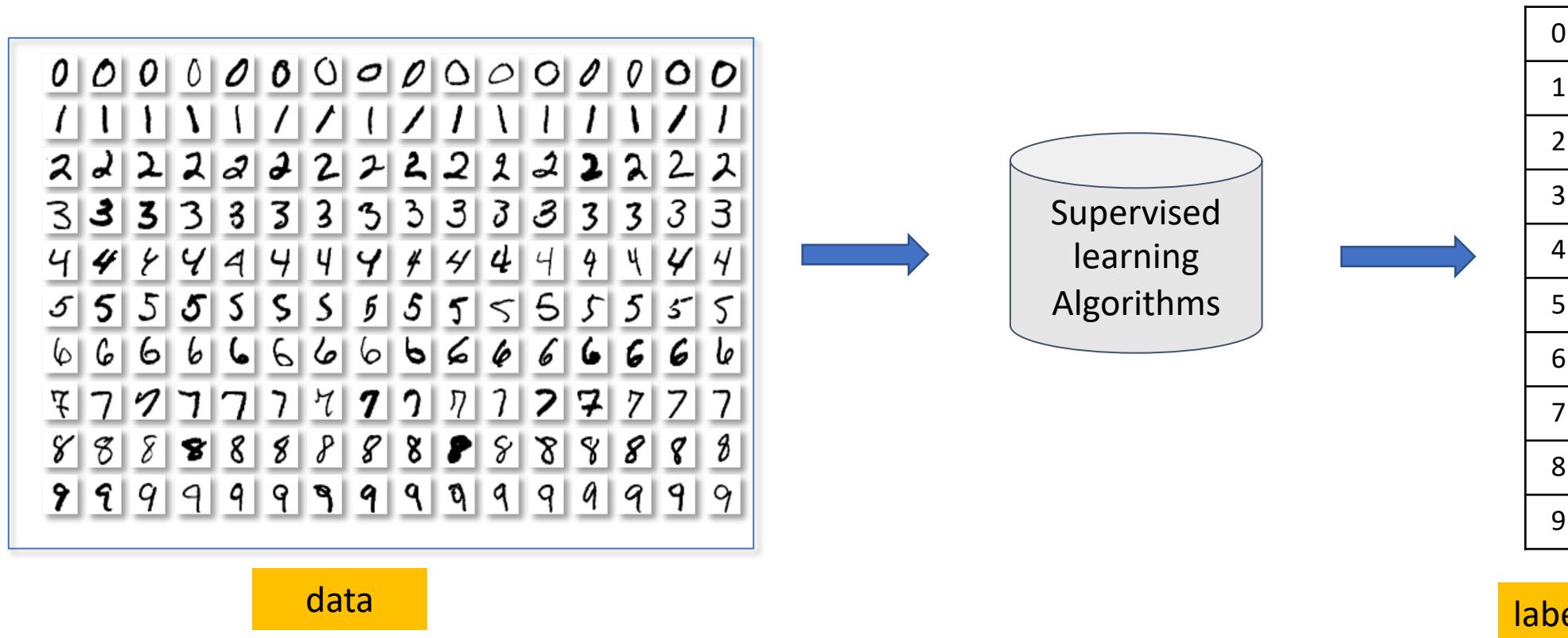
3,514 Retweets 10.8K Likes

Main Paradigms of Machine Learning



Supervised learning

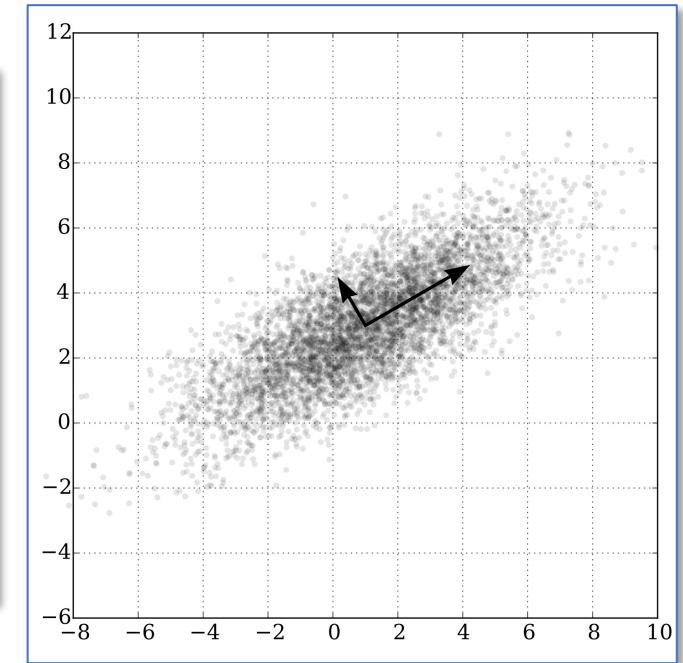
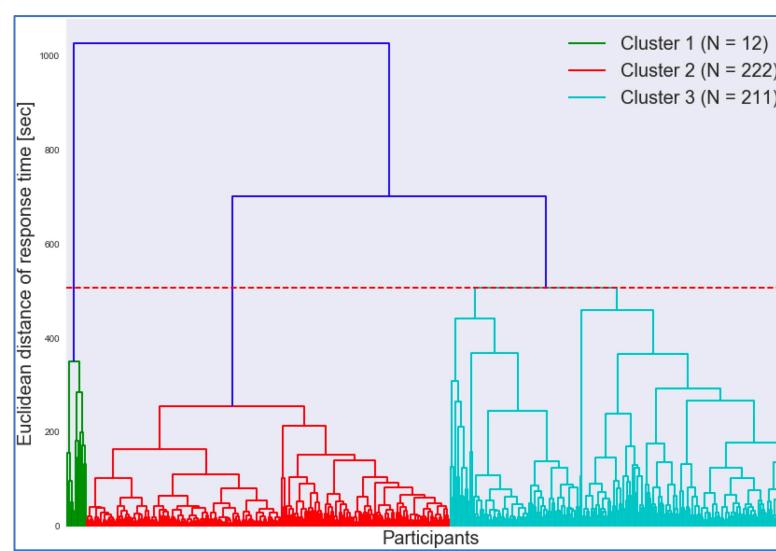
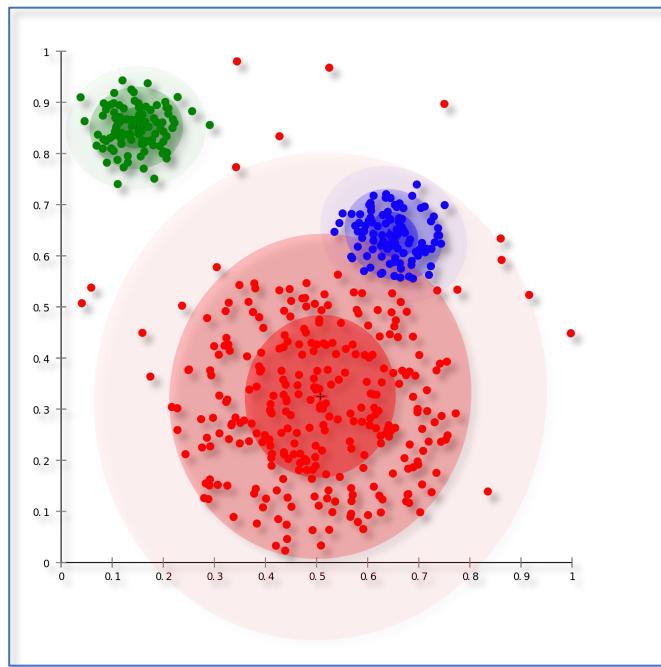
Machine learns the relationship (mapping) between
independent variables (features) and **dependent** variables (labels)



Unsupervised learning

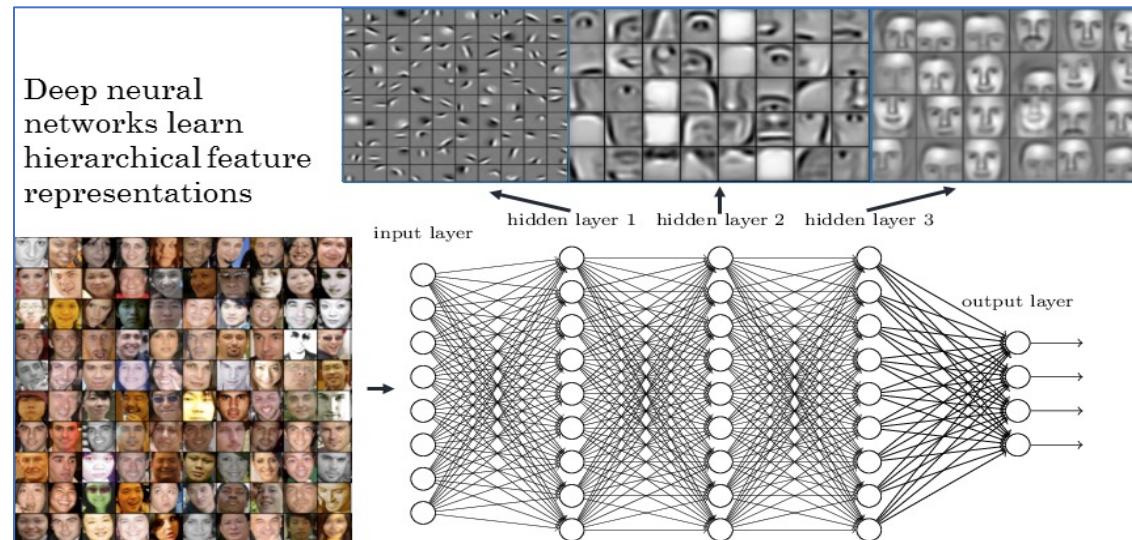
Methodology that discovers concentrations, associations, or correlations in data.

- Cluster analysis
- Dimensionality reduction



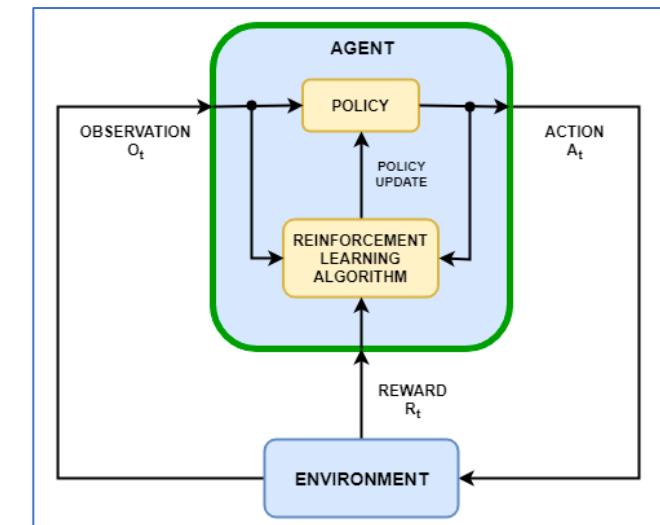
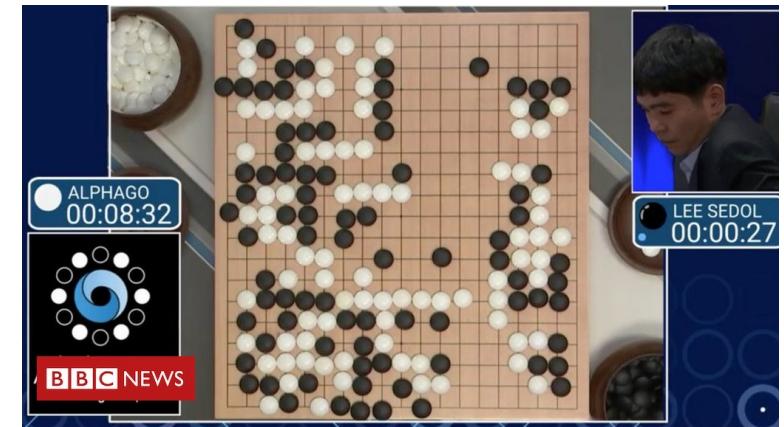
Representation Learning

- In supervised learning, feature representation is a crucial step
- Features design
 - By human experts – time consuming and expensive
 - By algorithms - representation learning
- End-to-end learning



Reinforcement Learning

- Machine learns optimal strategies by means of a goal-oriented exploration of a parameter or state-space to optimize a reward function (Sutton, & Barto, 1998).
- Key components
 - Agent
 - States
 - Hidden
 - Observable (observation)
 - Actions: what the agent can do
 - Policy/strategy: tell what the agent should do
 - Reward: what the agent gets after an action



From MathWorks

Supervised Machine Learning



Question 1

- How supervised learning is different from statistical inference, as they both aim at mapping the relationship between the independent and dependent variables?
 - I. Machine learning is more prediction-driven while statistical inference emphasizes both prediction and model parameter estimation.
 - II. Statistical inference emphasizes developing a probabilistic model to characterize the data and then estimating the model parameters based on some (usually well-studied) probability distribution functions, while machine learning emphasizes computation algorithms that can efficiently carry out the inference process by minimizing certain loss functions that do not necessarily have a probabilistic underpinning.
 - III. Statistical inference usually deals with data with a small number of variables obtained through planned *experiments* or quasi-experimental comparisons while machine learning handles sparse and high-dimensional data with a large number of variables (features), usually obtained through passive and uncontrolled *observations* (RWE).

Hao & Ho, 2019; Hao, 2021



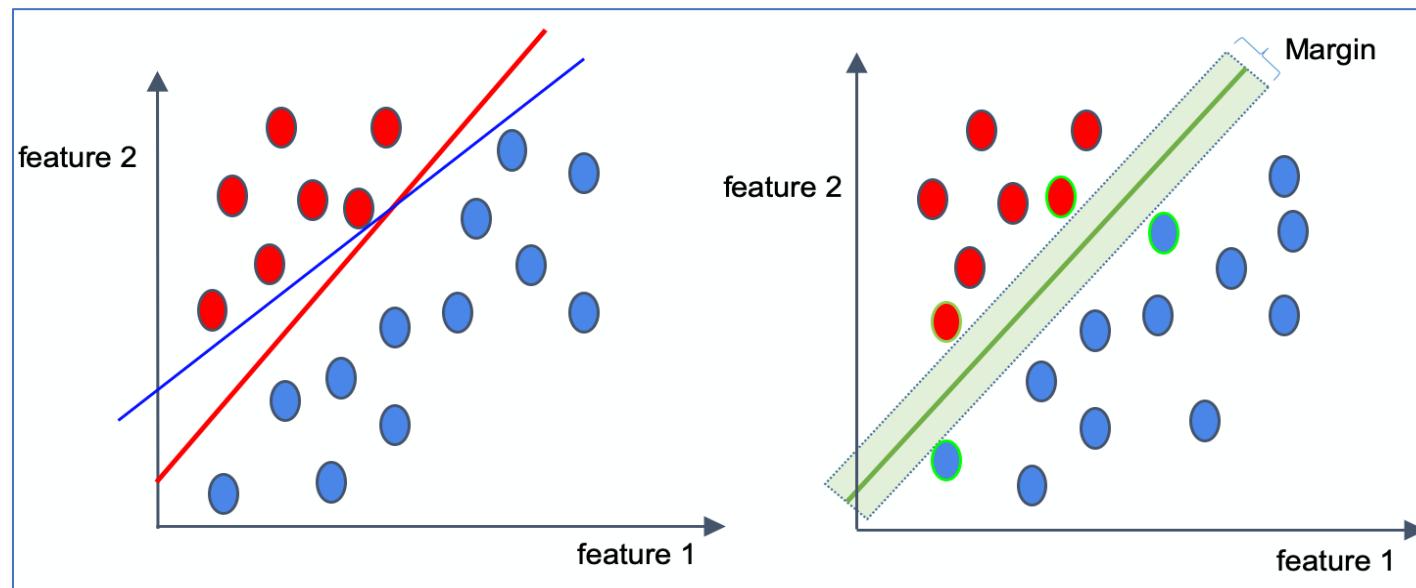
Question 2

- Which, if any, supervised learning methods are consistently superior to the others?
 - It is generally data and problem dependent
 - Some methods work well based on many explorations
 - SVM, Random Forest, XGBoost (Gradient Boosting Machine)
 - Deep Neural Network



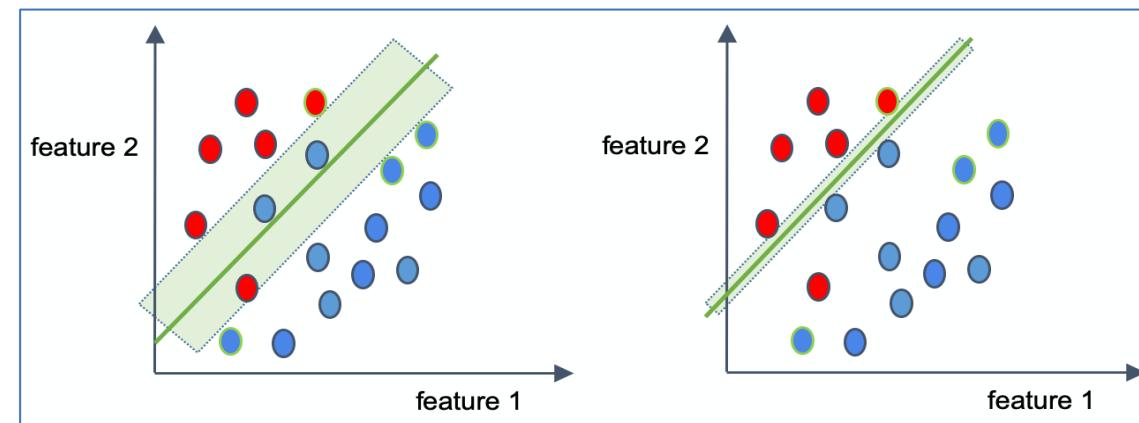
Support Vector Machine

It aims at finding a decision hyperplane in the feature space so that data points can be separated into different categories with a maximum margin that is defined as the distance of the closest points (support vectors) from each category to the decision hyperplane (Vapnik, 1963).



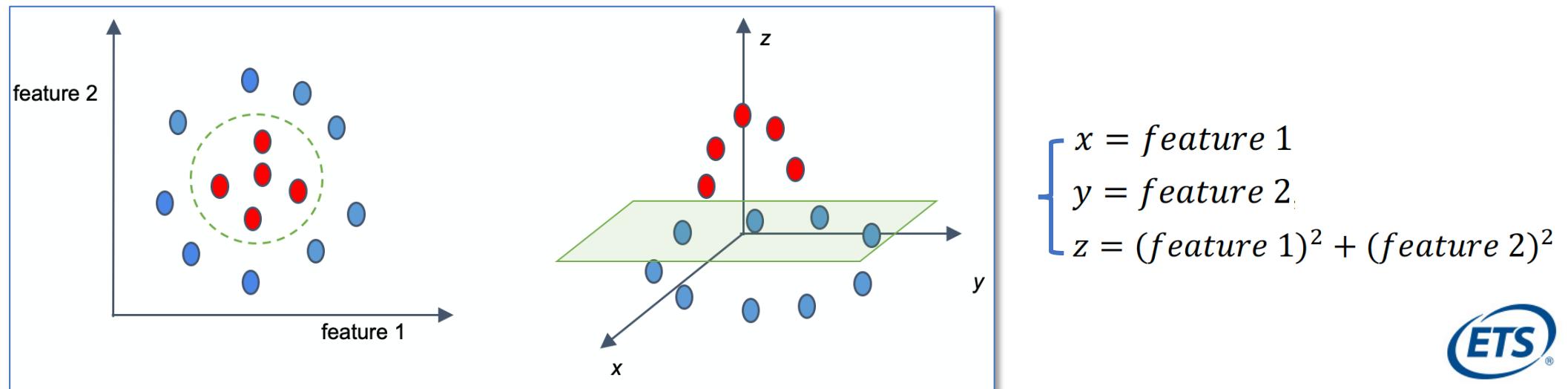
Non-linearly Separable Case

- Choose between a decision hyperplane that has a larger margin but more misclassified data points, and the one with a narrower margin but fewer misclassified data points, as illustrated in the left and right panels
- A hyperplane with a narrower margin may lead to better classification accuracy based on the training data, but is more susceptible to overfitting, while a hyperplane with a larger margin has lower classification accuracy with respect to the training data, but it is more robust against overfitting.
- In SVM algorithm, the regularization hyper-parameter (C), is used to control this balance. Choosing a larger C indicates that one favors a narrower margin with a lower tolerance of misclassified data points



Kernel Trick

- The transformation of the original feature space into a higher dimensional space to make the data linearly separable is done through a procedure called kernel trick.
- The characteristic of data being more likely to be linearly separable when being projected into a higher dimensional space via some non-linear transformation is known as Cover's theorem (Cover, 1965).
- In SVM algorithms, one can choose the kernel type – another hyperparameter. Typical kernels are polynomial and RBF.



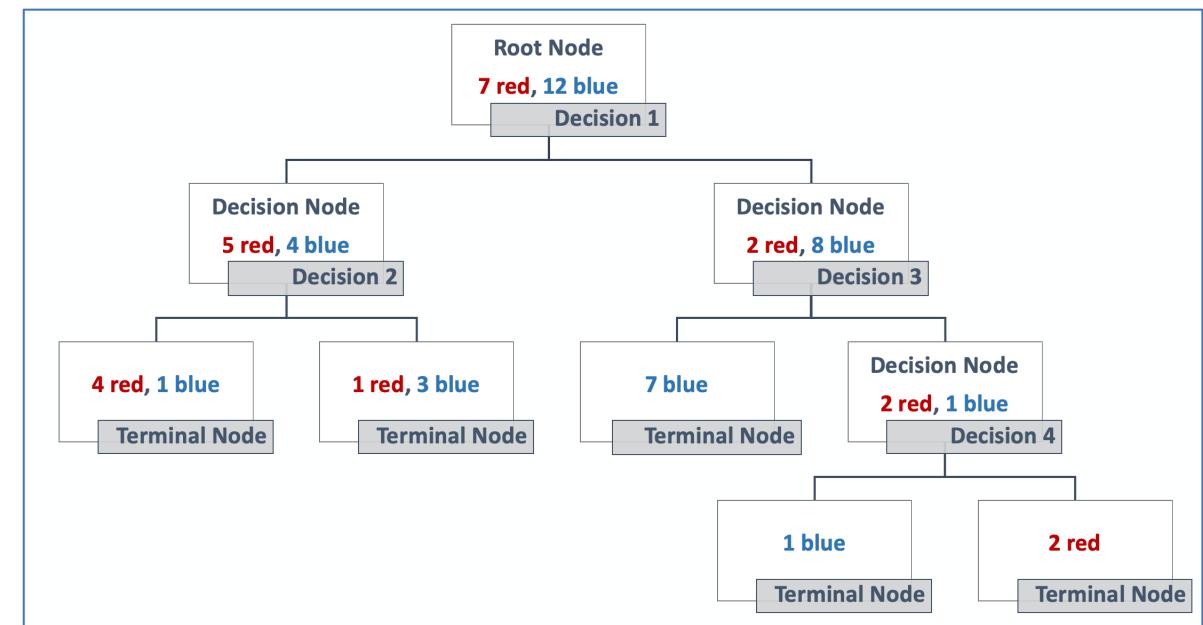
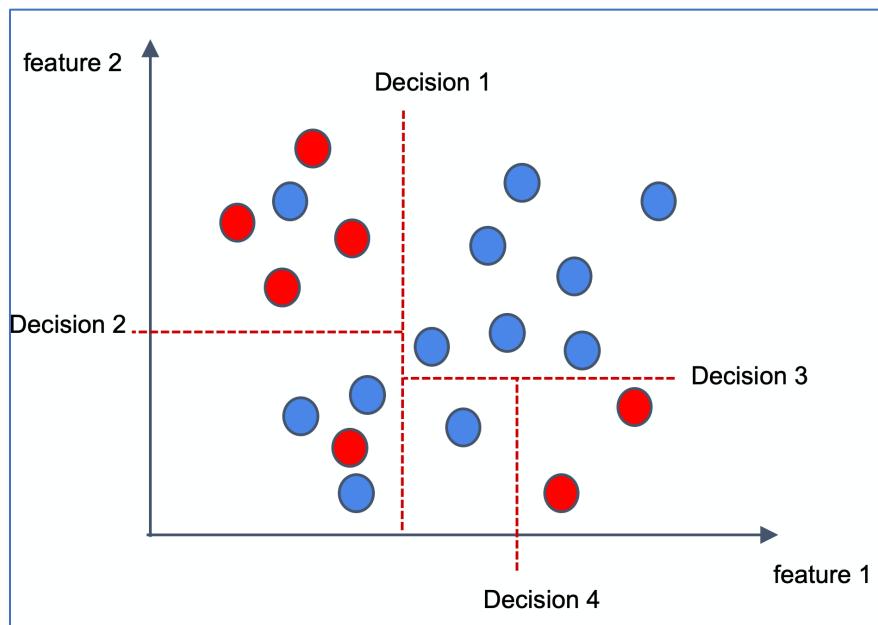
Ensemble Learning

- Given there are many supervised learning methods, it is always tantalizing to think whether some sort of “average” from an ensemble of learners will lead to better predictive performance. Research studies in this direction leads to ensemble learning (Dietterich, 2000).
- Two leading ideas regarding how an ensemble of learners is constructed:
 - Bagging (bootstrap aggregating) (Breiman, 1996)
 - The ensemble is created through bootstrapping the dataset (Efron & Tibshirani, 1994)
 - The outputs from the learners from each sample will be combined (average or voting) to improve the predictive performance.
 - Example: Random Forest (Ho, 1995; Breiman, 2001)
 - Boosting (Friedman, 2001)
 - The ensemble is constructed sequentially, rather than in parallel, as in the bagging approach.
 - A learner is first trained on the full dataset and subsequent learners are added by fitting the residuals (after applying all preceding learners), by which new learners will assign more weights to the poorly predicted observations by the preceding learners.
 - Example: Gradient Boosting Machine



Decision Tree

A decision tree works by forming decision rules on the features recursively to minimize some loss function of the classification.



Decision Tree Cont'd

- Loss function
 - Gini index as used in the CART algorithm (Breiman et al., 1984).
 - Information Gain as used in the Iterative Dichotomiser 3 algorithm (Quinlan, 1986).
- Stopping rule
 - Condition under which no further splitting will happen.
 - E.g., minimum leaf size
- Pruning
 - A procedure to remove the sub-nodes from decision nodes performed after the training to avoid overfitting.
- Main drawback:
 - Instability of the results, as a small change in a dataset may lead to quite different decision rules.



Random Forest

- Suppose we have N observations and M features in a dataset, then a typical random forest classifier goes as follows:
 1. Create K samples of the dataset using the bootstrapping technique, and each sample has N observations
 2. For each of the K samples, a decision tree is applied. In each of the decision nodes of each of the trees, a decision is made based on the most discriminative one of a randomly chosen F features from the total M features – random subspace or feature bagging (Ho, 1998).
 3. The average/voting of multiple trees will be the final estimator
- Hyper-parameters
 - The number of trees (K)
 - The size of the feature subspace (F)



Gradient Boosting Machine

- Assumption: the ideal mapping between the dependent and independent variables could be approximated by the sum of many functions belonging to the same parametric family.
- The parameters are fixed iteratively as following
 1. Fit the data (\mathbf{X}, y) with a base tree $h_0(\mathbf{X}; \boldsymbol{\alpha}_0)$
 2. Add another tree $h_1(\mathbf{X}; \boldsymbol{\alpha}_1)$ by fitting the residuals $(\mathbf{X}, y - h_0(\mathbf{X}; \boldsymbol{\alpha}_0))$.
 3. Repeat the procedure leads to $F(\mathbf{X}) = \sum_{m=0}^M \beta_m h_m(\mathbf{X}; \boldsymbol{\alpha}_m)$
- Where is the “Gradient” from?
 - For loss function: $L(y, F(\mathbf{X})) = \frac{1}{2}[y - F(\mathbf{X})]^2$
 - The Residual: $y_i - F_m(\mathbf{X}_i) = -\left. \frac{\partial L(y_i, F(\mathbf{X}_i))}{\partial F(\mathbf{X}_i)} \right|_{F(\mathbf{X})=F_{m-1}(\mathbf{X})} \equiv -g_m(\mathbf{X}_i)$



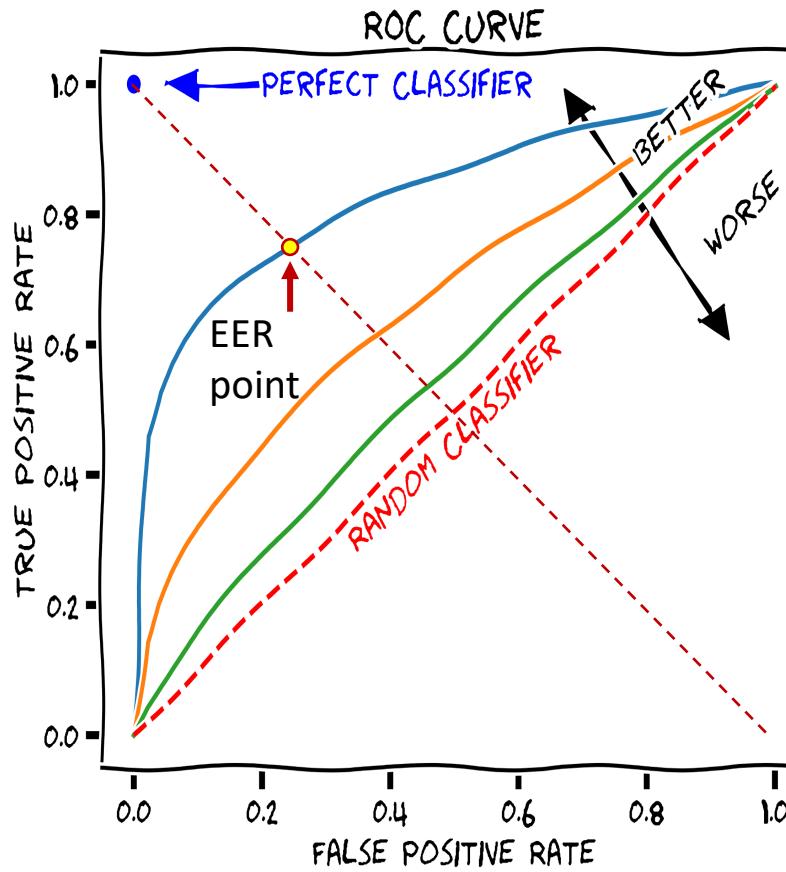
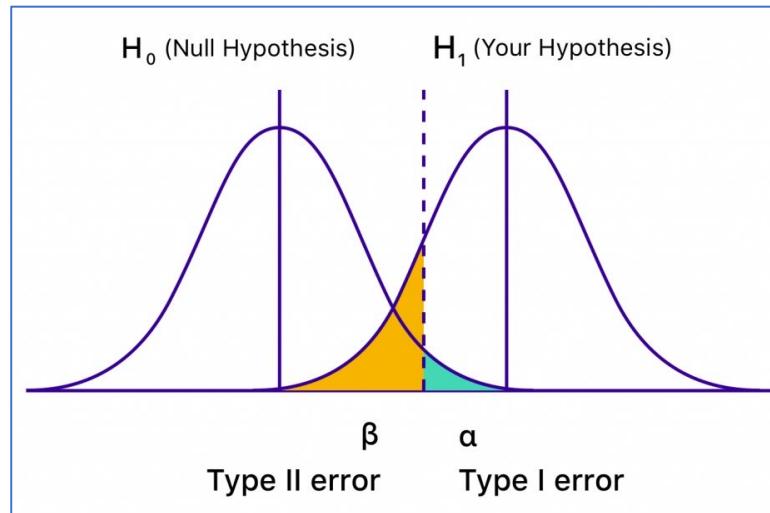
Evaluation Metrics for Binary Classification

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$ True negative rate
		Precision $\frac{TP}{(TP + FP)}$ Positive Predicted value	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Error Rate = $(FP+FN)/(TP+TN+FP+FN)$
 False positive rate = $FP/(FP+TN)$
 F-Score(Harmonic mean of precision and recall) = $(1+b)(PREC.REC)/(b^2PREC+REC)$ where b is commonly 0.5, 1, 2.

https://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers

Receiver Operating Characteristic - ROC

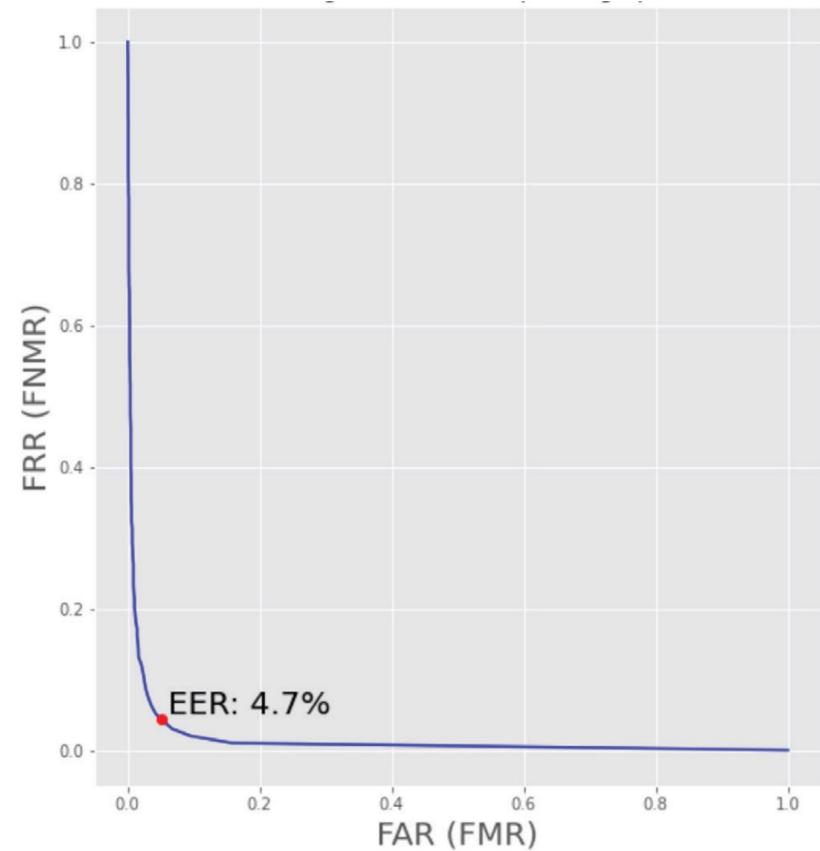


- Area under curve (AUC):
 - 0.5: non discriminative
 - 0.7 – 0.8: acceptable
 - 0.8 – 0.9: excellent
 - > 0.9: outstanding
- Equal Error Rate
 - Fingerprint: 0.2%
 - Signature: 2%
 - Voice: 2%
 - Keystroke: 4.7%
 - TouchID: 0.002%
 - FaceID: 0.001%

Example Application – Keystroke Biometrics

- Writing process data – keystrokes
- Feature representation of keystrokes
- Mapping between keystroke features and labels (same person or not)

Feature name	Definition	Within-person correlation
inword_logIKI_median*	Median duration of in-word keystrokes, measured in log milliseconds	0.95
inword_logIKI_mean	Mean duration of in-word keystrokes in log milliseconds	0.95
wordinitial_logIKI_median*	Median duration of word-initial keystrokes in log milliseconds	0.92
append_interword_interval_logIKIs_mean	The mean log interkey intervals for keystrokes that add white space between words	0.92
wordinitial_logIKI_mean	Mean duration of word-initial keystrokes in log milliseconds	0.92
append_interword_interval_logIKIs_median*	The median log interkey interval for keystrokes that add white space between words	0.91
append_interword_interval_speed_median*	The speed of keystrokes that add white space between words, measured in characters per second	0.91
wordinitial_char_per_sec_median*	Median speed of typing the first character of a word, in characters per second	0.91
iki400_AppendBurst_len_mean	Mean length in characters of bursts of append keystrokes where no pause is greater than 400 milliseconds	0.91
iki400_AllActionBurst_len_mean	Mean length in characters of bursts where all keystrokes count as part of the burst, and bursts end on pauses longer than 400 milliseconds	0.90
initial_backspace_char_per_sec_median*	The median speed of the first in a series of backspace actions, measured in characters per second	0.90
iki200_AppendBurst_len_mean	Mean length in characters of bursts of append keystrokes where no pause is longer than 200 milliseconds	0.90
initial_backspace_logIKI_median*	The median log interkey interval for backspace actions that appear first in a series of backspace actions	0.89



(Choi, Hao, Deane & Zhang, 2021)



Unsupervised Machine Learning



Cluster Analysis

- Group similar things together
- What is “similar”
 - Similarity is measured by certain distance in a metric space

A **metric space** is an ordered pair (M, d) where M is a set and d is a **metric** on M , i.e., a function

$$d: M \times M \rightarrow \mathbb{R}$$

such that for any $x, y, z \in M$, the following holds:^[2]

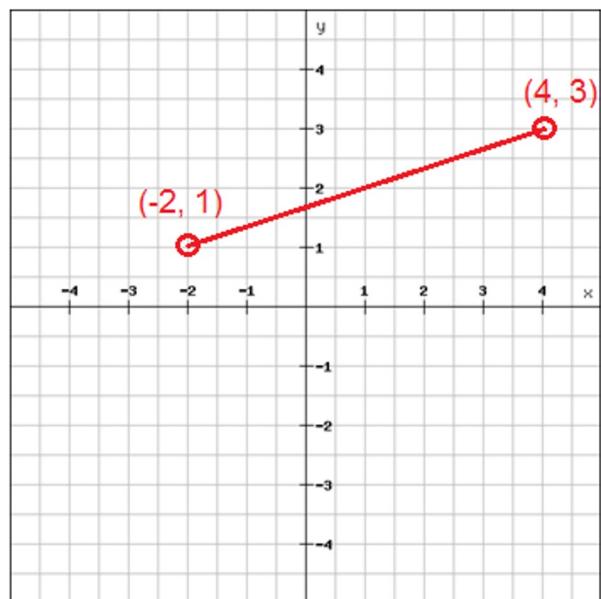
1. $d(x, y) = 0 \Leftrightarrow x = y$ identity of indiscernibles
2. $d(x, y) = d(y, x)$ symmetry
3. $d(x, z) \leq d(x, y) + d(y, z)$ subadditivity or triangle inequality

- Clusters are not unique, depending on the ways you look at or interpret the data

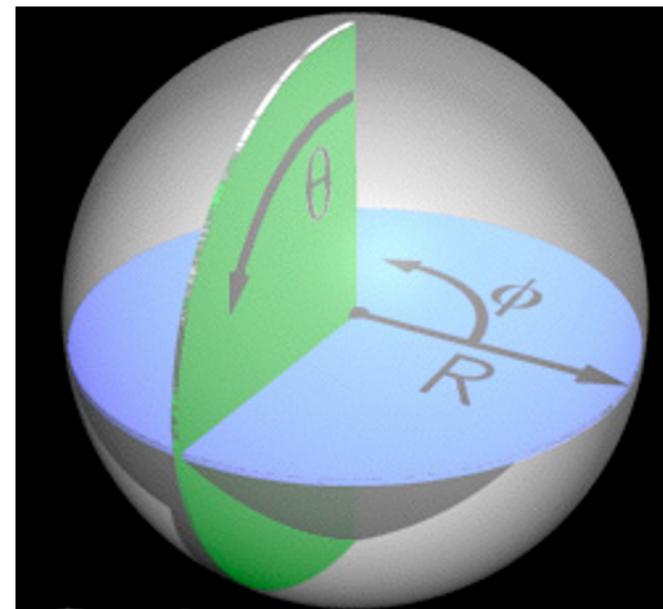


Example Distance

- Euclidean Space



- Non-Euclidean Space



$$ds^2 = dx^2 + dy^2$$

$$ds^2 = R^2 d\theta^2 + R^2 \sin^2 \theta d\phi^2$$

Popular Distances in ML

- Euclidean distance
- Minkowski distance $D(X, Y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$
- Mahalanobis distance $d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})}$
- Cosine distance $1 - \frac{u \cdot v}{\|u\|_2 \|v\|_2}$
- Edit distance: the minimum number of editing operations to bring two strings the same

*Analyzing Process Data from Game/Scenario-Based Tasks:
An Edit Distance Approach (Hao, Shu, & von Davier, 2015)*

Distances in SciPy package

Distance functions between two numeric vectors u and v . Computing distances over a large collection of vectors is inefficient for these functions. Use `pdist` for this purpose.

<code>braycurtis(u, v[, w])</code>	Compute the Bray-Curtis distance between two 1-D arrays.
<code>canberra(u, v[, w])</code>	Compute the Canberra distance between two 1-D arrays.
<code>chebyshev(u, v[, w])</code>	Compute the Chebyshev distance.
<code>cityblock(u, v[, w])</code>	Compute the City Block (Manhattan) distance.
<code>correlation(u, v[, w, centered])</code>	Compute the correlation distance between two 1-D arrays.
<code>cosine(u, v[, w])</code>	Compute the Cosine distance between 1-D arrays.
<code>euclidean(u, v[, w])</code>	Computes the Euclidean distance between two 1-D arrays.
<code>jensenn Shannon(p, q[, base])</code>	Compute the Jensen-Shannon distance (metric) between two 1-D probability arrays.
<code>mahalanobis(u, v, VI)</code>	Compute the Mahalanobis distance between two 1-D arrays.
<code>minkowski(u, v[, p, w])</code>	Compute the Minkowski distance between two 1-D arrays.
<code>seuclidean(u, v, V)</code>	Return the standardized Euclidean distance between two 1-D arrays.
<code>squared_euclidean(u, v[, w])</code>	Compute the squared Euclidean distance between two 1-D arrays.
<code>wminkowski(u, v, p, w)</code>	Compute the weighted Minkowski distance between two 1-D arrays.

Distance functions between two boolean vectors (representing sets) u and v . As in the case of numerical vectors, `pdist` is more efficient for computing the distances between all pairs.

<code>dice(u, v[, w])</code>	Compute the Dice dissimilarity between two boolean 1-D arrays.
<code>hamming(u, v[, w])</code>	Compute the Hamming distance between two 1-D arrays.
<code>jaccard(u, v[, w])</code>	Compute the Jaccard-Needham dissimilarity between two boolean 1-D arrays.
<code>kulsinski(u, v[, w])</code>	Compute the Kulsinski dissimilarity between two boolean 1-D arrays.
<code>rogerstanimoto(u, v[, w])</code>	Compute the Rogers-Tanimoto dissimilarity between two boolean 1-D arrays.
<code>russellrao(u, v[, w])</code>	Compute the Russell-Rao dissimilarity between two boolean 1-D arrays.
<code>sokalmichener(u, v[, w])</code>	Compute the Sokal-Michener dissimilarity between two boolean 1-D arrays.
<code>sokalsneath(u, v[, w])</code>	Compute the Sokal-Sneath dissimilarity between two boolean 1-D arrays.
<code>yule(u, v[, w])</code>	Compute the Yule dissimilarity between two boolean 1-D arrays.

`hamming` also operates over discrete numerical vectors.

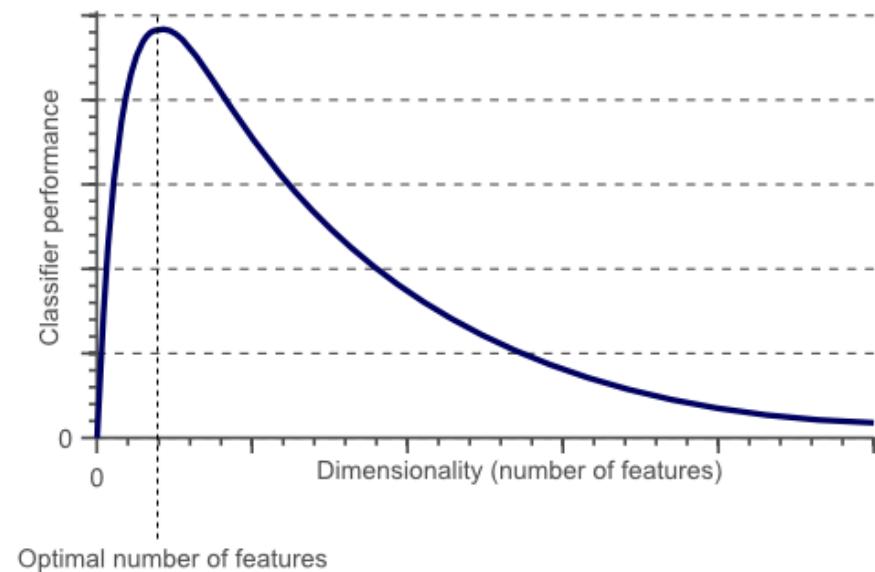


High Dimensional Space

- High dimensional space is beyond our intuition
- Do not take things for granted in high dimensional space
- The curse of dimensionality

$$\lim_{d \rightarrow \infty} E \left(\frac{\text{dist}_{\max}(d) - \text{dist}_{\min}(d)}{\text{dist}_{\min}(d)} \right) \rightarrow 0$$

Beyer et al., 1997



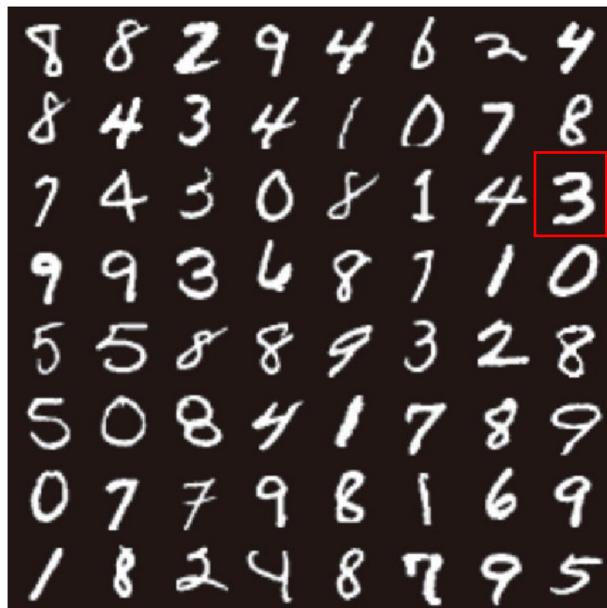
<https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/>



Manifold Hypothesis

Real world high-dimensional data lie on low-dimensional manifolds embedded within the high-dimensional space.

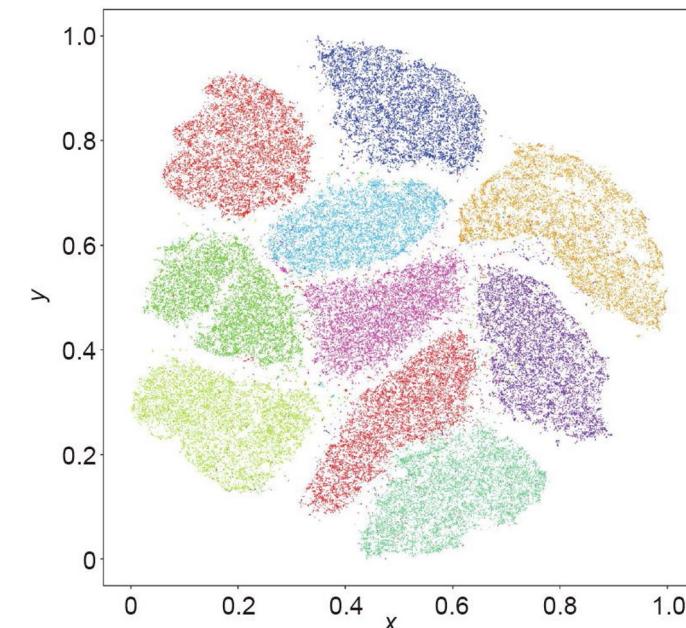
MNIST dataset



Each handwritten digit image is 28 by 28 pixels, leading to a space of 784 dimensions

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

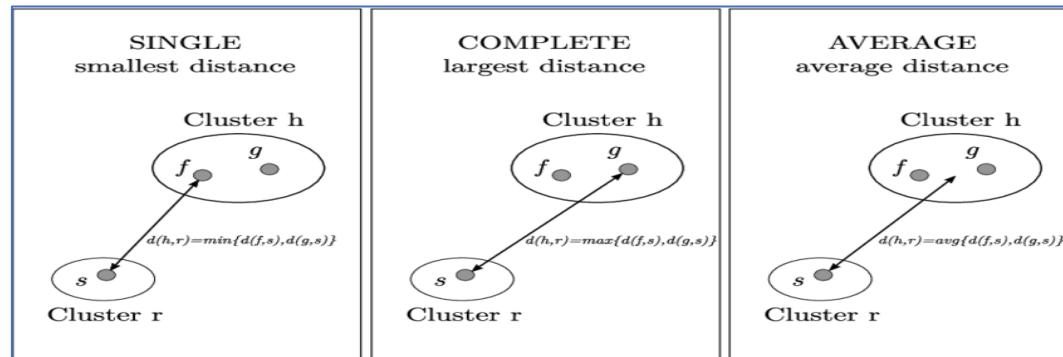
TSNE embedding



<https://www.sciencedirect.com/science/article/pii/S2095809919302279#b0015>

Hierarchical Clustering

- Hierarchical clustering: method seeking to build a hierarchy of clusters.
- Two strategies:
 - Divisive – top down approach
 - Agglomerative – bottom up approach
- Distance: pairwise distance between the points
- Linkage: determines the distance between sets of points as a function of the pairwise distances between points.



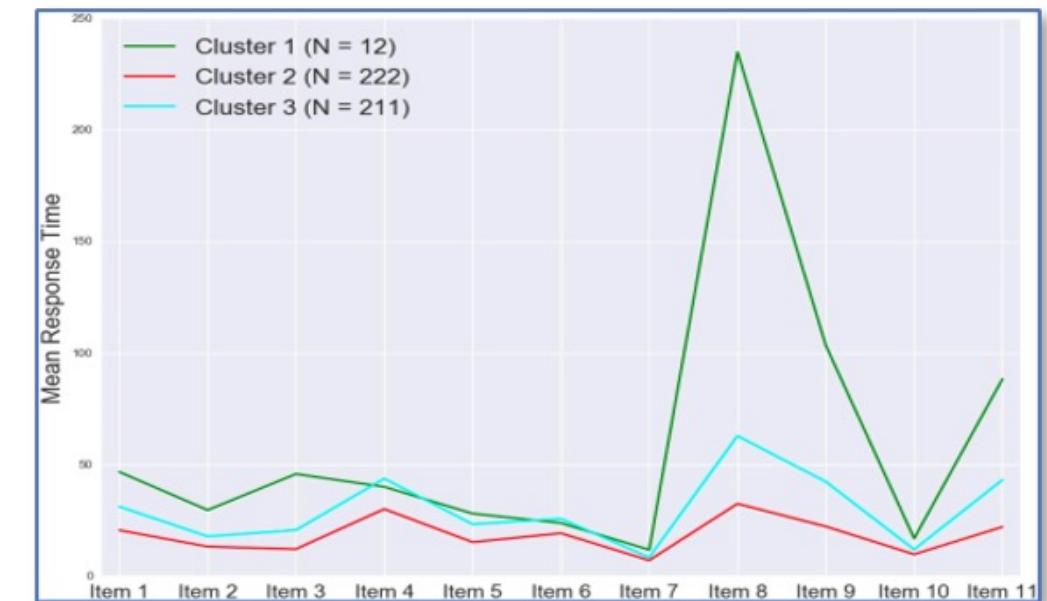
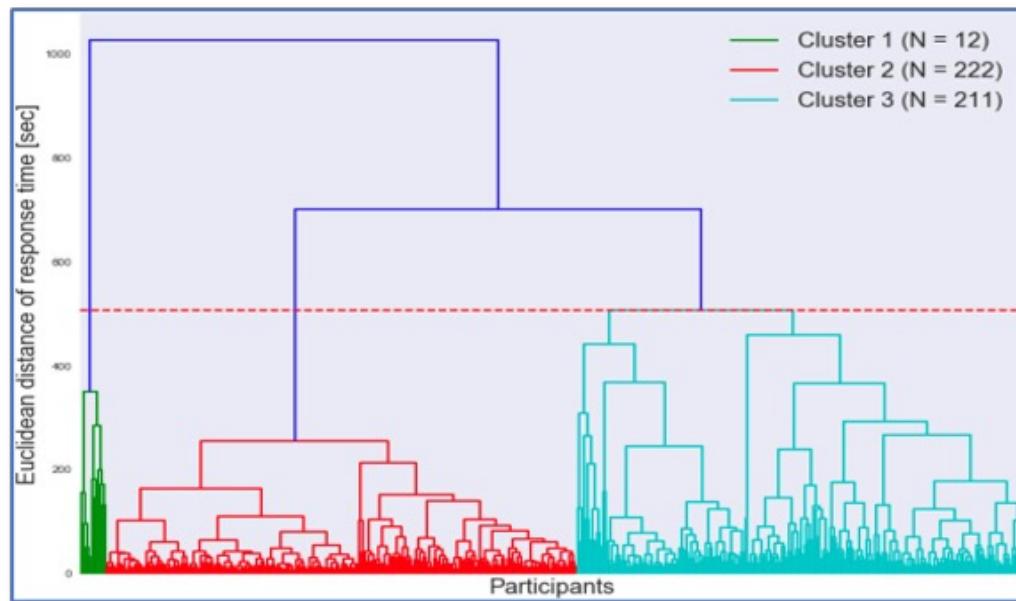
- Ward's minimum variance criterion: minimizes the total within-cluster variance.

https://en.wikipedia.org/wiki/Hierarchical_clustering

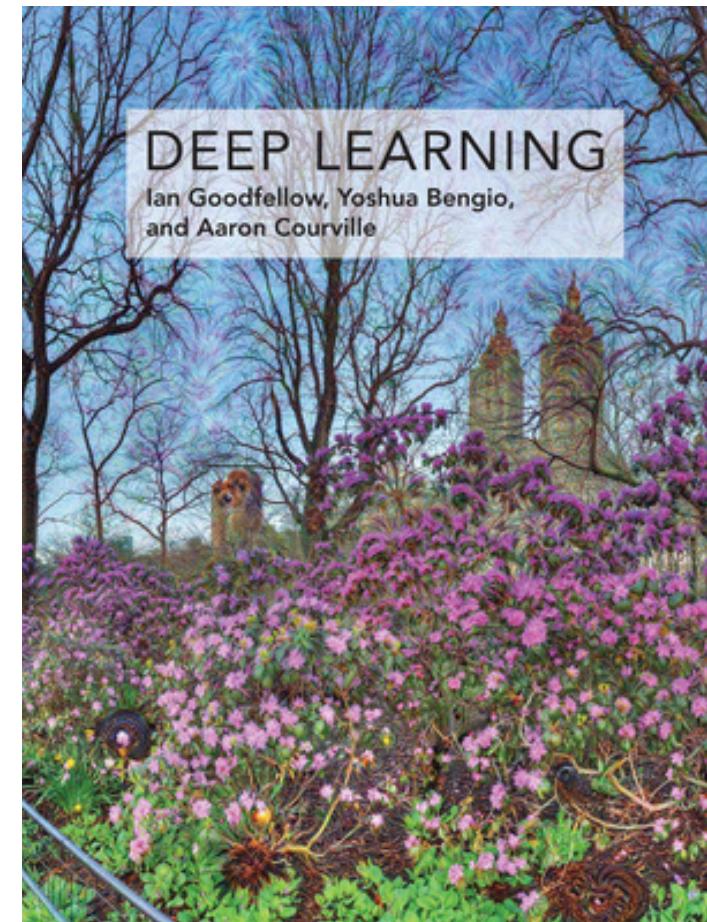
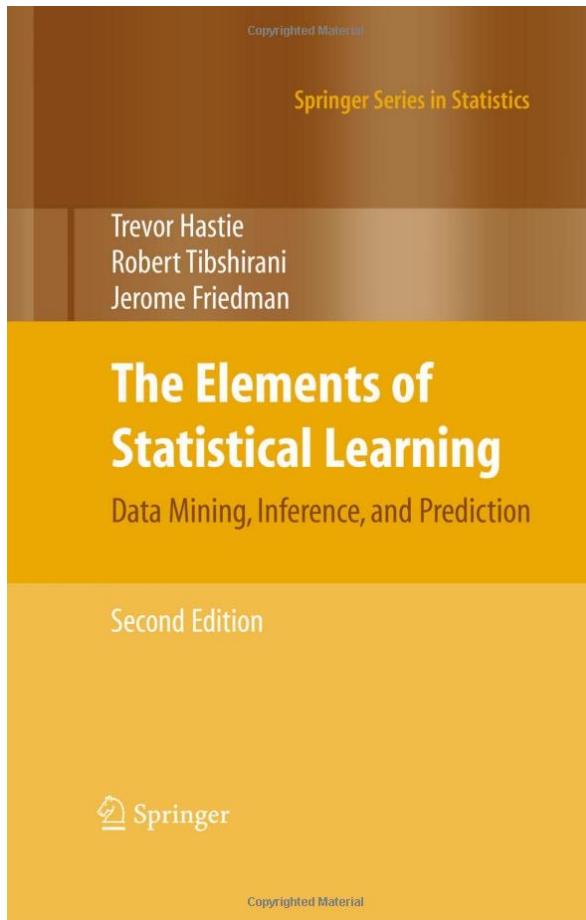
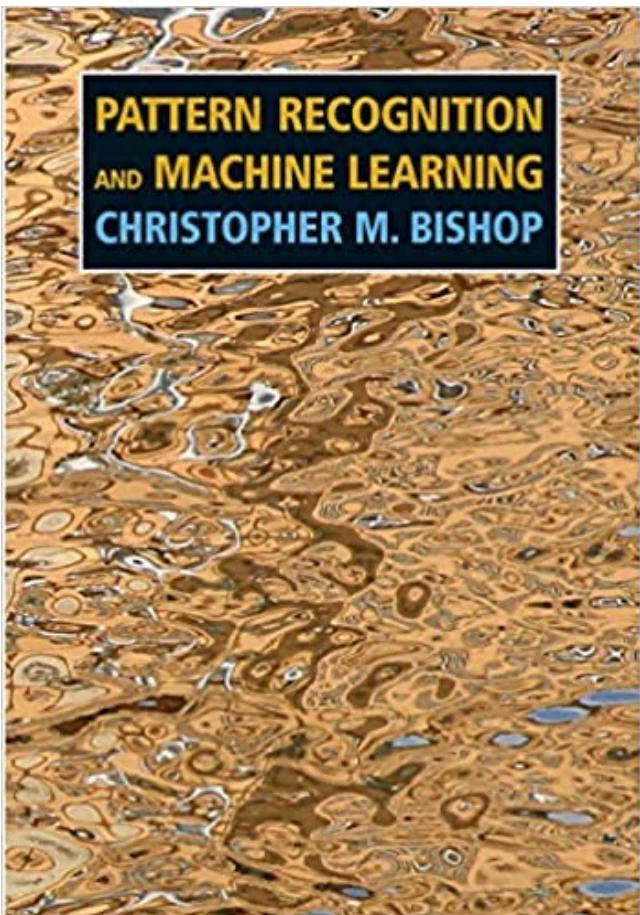


Example Application - Student Clustering

- Based on student's response time to different items, cluster them into different groups (response style)



Recommended Books



Software and Packages



Scikit-learn

scikit-learn
Machine Learning in Python

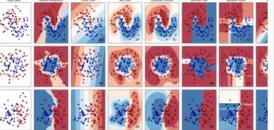
Getting Started | Release Highlights for 0.23 | GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification
Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...

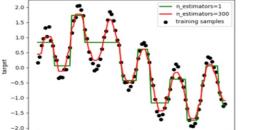


Examples

Regression
Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...



Examples

Clustering
Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, and more...

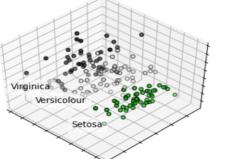


Examples

Dimensionality reduction
Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: k-Means, feature selection, non-negative matrix factorization, and more...

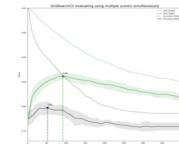


Examples

Model selection
Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning

Algorithms: grid search, cross validation, metrics, and more...

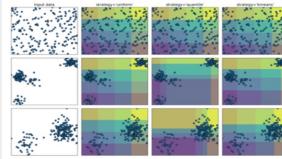


Examples

Preprocessing
Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.

Algorithms: preprocessing, feature extraction, and more...



Examples

One of the main open-source ML package in Python

Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language

Jiangang Hao, Tin Kam Ho

First Published February 20, 2019 | Review Article | <https://doi.org/10.3102/1076998619832248>

Article information ▾

Altmetric 5



Abstract

Machine learning is a popular topic in data analysis and modeling. Many different machine learning algorithms have been developed and implemented in a variety of programming languages over the past 20 years. In this article, we first provide an overview of machine learning and clarify its difference from statistical inference. Then, we review *Scikit-learn*, a machine learning package in the Python programming language that is widely used in data science. The *Scikit-learn* package includes implementations of a comprehensive list of machine learning methods under unified data and modeling procedure conventions, making it a convenient toolkit for educational and behavior statisticians.

Keywords

machine learning, Python, Scikit-learn

```
model = classifier(hyperparameters = something)
model.fit(X_train, y_train)
y_test = model.predict(X_test)
```



Orange

The screenshot shows the homepage of the Orange data mining software. At the top, there's a navigation bar with links for Features, Screenshots, Workflows, Download, Blog, Docs, Workshops, and a yellow 'Donate' button. The main title 'orange' is displayed in a large, stylized font where the letter 'o' has a magnifying glass icon integrated into it. Below the title, the tagline 'Data Mining Fruitful and Fun' is written in a large serif font. Underneath the tagline, two descriptive sentences are present: 'Open source machine learning and data visualization.' and 'Build data analysis workflows visually, with a large, diverse toolbox.' To the right of the text, there's a cartoon illustration of a large orange character wearing glasses and holding a clipboard, surrounded by smaller circular characters representing different data mining tools like classifiers and regression models.

orange.biolab.si

Features Screenshots Workflows Download Blog Docs Workshops Donate

orange

Data Mining
Fruitful and Fun

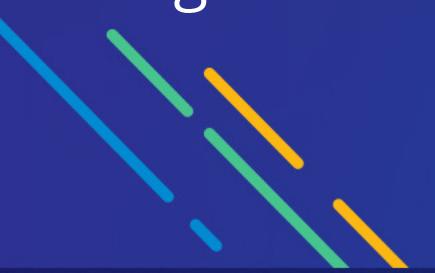
Open source machine learning and data visualization.
Build data analysis workflows visually, with a large, diverse toolbox.

Download Orange





Natural Language Processing Basics: Text Mining and Automated Scoring

- Language models
 - Text representation and mining
 - Automated scoring
 - Deep learning models
- 

Chapter 14

Text Mining and Automated Scoring

Michael Flor and Jiangang Hao



Abstract Natural Language Processing (NLP) is playing an increasingly important role in learning and assessments. Some typical applications of NLP in education include automated scoring, automated item generation, conversation-based assessments, writing assistants, text mining for education, and so on. In this chapter, we aim at introducing some basics of NLP through two typical applications in educational contexts, text mining and automated scoring. We hope readers can get an overall picture of NLP and get familiarized with some basic tools for handling natural language data, which may serve as stepping stones for their future work with NLP.

14.1 Overview

Natural Language Processing, also known as Computational Linguistics, is an interdisciplinary area of research, spanning computer science and artificial intelligence (AI), linguistics and cognitive psychology, as well as other disciplines, such as mathematics, logic, philosophy, and neuroscience. The central questions of research revolve around the notion of enabling computer programs to automatically analyze and represent natural human language (e.g. English). Sub-areas of inquiry include natural language understanding and natural language generation (for text-based data), and speech recognition and generation (for spoken data). As an area of study, computational linguistics has both theoretical and applied perspectives, and the notion of NLP often refers to the more applied areas of the field.

The R or Python codes can be found at the GitHub repository of this book: https://github.com/jgbrainstorm/computational_psychometrics

M. Flor (✉) · J. Hao
Educational Testing Service, Princeton, NJ, USA
e-mail: mflor@ets.org; jhao@ets.org



all\author

The limits of my language are the limits of
my mind. All I know is what I have words
for.

-Ludwig Wittgenstein

NLP Basics

- Natural Language Processing, also known as computational linguistics, focuses on enabling computer programs to automatically analyze and represent natural human language
- NLP tasks: https://en.wikipedia.org/wiki/Natural_language_processing
- NLP in educational assessment
 - Text mining
 - Conversational assessment
 - Automated scoring
 - Question generation
 - ...



Language Model

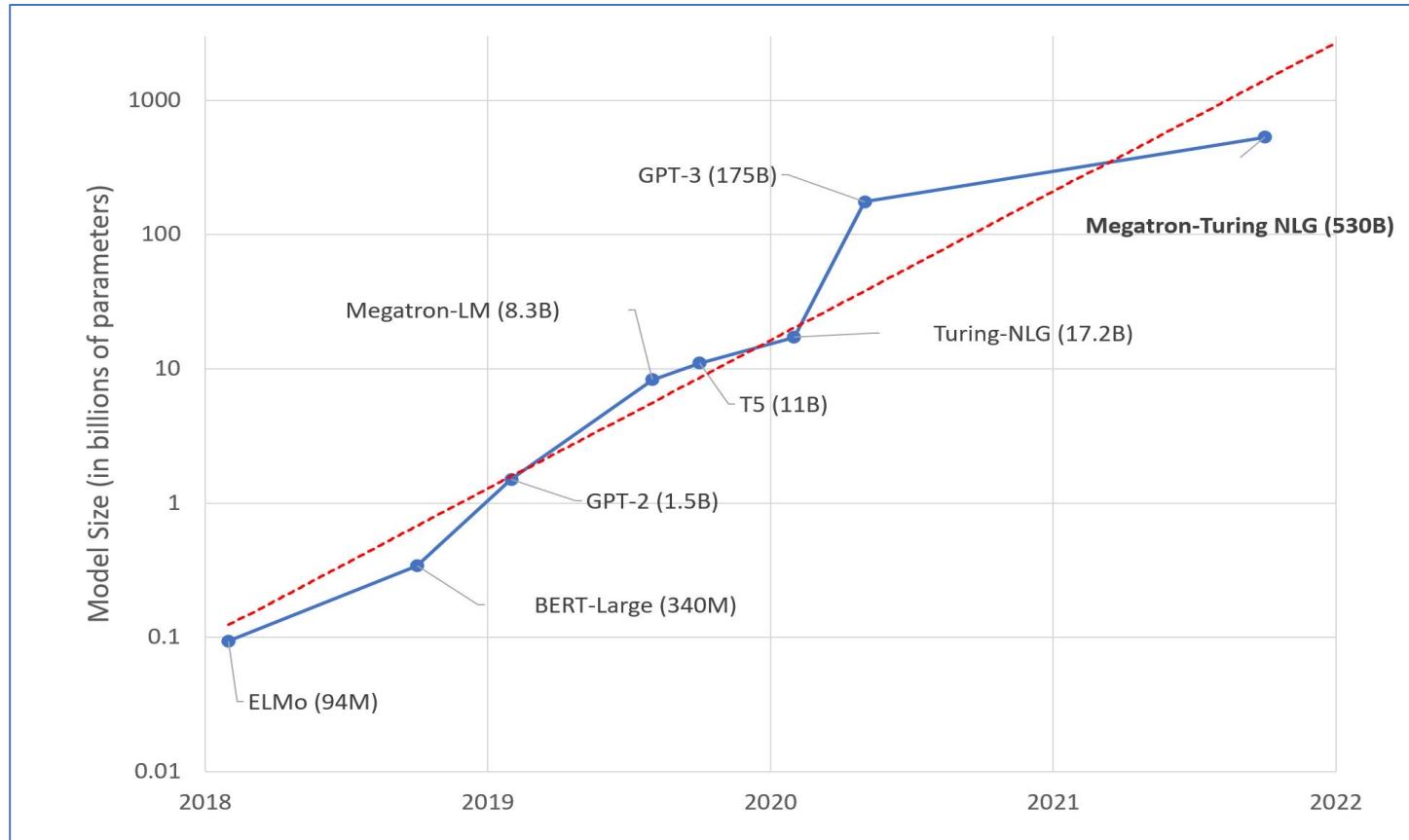
- Language model specify the probability of a given sequence of words
- N-gram model: <https://web.stanford.edu/~jurafsky/slp3/3.pdf>
 - $P(w_1)$
 - $P(w_1, w_2) = P(w_1 | w_2)P(w_2)$
 - $P(w_1, w_2, w_3) = P(w_1 | w_2, w_3)P(w_2 | w_3)P(w_3)$
 - All we need to know is $P(w_t | w_1, w_2, \dots, w_{t-1})$ and $P(w)$
 - Bigram model: $P(w_t | w_1, w_2, \dots, w_{t-1}) = P(w_t | w_{t-1})$ – Markov assumption
 - N-gram model: $P(w_t | w_{t-n+1}, \dots, w_{t-1})$
 - Problems: the computation scales exponentially with N; Sparse
- Neural language models



<https://web.stanford.edu/~jurafsky/slp3/7.pdf>



Large Language Models - LLM



<https://clockwise.software/blog/what-is-gpt-3/>

<https://huggingface.co/blog/large-language-models>



Text Representation

- Convert text into numerical representations
- Preprocessing
- Bag of words
 - Frequency of unique tokens/words
- N-gram
 - Frequency of n consecutive tokens/words
- Word embedding
 - Each word is mapped into a vector of smaller dimension
 - Latent semantic analysis - LSA
 - Neural embedding, e.g., Word2vec



Preprocessing

- Case normalization
- Tokenization
- Stop words removal
- Typo correction
- POS tagging
- Stemming and lemmatization

CC coordinating conjunction
CD cardinal digit
DT determiner
EX existential there (like: "there is" ... think of it like "there exists")
FW foreign word
IN preposition/subordinating conjunction
JJ adjective 'big'
JJR adjective, comparative 'bigger'
JJS adjective, superlative 'biggest'
LS list marker 1)
MD modal could, will
NN noun, singular 'desk'
NNS noun plural 'desks'
NNP proper noun, singular 'Harrison'
NNPS proper noun, plural 'Americans'
PDT predeterminer 'all the kids'
POS possessive ending parent's
PRP personal pronoun I, he, she
PRP\$ possessive pronoun my, his, hers
RB adverb very, silently,
RBR adverb, comparative better
RBS adverb, superlative best
RP particle give up
TO to go 'to' the store.
UH interjection errrrrrrrm
VB verb, base form take
VBD verb, past tense took
VBG verb, gerund/present participle taking
VBN verb, past participle taken
VBP verb, sing. present, non-3d take
VBZ verb, 3rd person sing. present takes
WDT wh-determiner which
WP wh-pronoun who, what
WP\$ possessive wh-pronoun whose
WRB wh-abverb where, when

stemming

Form	Suffix	Stem
studies	-es	studi
studying	-ing	study
niñas	-as	niñ
niñez	-ez	niñ

lemmatization

Form	Morphological information	Lemma
studies	Third person, singular number, present tense of the verb study	study
studying	Gerund of the verb study	study
niñas	Feminine gender, plural number of the noun niño	niño
niñez	Singular number of the noun niñez	niñez



Bag of Words - Unigram

Document-Term Matrix

	<i>Term 1</i>	<i>Term 2</i>	...	<i>Term t</i>
<i>Document 1</i>	2	1
<i>Document 2</i>	5	1
<i>Document 3</i>	6	1
...
<i>Document d</i>	1	2	...	

- The cell numbers are the counts of the term in the document
- The matrix is sparse

	about	bird	heard	is	the	word	you
About the bird, the bird, bird bird bird	1	5	0	0	2	0	0
You heard about the bird	1	1	1	0	1	0	1
The bird is the word	0	1	0	1	2	1	0

TF-IDF Transformation

- Limitation of the document-term matrix
 - Frequency of different terms depends on the length of the documents
 - Some words more likely to appear in more documents
- Term frequency weighting (TF)

$$TF(\text{term}_j) = \frac{\text{Total frequency of term}_j \text{ in document}_i}{\text{Total frequency of all terms in document}_i}$$

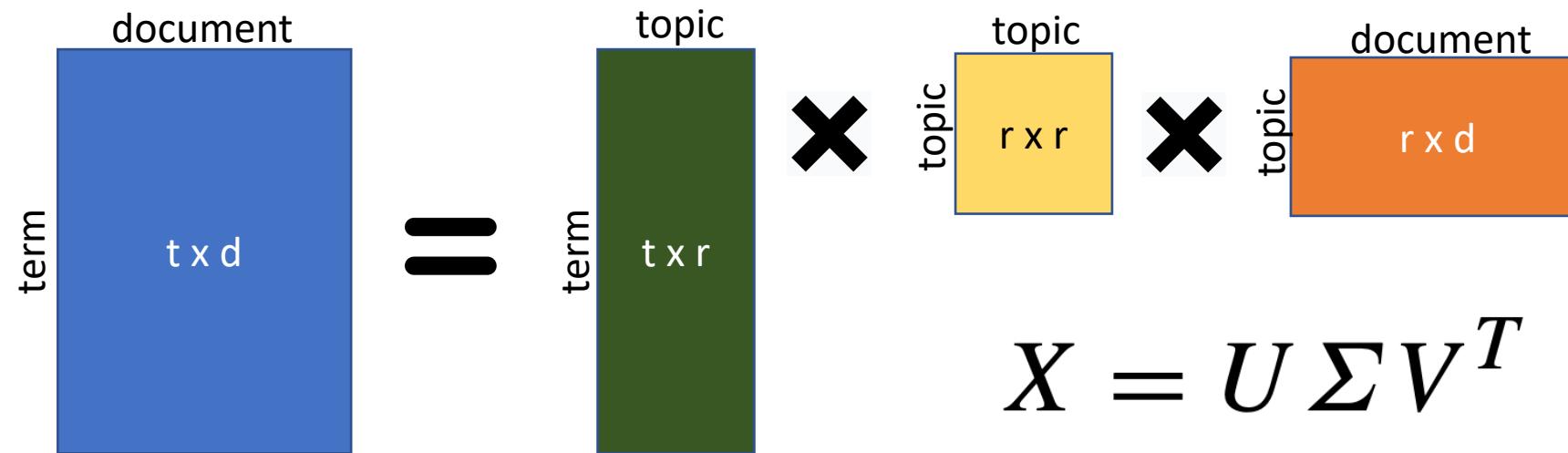
- Inverse document frequency weighting (IDF)

$$IDF(\text{term}_j) = \log \frac{\text{Total number of documents}}{1 + \text{Number of documents containing term}_j}$$



Latent Semantic Analysis - LSA

Singular Value Decomposition



Where U is a t by r matrix and V is a d by r matrix with r is the rank of X . Both U and V have orthonormal columns, e.g., $UU^T = VV^T = I$. Σ is a r by r diagonal matrix of singular values that are non-negative and ordered in decreasing magnitude by convention.

Dimensional “Reduction”

Approximate the original term-document matrix

$$\hat{X} = U_k \Sigma_k V_k^T \xrightarrow{\text{Approximate}} X = U \Sigma V^T$$

SVD provides a way to find a lower rank matrix to approximate the term-document matrix. If we consider only the part of Σ that contains the largest k singular values (e.g., k topics in the LSA context), we have a $\hat{X} = U_k \Sigma_k V_k^T$, where U_k is a t by k matrix, Σ_k is a k by k matrix, and V_k is a d by k matrix. The U_k matrix contains information regarding how the k topics are related to the terms and the V_k matrix contains information on how the k topics are related to the documents, as



Vectors from LAS Embedding

Term (word) vectors

	<i>Topic 1</i>	<i>Topic 2</i>	...	<i>Topic k</i>
<i>Term 1</i>
<i>Term 2</i>
<i>Term 3</i>
...
<i>Term t</i>

Document vectors

	<i>Topic 1</i>	<i>Topic 2</i>	...	<i>Topic k</i>
<i>Document 1</i>
<i>Document 2</i>
<i>Document 3</i>
...
<i>Document t</i>

Choosing the k : coherence score

$$\text{Coherence} = \sum \text{score}(w_i, w_j)$$

$$\text{score}_{\text{UMass}}(w_i, w_j) = \log \frac{D(w_i, w_j) + 1}{D(w_i)}$$

- $D(w_i)$ is the count of documents containing the word w_i
- $D(w_i, w_j)$ is the count of documents containing both words w_i and w_j



Neural Embedding

- Word2vec: Mikolov et al, 2013
 - Each word is represented by a vector of several hundred dimensions
 - Words that share common contexts in the corpus are located close to each other in the vector space – distributional hypothesis
 - Better than LSA for large language corpora
- Two model architectures of word2vec
 - Continuous bag-of-words (CBOW): predict the target word from the window of surrounding context words
 - Continuous skip-gram: predict the surrounding window of context words using the target word

https://web.stanford.edu/~jurafsky/slp3/slides/6_Vector_Apr18_2021.pdf

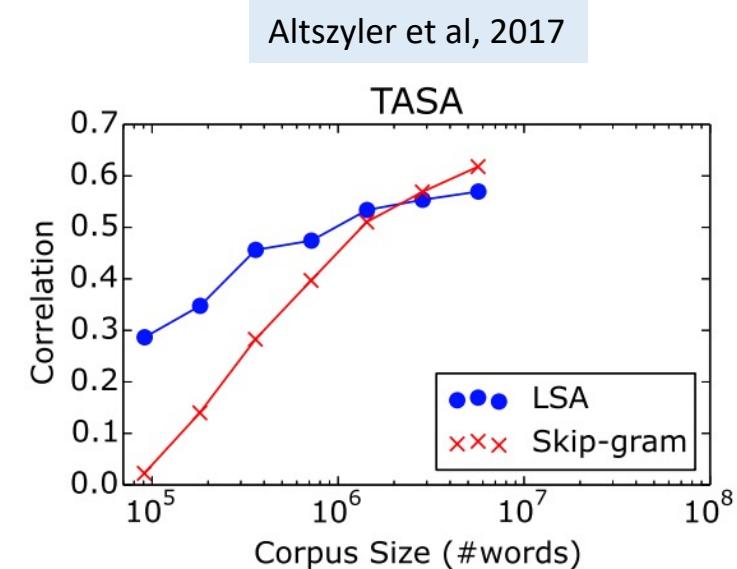


Performance of Different Embedding

- For large corpus, neural embedding (word2vec) is generally doing better
- For small corpus, LSA embedding is generally doing better than word2vec
- For small corpus, ngram is also doing better than word2vec

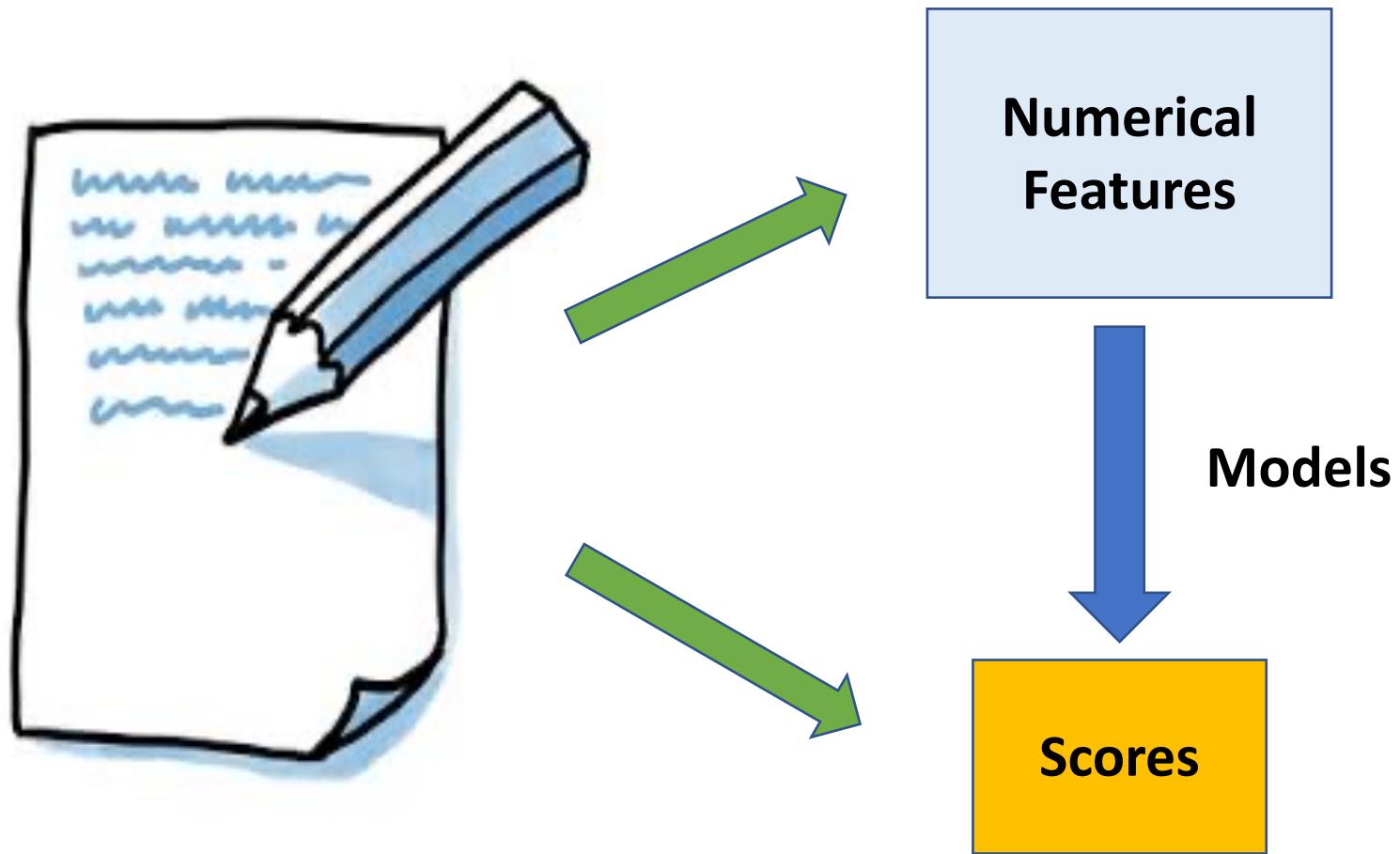
Methods	N-gram		Document vector	
	Accuracy	Kappa	Accuracy	Kappa
MaxEnt	0.669	0.551	0.625	0.488
Random Forest	0.697	0.588	0.611	0.468
Linear SVC	0.723	0.623	0.619	0.479
Linear Chain CRF	0.736	0.641	0.643	0.515

Hao, et al, 2018 (unpublished)



Corpus dependent!

Automated Scoring



SAS vs. AES

- Short Answer Scoring (SAS)
 - Contextualized features
 - Ngram
 - Word/sentence vectors
 - Machine learning model to map the features to scores
 - Example: ETS CPS-rater (Hao, et al., 2017)
- Automated Essay Scoring (AES)
 - Generic features
 - E.g., Grammar, Usage, Mechanics, Style, Development, etc.
 - Contextualized features (on/off topic)
 - Statistical/Machine learning model to map the features to scores
 - Example: ETS e-rater (Burstein, 2003)



Software Packages

Top Python NLP Libraries



NLTK (Natural Language Toolkit)
All-inclusive Python library with the all the necessary set of features



Text Blob
An open-soused solution perfectly suitable for starting the journey into the NLP programming and testing AI projects



spaCy
Implementation of individual communication with clients



Gensim
Text-specific library for working with topic modelling and word vectors



CoreNLP
One of the most powerfull Python libraries developed by Stenford University and perfectly suitable for large projects

Many many others...



Deep Learning Models

The screenshot shows the Hugging Face website interface. The top navigation bar includes a lock icon, the URL "huggingface.co/models", the Hugging Face logo, a search bar, and links for Models, Datasets, Spaces, Docs, Solutions, and Pricing.

The main content area is divided into sections: Tasks, Libraries, and Datasets on the left, and a list of Models on the right.

Tasks:

- Image Classification
- Translation
- Image Segmentation
- Fill-Mask
- Automatic Speech Recognition
- Token Classification
- Sentence Similarity
- Audio Classification
- Question Answering
- Summarization
- Zero-Shot Classification
- + 22 Tasks

Libraries:

- PyTorch
- TensorFlow
- JAX + 30

Datasets:

- mozilla-foundation/common_voice_7_0
- squad
- wikipedia
- common_voice
- glue
- emotion
- bookcorpus
- xtreme + 303

Models (81,994):

- xlm-roberta-base**
Updated Jun 6 · ↓ 32.7M · ❤️ 94
- gpt2**
Updated 8 days ago · ↓ 32.7M · ❤️ 254
- bert-base-uncased**
Updated 24 days ago · ↓ 25.3M · ❤️ 313
- openai/clip-vit-large-patch14**
Updated 24 days ago · ↓ 12.1M · ❤️ 51
- roberta-base**
Updated 28 days ago · ↓ 11M · ❤️ 68
- bert-base-cased**
Updated Sep 6, 2021 · ↓ 8.51M · ❤️ 50

Recommended Books

Speech and Language Processing

An Introduction to Natural Language Processing,
Computational Linguistics, and Speech Recognition

Third Edition draft

Daniel Jurafsky
Stanford University

James H. Martin
University of Colorado at Boulder

Copyright ©2021. All rights reserved.

Draft of January 12, 2022. Comments and typos welcome!



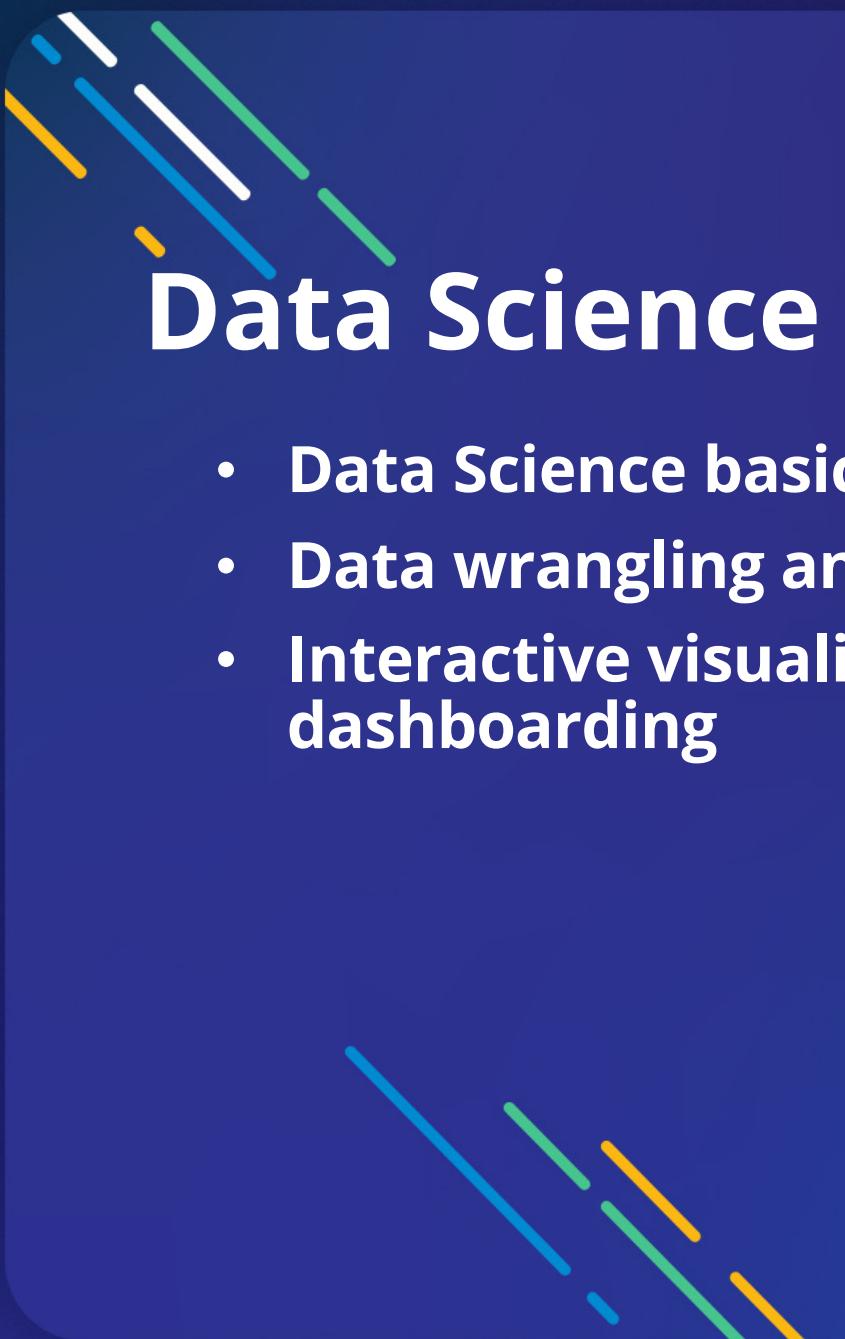
Summary

- New methods from data science, machine learning, NLP and other quantitative disciplines are needed to handle complex data in educational measurement.
- Psychometric researchers need to acquire these new skills to meet the challenges from digital learning and assessment. You are playing critical roles in helping to build the future of the field.

Thank You!

jhao@ets.org





Self Study

Data Science

- **Data Science basics**
- **Data wrangling and processing**
- **Interactive visualization and dashboarding**

Chapter 8 A Data Science Perspective on Computational Psychometrics

Jiangang Hao and Robert J. Mislevy



Abstract Digitally based learning and assessment systems generate large volumes of complex process data. The next generation psychometricians need to acquire new data science skills to meet the data challenge. In this chapter, we summarize data science skills and identify the subset that psychometricians need to prioritize. We introduce an evidence identification centered data design (EICDD) process during the task design, as an important way to address the data challenges from digitally based assessments. We describe some specific data techniques to parse and process complex process data with example codes in Python programming language. We also outline the general methodological strategies when dealing with process data from digitally based assessments.

8.1 Introduction

Digitally Based Assessments (DBAs) enable the capture of test takers' response process information at finer time granularity than traditional forms of assessment, and these rich process data can provide new opportunities for validating assessments, improving measurement precision, revealing response patterns/styles, uncovering group difference (fairness), detecting test security breaches, identifying new constructs, and providing feedback to learners and other stakeholders (Ercikan & Pellegrino, 2017; Mislevy et al., 2014). But these potential benefits do not come for free. The significantly increased volume, velocity, and variety of data pose new challenges to psychometricians for handling, analyzing, and interpreting the

The R or Python codes can be found at the GitHub repository of this book: https://github.com/jgbrainstorm/computational_psychometrics

J. Hao (✉) · R. J. Mislevy
Educational Testing Service, Princeton, NJ, USA
e-mail: jhao@ets.org; rmislevy@umd.edu

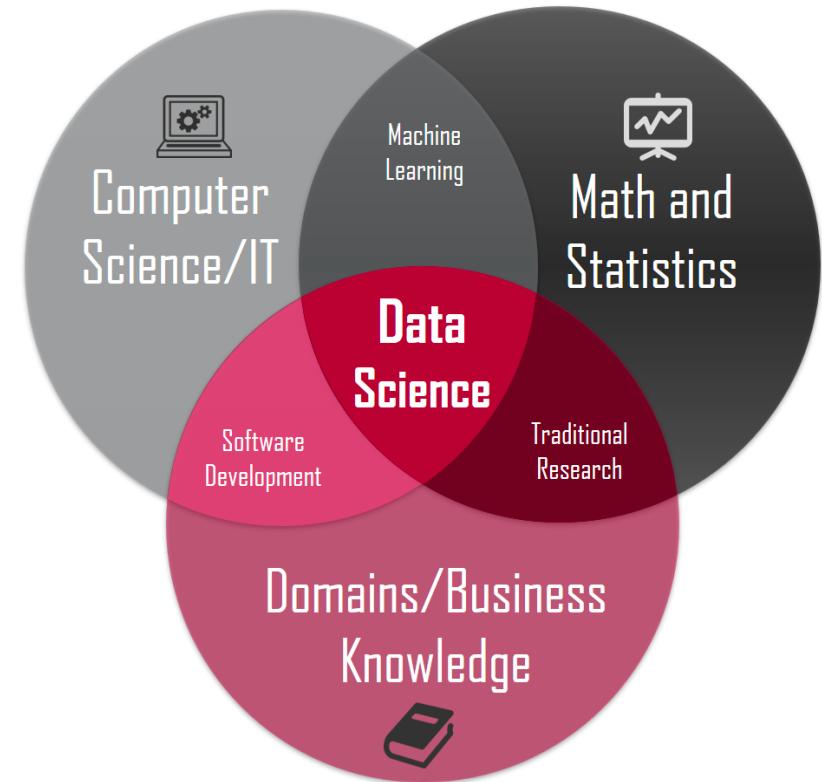
Data Science Basics



What is Data Science

Data science is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data

- Wikipedia



Skills Needed for a Data Scientist

Quality as a scientist

- Curious & creative
- Patient, detail-oriented, & persistent
- Open-minded & courageous
- Computational thinking
- Critical thinking
- Problem solving
- Hands-on and doing
- Collaborative and communicative

Technical skills

- Programming in Python/R
- Computer/IT/DevOps
- Cloud computing for big data
- Data wrangling and visualization
- Statistics, Math
- Machine learning



Data Types and Data Storage

Data Types

- Raw data (structured or non-structured)
 - Images, videos, audios
 - Documents
 - Records from a single testing sessions
 - ...
- Processed/value-added data (mostly structured)
 - Features from raw data
 - Rational tables
 - Records of many students' test score
 - ...

Data Storage

- **Data Lake:** a centralized repository that allows you to store all your **Structured** and unstructured raw data at any scale
 - File folders
 - Cloud-based folder - AWS S3 bucket
- **Data Warehouse:** a central repository of processed data that can be used for multiple purpose
 - RDBMS: Relational Database Management System
 - MySQL, Oracle, PostgreSQL, etc.



Design Data through Data Model

A **data model** is an abstract model that organizes elements of data and standardizes how they relate to one another and to the properties of real-world entities. (Wikipedia)

- **Relational Model (SQL)**

- Data is organized into relations (tables in SQL DB) where each relation is an unordered collection of tuples (rows in SQL) – Codd (1970)
- Suited for data where there are many-to-many relations
- SQL database: Oracle, PostgreSQL, MySQL, etc.

- **Document Model (NoSQL)**

- Hierarchical/tree structure (JSON/XML)
- Suited for data where mostly are one-to-many relations
- NoSQL database: Mongodb, Redis, CouchDB, Hbase, etc.

- **Graph Model**

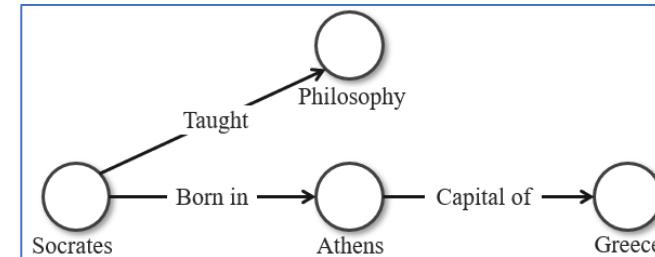
- Vertices (nodes or entities) and edges (relationship or arcs)
- Social graph: Vertices -> people, Edges -> which people know each other
- Web graph: Vertices -> web pages, Edges -> html links
- Suited for data where the many-to-many relationship is too complicated to be handled by the relational model
- Neo4j, ArangoDB, etc.



Triple Store and RDF

- **Triple Store Model**

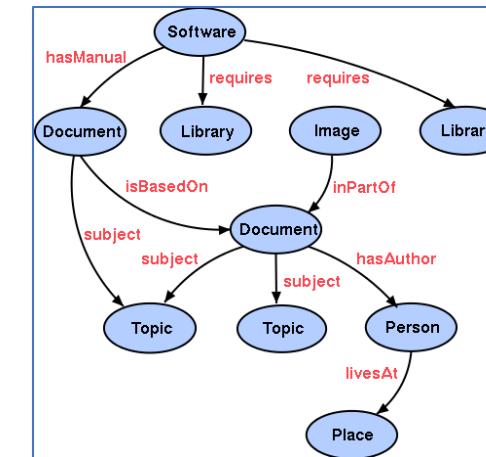
- (subject, predicate, object)
- E.g. (Jiangang, like, apple), (Oren, like, orange)
- Knowledge graph for search engine and AI applications



<https://towardsdatascience.com/auto-generated-knowledge-graphs-92ca99a81121>

- **Resource Description Framework (RDF)**

- Semantic web (Berners-Lee, 2001)
- Similar to the triple store model, but referring to webpages
- Uniform Resource Identifier (URI)
 - URN: Uniform Resource Name
 - URL: Uniform Resource Locator
- (URI_1, URI_2, URI_3)



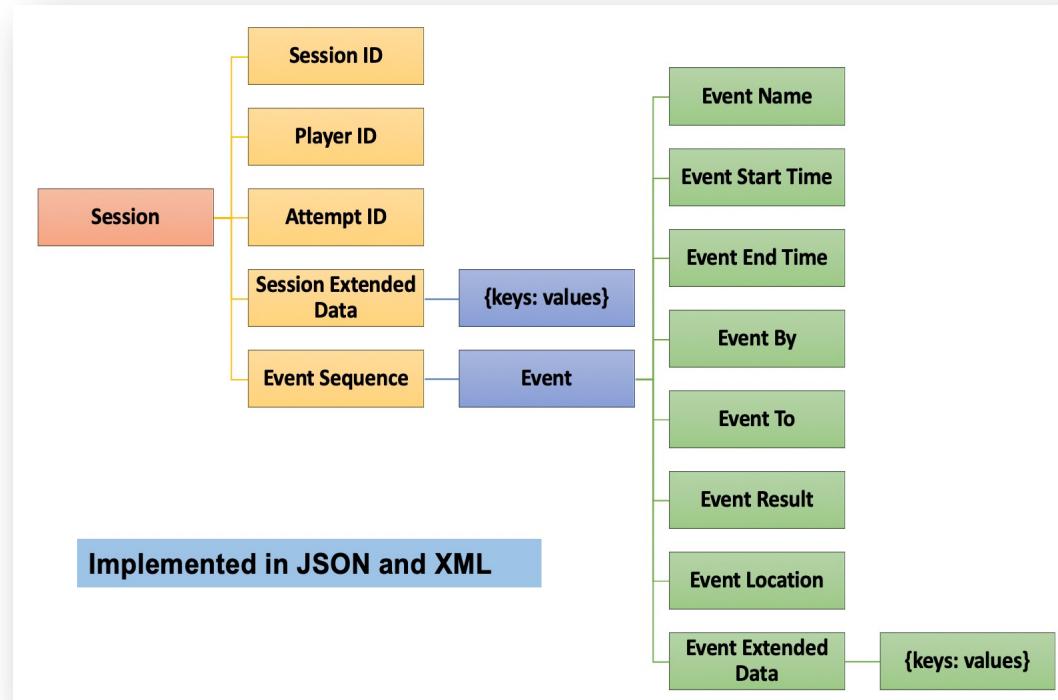
<https://vodkhang.com/intelligent-system/semantic-technology-and-its-application>

Data Models in Learning and Assessments

- **xAPI:** experience API (or Tin Can API) is an eLearning specification that makes it possible to collect data about the wide range of experiences a person has within online and offline training activities.
 - Evolved from SCORM (Shareable Content Object Reference Model, <http://scorm.com>)
 - For learning management system (LMS)
 - Triple store style data model: Actor > Verb > Object (Activity)
 - <https://adlnet.gov/projects/xapi-architecture-overview/>
- **IMS Global Learning Consortium' Caliper:** IMS enables a plug-and play-architecture and ecosystem that provides a foundation on which innovative products can be rapidly deployed and work together seamlessly.
 - For learning management system (LMS)
 - Triple store style data model: Actor > Verb > Object (Activity)
 - <https://www.imsglobal.org/activity/caliper#caliperpublic>
- **ETS Data Model for Virtual Performance Assessment**
 - Document data model for process data from virtual performance assessments (VPAs, such as game/simulation-based assessments)
 - <https://onlinelibrary.wiley.com/doi/full/10.1002/ets2.12096>



ETS Data Model for Virtual Performance Assessment



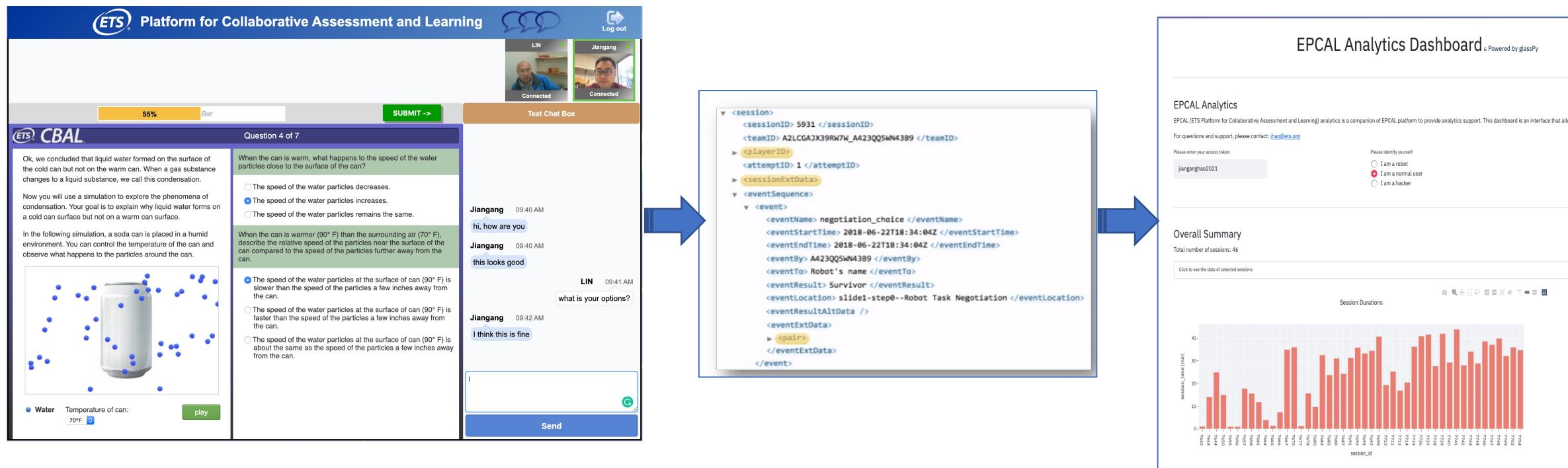
- Evidence Identification Centered Data Design - an extra process in Evidence Centered Design
 - Log file -> Evidence Trace File

JSON Implementation

XML Implementation

A Full Implementation in EPCAL

Full implemented in the ETS Platform for Collaborative Assessment and Learning



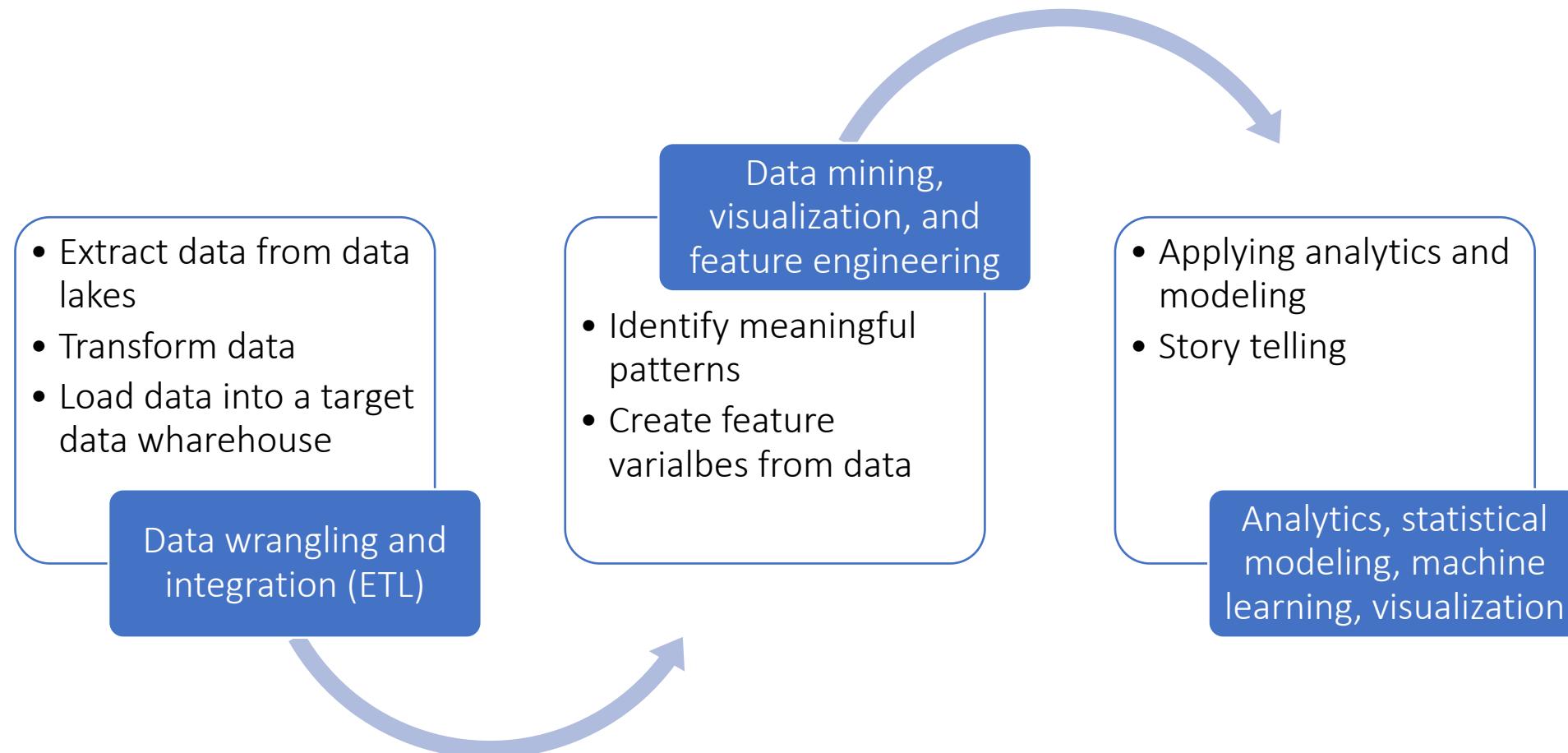
<https://onlinelibrary.wiley.com/doi/full/10.1002/ets2.12181>



Data Processing and Wrangling



General Steps of Data Processing



Types of Data Processing

- Batch processing
 - Process data after the data are collected
 - E.g., Apache Hadoop, ...
- Stream processing
 - Process live data
 - E.g., Apache Kafka, Storm, Streamz, ...
- Micro-batch processing
 - Something in between the Batch and Stream processing
 - Apache Spark

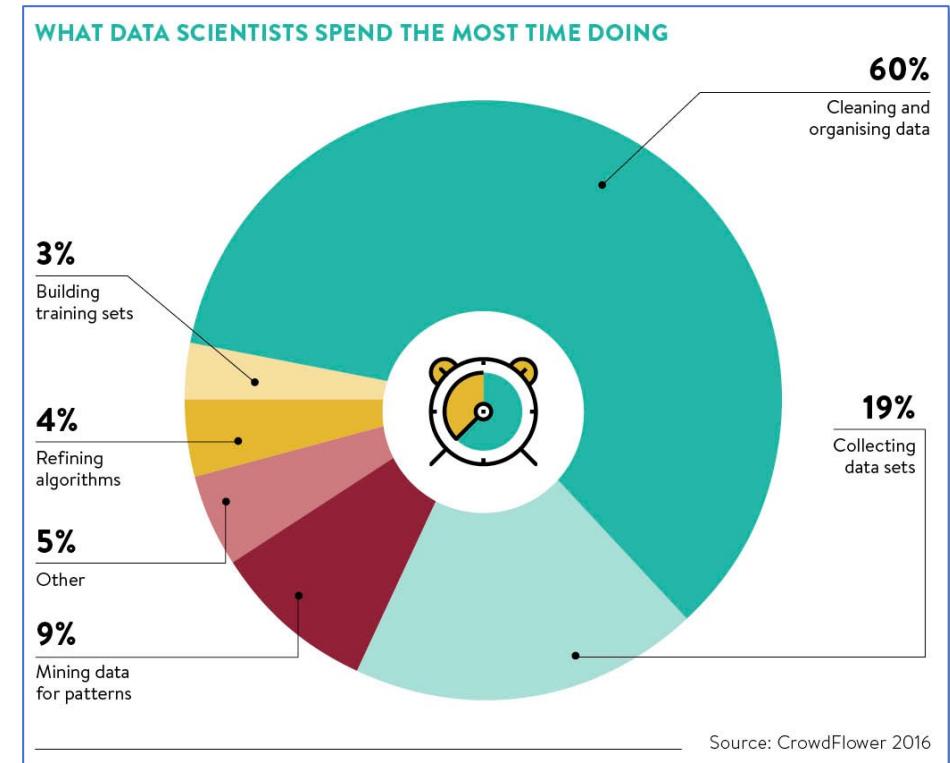
The landscape of the software tools changes very fast and new tools emerge from time to time



Data Wrangling

Data wrangling, also known as data munging, is the process of transforming and mapping data from one “raw” format into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics

- Wikipedia



Structured and Unstructured Data

- **Unstructured data**: information that either does not have a predefined data model or is not organized in a pre-defined manner
- **Structured data**: information created by following a predefined data model (usually a tabular format)
- **Semi-structured data (also known as self-describing structure)**: a form of structured data that does not obey the tabular structure of data models. It contains tags or other markers to separate semantic elements and enforce hierarchies of records and fields within the data. HTML, XML, JSON, YAML, etc.



Unstructured text file

```
[5/15/2013 2:17:26 PM] Session Start
[5/15/2013 2:17:26 PM] Leaving sequence: loadXML, moving forward.
[5/15/2013 2:17:30 PM] Player submitted name: Carl
[5/15/2013 2:17:30 PM] Leaving sequence: InputNameScreen, moving forward.
[5/15/2013 2:17:31 PM] Player submitted name: Carl
[5/15/2013 2:17:31 PM] Leaving sequence: startScreen, moving forward.
[5/15/2013 2:17:50 PM] Player submitted name: Carl
[5/15/2013 2:17:50 PM] Leaving sequence: slide2, moving forward.
[5/15/2013 2:17:55 PM] Player submitted name: Carl
[5/15/2013 2:17:55 PM] Leaving sequence: slide2b, moving forward.
[5/15/2013 2:18:34 PM] Player submitted name: Carl
[5/15/2013 2:18:34 PM] Leaving sequence: slide2c, moving forward.
[5/15/2013 2:20:09 PM] Player submitted name: Carl
[5/15/2013 2:20:09 PM] Leaving sequence: slide3, moving forward.
[5/15/2013 2:20:13 PM] Player submitted name: Carl
[5/15/2013 2:20:13 PM] Leaving sequence: slide4, moving forward.
```



Semi-structured - XML

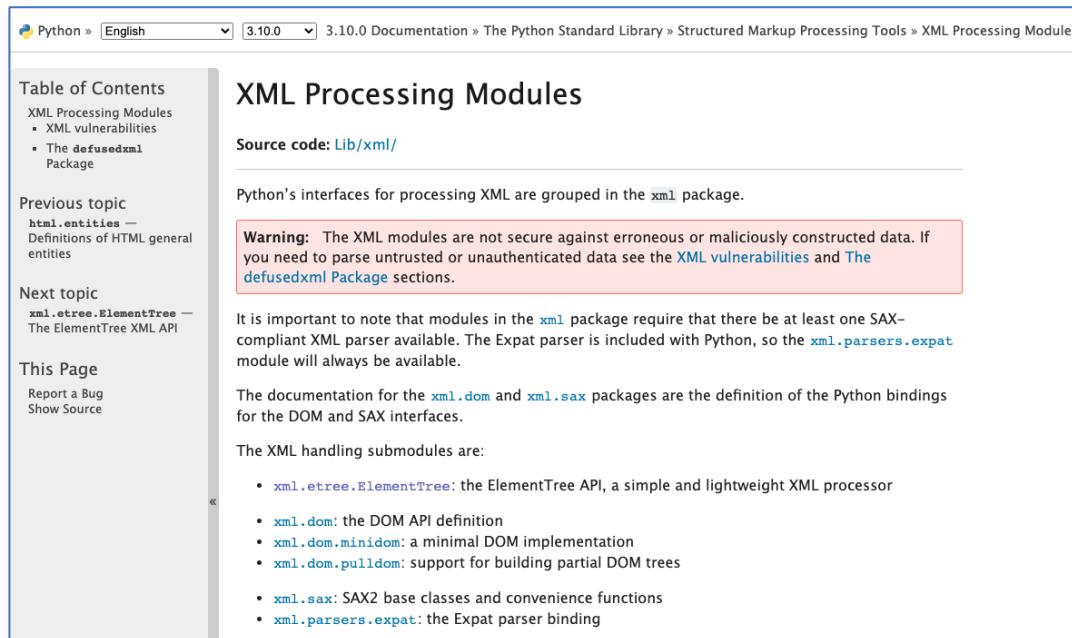
- XML – Extensible Markup Language, W3C 1998
 - Markup and contents
 - <start_time> 10:10 </start_time>
 - <![CDATA[this is the comments]]>
 - Tag
 - Start-tag <shape>
 - End-tag </shape>
 - Empty-tag <shape/>
 - Elements
 - <start_time> 10:10 </start_time>
 - Attributes: name-value pair
 - <start_time timezone="EST"> 10:10 </start_time>
 - Declaration: <?xml version="1.0" encoding="UTF-8"?>



Data from EPCAL

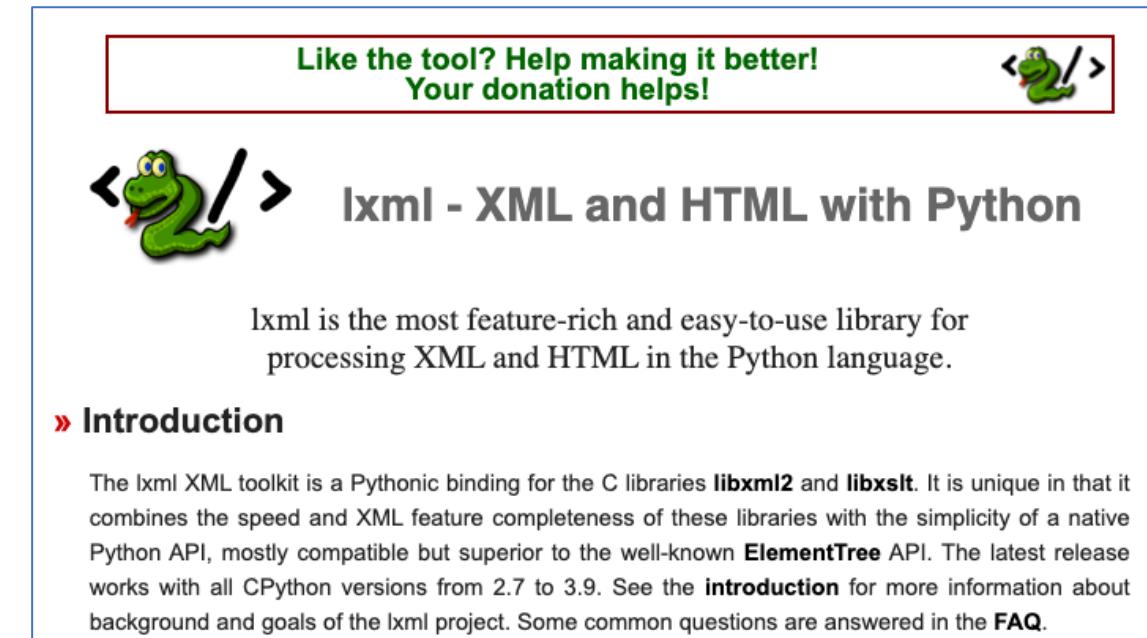
Python Tool for XML

- “Official” XML library:
<https://docs.python.org/3/library/xml.html>



The screenshot shows the Python 3.10.0 Documentation page for the `xml` package. The page title is "XML Processing Modules". It includes a "Table of Contents" sidebar with links to XML Processing Modules, XML vulnerabilities, and defusedxml. The main content area has a "Source code: Lib/xml/" link and a note stating that Python's interfaces for processing XML are grouped in the `xml` package. A warning box points out that XML modules are not secure against erroneous or maliciously constructed data. Below this, it notes that at least one SAX-compliant XML parser is required, mentioning the Expat parser included with Python. The page also lists submodules like `ElementTree`, `DOM`, `minidom`, `pulldom`, `SAX`, and `parsers`.

- LXML: <https://lxml.de/>



The screenshot shows the LXML project landing page. At the top, there is a green banner with the text "Like the tool? Help making it better! Your donation helps!" and a green snake icon. The main heading is "lxml - XML and HTML with Python". Below the heading, a paragraph explains that lxml is a feature-rich and easy-to-use library for processing XML and HTML in Python. A section titled "» Introduction" provides an overview of the project, stating that it is a Pythonic binding for libxml2 and libxslt, combining speed, completeness, and simplicity. It also mentions compatibility with ElementTree and support for various Python versions from 2.7 to 3.9.



Semi-structured - JSON

JavaScript Object Notation



Filename extension	.json
Internet media type	application/json
Type code	TEXT
Uniform Type Identifier (UTI)	public.json
Type of format	Data interchange
Extended from	JavaScript
Standard	STD 90 (RFC 8259) ECMA-404, ISO/IEC 21778:2017
Open format?	Yes
Website	json.org

Introduced in 2000

Valid Data Types

In JSON, values must be one of the following data types:

- a string
 - a number
 - an object (JSON object)
 - an array
 - a boolean
 - *null*

JSON values **cannot** be one of the following data types:

- a function
 - a date
 - *undefined*

<https://www.w3schools.com/js/>



Specifying Structure - Schema

- Schema: define the structure and data type of a xml file
- Practical use: validation- comparing data to the schema to check for compliance

XML schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xss:schema xmlns:xss="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
<!--
  Game Log Schema
  Version 2.0
  Authors:
    Lonnie Smith (lsmith@ets.org)
    Jiangang Hao (jhao@ets.org)

  Note: this is the final version as of 7/27/2015
  (c) 2014, Educational Testing Service
-->

<!-- Root element and children -->
<xss:element name="gameLog">
  <xss:complexType>
    <xss:sequence>
      <xss:element name="session" minOccurs="1" maxOccurs="unbounded">
        <xss:complexType>
          <xss:sequence>
            <xss:element name="sessionId" type="idType" minOccurs="1" maxOccurs="1"/>
            <xss:element name="playerID" type="idType" minOccurs="1" maxOccurs="1"/>
            <xss:element name="attemptID" type="idType" minOccurs="1" maxOccurs="1"/>
            <xss:element name="sessionExtData" type="dictType" minOccurs="0" maxOccurs="1"/>
            <xss:element name="eventSequence" minOccurs="1" maxOccurs="1">
              <xss:complexType>
                <xss:sequence>
                  <xss:element name="event" type="eventType" minOccurs="1" maxOccurs="unbounded"/>
                </xss:sequence>
              </xss:complexType>
            </xss:element>
          </xss:sequence>
        </xss:complexType>
      </xss:element>
    </xss:sequence>
  </xss:complexType>
</xss:element>

<!-- Data type definitions -->

<!-- ID definition. All identifiers must follow this rule -->
<xss:simpleType name="idType">
  <xss:restriction base="xs:string">
    <xss:pattern value="[a-zA-Z0-9_-]{1,}" />
  </xss:restriction>
</xss:simpleType>

<!-- Timestamps must follow subset of ISO 8601 standard, be resolved to (at least) the millisecond, and must use UTC -->
<xss:simpleType name="timestampType">
  <xss:restriction base="xs:dateTime">
    <xss:pattern value="20\d{2}-\d{2}-\d{2}\T\d{2}:\d{2}:\d{2}Z"/>
  </xss:restriction>
</xss:simpleType>
```

JSON schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description": "Gamelog Schema v1.2, Created by Jiangang Hao @ ETS",
  "type": "object",
  "properties": {
    "gameLog": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "sessionId": {
            "type": "string",
            "pattern": "[a-zA-Z0-9_-]{1,}"
          },
          "playerID": {
            "type": "string",
            "pattern": "[a-zA-Z0-9_-]{1,}"
          },
          "attemptID": {
            "type": "integer"
          },
          "sessionExtData": {
            "type": "object"
          },
          "eventSequence": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "eventName": {
                  "type": "string",
                  "pattern": "[a-zA-Z0-9_-]{1,}"
                }
              }
            }
          },
          "eventStartTime": {
            "type": "string",
            "pattern": "20\\d{2}-\\d{2}-\\d{2}\\T\\d{2}:\\d{2}:\\d{2}Z"
          },
          "eventEndTime": {
            "type": "string",
            "pattern": "20\\d{2}-\\d{2}-\\d{2}\\T\\d{2}:\\d{2}:\\d{2}Z"
          }
        }
      }
    }
  }
}
```



Python for Data Wrangling

 **pandas**

Original author(s)	Wes McKinney
Developer(s)	Community
Initial release	11 January 2008; 13 years ago [citation needed]
Stable release	1.3.0 ^[1] / 2 July 2021; 3 months ago
Repository	github.com/pandas-dev/pandas 
Written in	Python, Cython, C
Operating system	Cross-platform
Type	Technical computing
License	New BSD License
Website	pandas.pydata.org 

Data Wrangling
with pandas Cheat Sheet
<http://pandas.pydata.org>

Pandas [API Reference](#) Pandas [User Guide](#)

Creating DataFrames

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index=[1, 2, 3])
Specify values for each column.
```

```
df = pd.DataFrame(
    [[4, 7, 10],
     [5, 8, 11],
     [6, 9, 12]],
    index=[1, 2, 3],
    columns=['a', 'b', 'c'])
Specify values for each row.
```

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index=pd.MultiIndex.from_tuples([
        ('d', 1), ('d', 2),
        ('e', 2)], names=['n', 'v']))
Create DataFrame with a MultiIndex.
```

Method Chaining

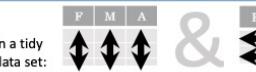
Most pandas methods return a DataFrame so that another pandas method can be applied to the result. This improves readability of code.

```
df = (pd.melt(df)
      .rename('columns={
          'variable': 'var',
          'value': 'val'})  

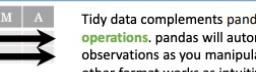
      .query('val >= 200'))
```

Tidy Data – A foundation for wrangling in pandas

In a tidy data set:



&



Tidy data complements pandas's **vectorized operations**. pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas.

Reshaping Data – Change layout, **sorting, reindexing, renaming**

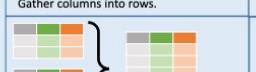
In a tidy data set:



pd.melt(df) Gather columns into rows.



df.pivot(columns='var', values='val') Spread rows into columns.



pd.concat([df1, df2]) Append rows of DataFrames



pd.concat([df1, df2], axis=1) Append columns of DataFrames

Subset Observations - rows

Subset Variables - columns

Subsets - rows and columns

Using query

regex (Regular Expressions) Examples

Cheatsheet for pandas (<http://pandas.pydata.org>) originally written by Iván Lústig, Princeton Consultants, inspired by RStudio Data Wrangling CheatSheet

https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf



Interactive Visualization and Dashboarding



Visualization is Important

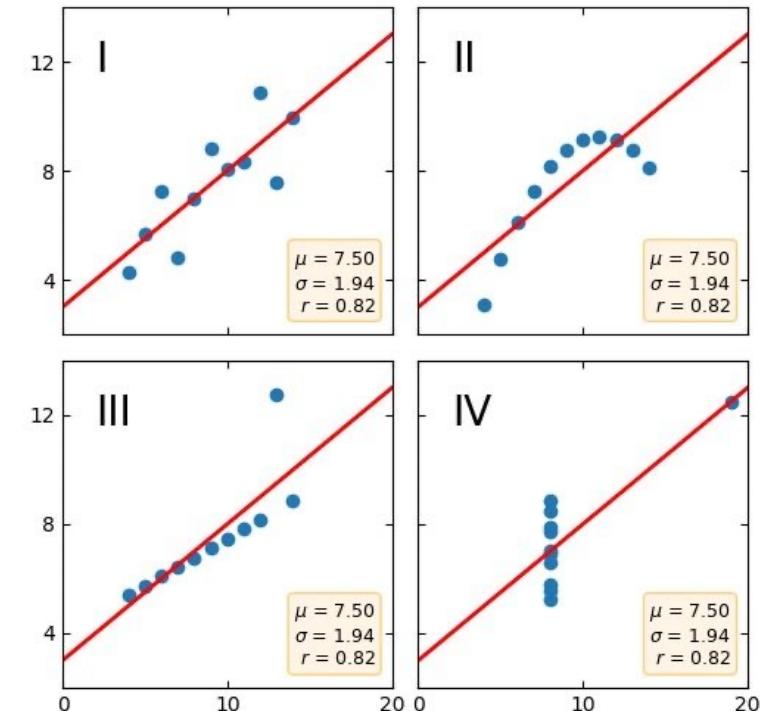


Numerical quantities focus on
expected values, graphical
summaries on unexpected values.

— John Tukey —

AZ QUOTES

Anscombe's quartet, 1973



Steps to Visualization

- Understand the nature of your data
 - Categorical/continuous
 - Sparse
 - Understand your goals
 - Audience
 - Production/exploration?
 - Interactivity required?
 - Plan the types of visualizations you'll create
 - Scatter plot, distribution, dendrogram, etc.
 - Choose a visualization framework(s)
 - Do it!

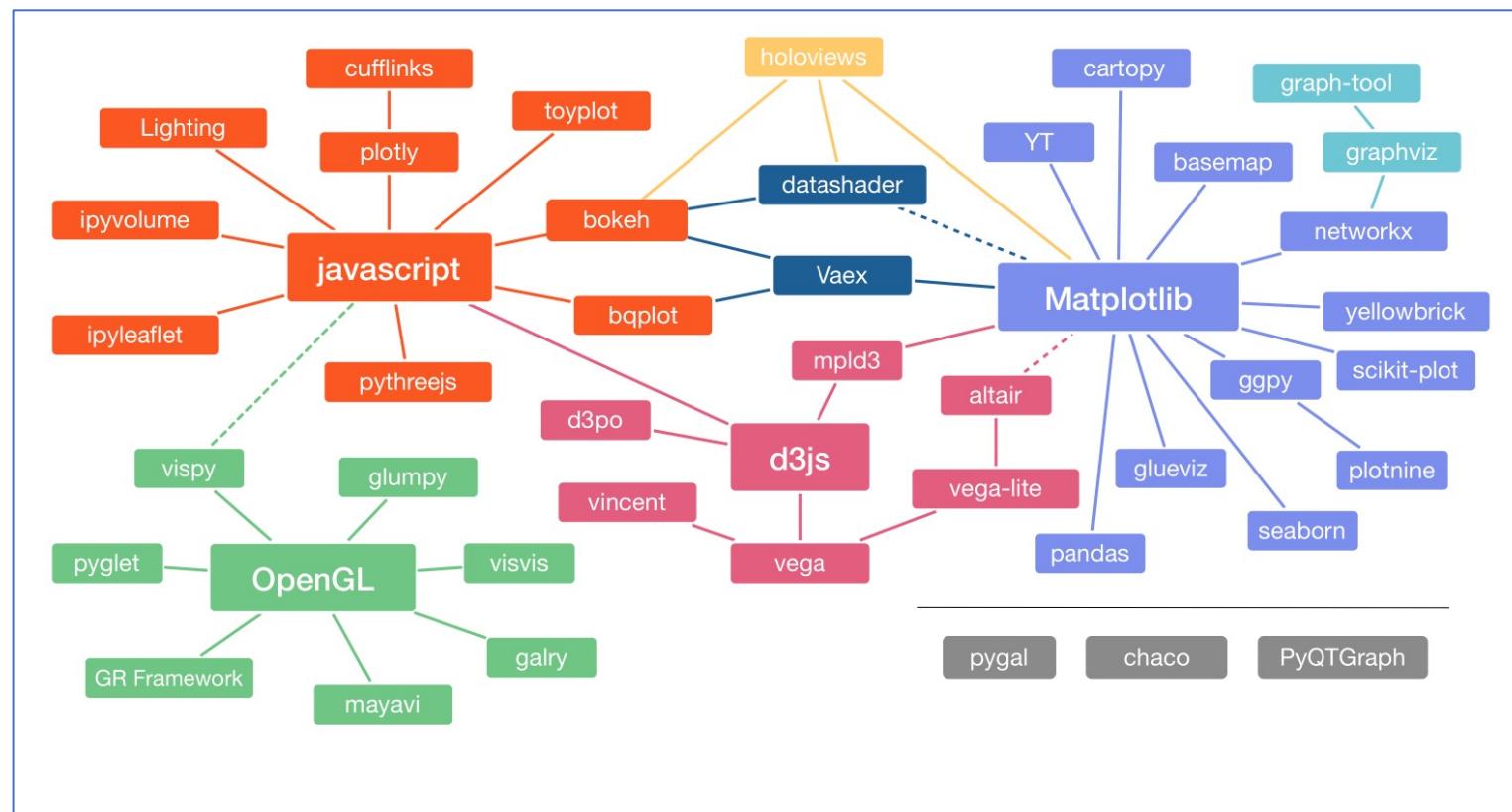


<https://chart.guide/topics/chartguide-poster-4-0/>



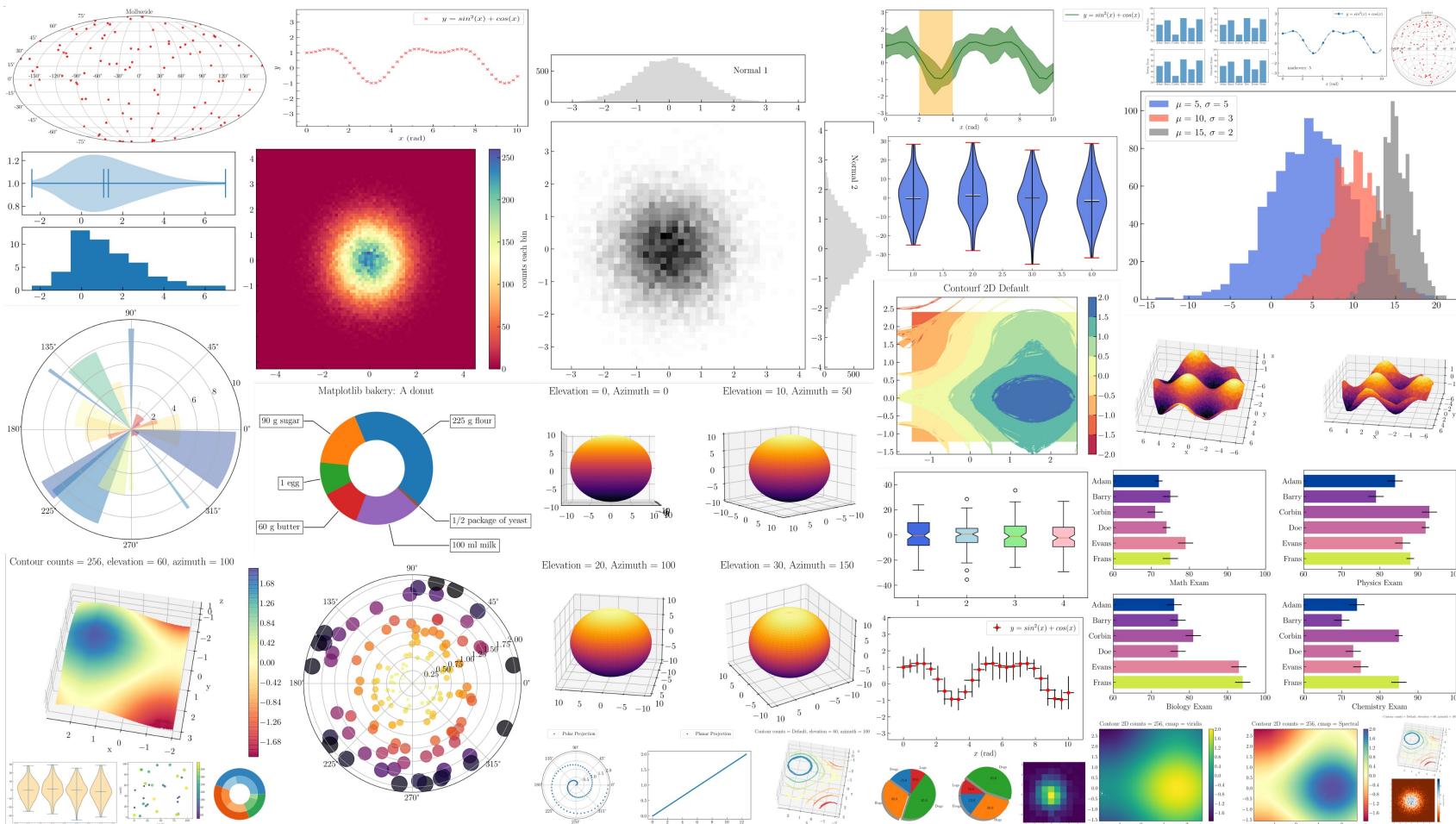
Visualization in Python

- PyViz is your gateway: www.pyviz.org
- Visualization paradigms



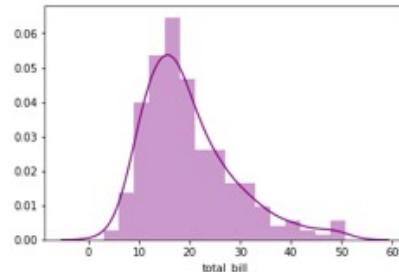
Matplotlib

<https://matplotlib.org/>

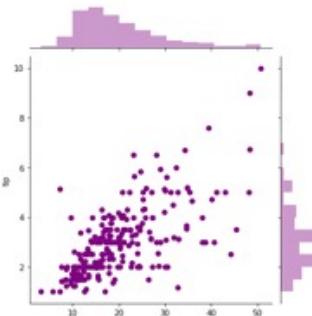


Seaborn: built on top of matplotlib

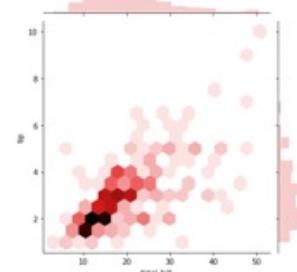
<https://seaborn.pydata.org/>



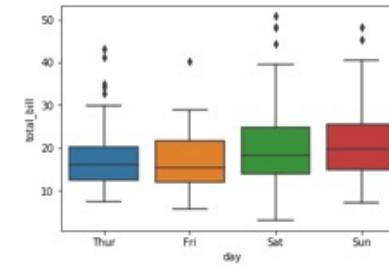
distplot



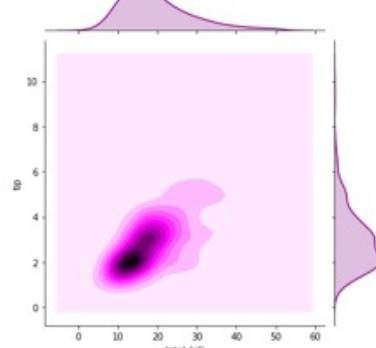
Jointplot



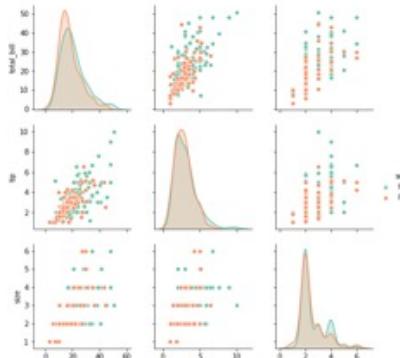
Hexplots



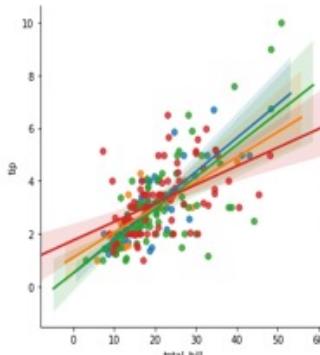
Boxplots



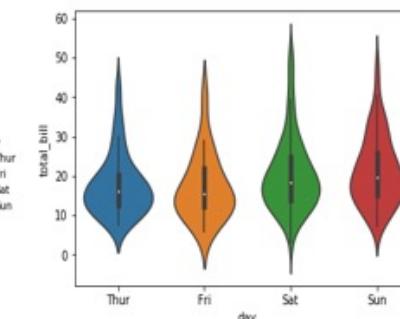
KDE Plot



Pair Plots



LM Plots



Violin Plots



Plotly Express and ipywidgets

The `plotly.express` module (usually imported as `px`) contains functions that can create entire figures at once, and is referred to as Plotly Express or PX. Plotly Express is a built-in part of the `plotly` library, and is the recommended starting point for creating most common figures. Every Plotly Express function uses `graph_objects` internally and returns a `plotly.graph_objects.Figure` instance. Throughout the `plotly` documentation, you will find the Plotly Express way of building figures at the top of any applicable page, followed by a section on how to use graph objects to build similar figures. Any figure created in a single function call with Plotly Express could be created using graph objects alone, but with between 5 and 100 times more code.

Plotly Express provides [more than 30 functions for creating different types of figures](#). The API for these functions was carefully designed to be as consistent and easy to learn as possible, making it easy to switch from a scatter plot to a bar chart to a histogram to a sunburst chart throughout a data exploration session. *Scroll down for a gallery of Plotly Express plots, each made in a single function call.*

Plotly Express currently includes the following functions:

- **Basics:** `scatter`, `line`, `area`, `bar`, `funnel`, `timeline`
- **Part-of-Whole:** `pie`, `sunburst`, `treemap`, `funnel_area`
- **1D Distributions:** `histogram`, `box`, `violin`, `strip`
- **2D Distributions:** `density_heatmap`, `density_contour`
- **Matrix Input:** `imshow`
- **3-Dimensional:** `scatter_3d`, `line_3d`
- **Multidimensional:** `scatter_matrix`, `parallel_coordinates`, `parallel_categories`
- **Tile Maps:** `scatter_mapbox`, `line_mapbox`, `choropleth_mapbox`, `density_mapbox`
- **Outline Maps:** `scatter_geo`, `line_geo`, `choropleth`
- **Polar Charts:** `scatter_polar`, `line_polar`, `bar_polar`
- **Ternary Charts:** `scatter_ternary`, `line_ternary`

IntSlider

- The slider is displayed with a specified, initial `value`. Lower and upper bounds are defined by `min` and `max`, and the value can be incremented according to the `step` parameter.
- The slider's label is defined by `description` parameter
- The slider's `orientation` is either 'horizontal' (default) or 'vertical'
- `readout` displays the current value of the slider next to it. The options are `True` (default) or `False`
 - `readout_format` specifies the format function used to represent slider value. The default is `'.2f'`

```
[2]: widgets.IntSlider(
    value=7,
    min=0,
    max=10,
    step=1,
    description='Test:',
    disabled=False,
    continuous_update=False,
    orientation='horizontal',
    readout=True,
    readout_format='d'
)
```

[2]: Test: 7

FloatSlider

```
[3]: widgets.FloatSlider(
    value=7.5,
    min=0,
    max=10.0,
    step=0.1,
    description='Test:',
    disabled=False,
    continuous_update=False,
    orientation='horizontal',
    readout=True,
    readout_format='.1f',
)
```

[3]: Test: 7.5

An example of sliders displayed vertically.

Demos



Major Dashboard Tools in Python

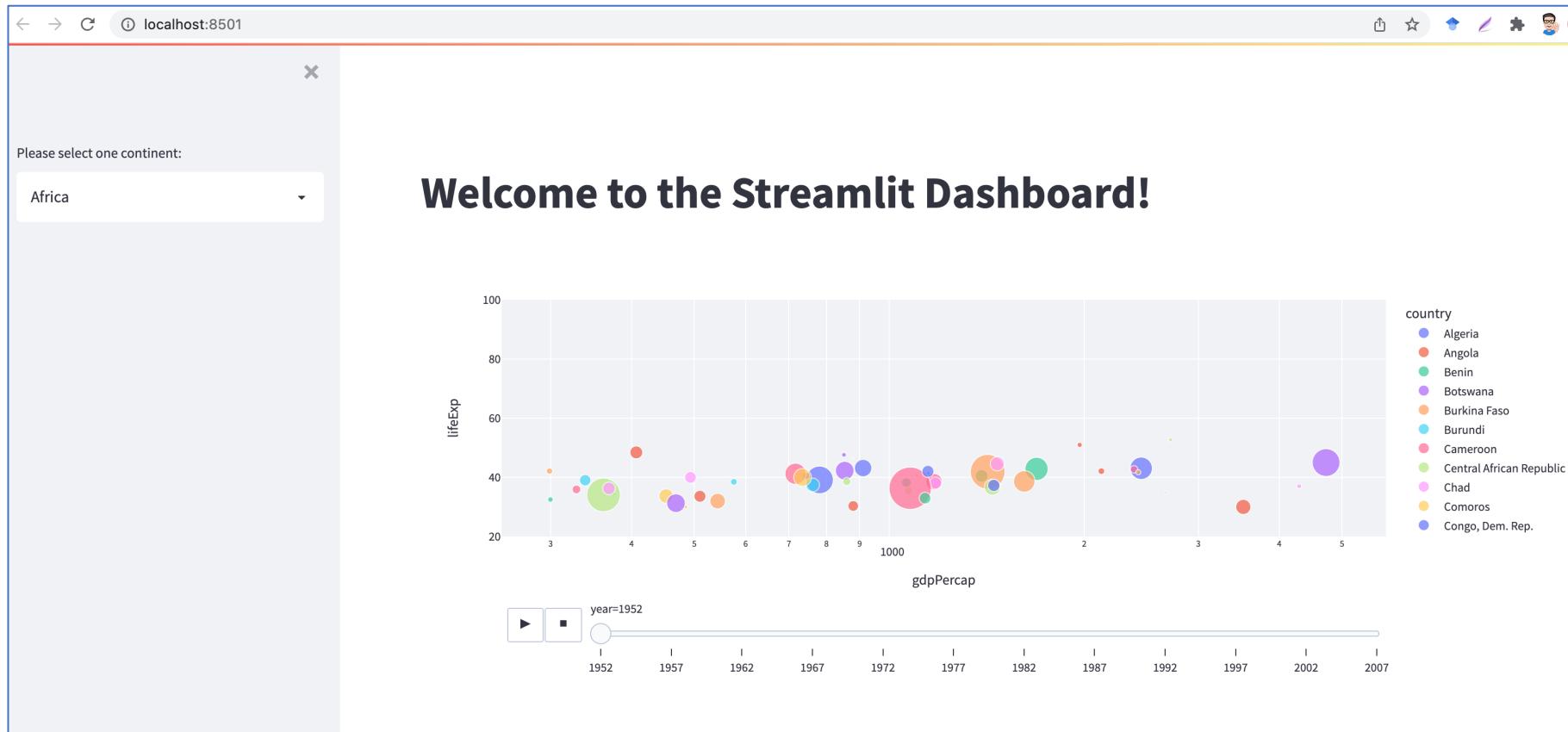
The figure displays four browser screenshots side-by-side, each showing a different Python dashboard tool:

- Plotly Dash:** A screenshot of the Plotly Dash documentation page. It features a dark header with the Plotly logo and a navigation bar for "Dash", "Low-Code Development", and "Deployment & Scaling". Below this is a section titled "Overview of Dash & Dash Apps" which describes Dash as a point-&-click interface for Python, R, and Julia. It also mentions Plotly's Dash Enterprise for business deployment.
- Panel:** A screenshot of the Panel documentation page. It has a dark header with the Panel logo. The main content area features the text "A high-level app and dashboarding solution for Python" and five thumbnail images of different dashboard examples: Attractors, Gapminders, NYC Taxi, Glaciers, and Portfolio Optimizer.
- Voila:** A screenshot of the Voila documentation page. It features a large teal "voila" logo. Below it are links for "docs", "passing", "launch", "binder", "gitter", and "join chat". A brief description states "Rendering of live Jupyter notebooks with interactive widgets." and "Introduction".
- Streamlit:** A screenshot of the Streamlit landing page. It has a dark header with the Streamlit logo. The main content area features the text "The fastest way to build and share data apps" and "Streamlit turns data scripts into shareable web apps in minutes. All in Python. All for free. No front-end experience required." Below this are two buttons: "Try Streamlit now" and "Sign up for Streamlit Cloud".

They are kind of equivalent to the Shiny package in R



Examples



Demo



Data Science Recap

- Data science is a teamwork, so choose the most suitable workflow for your project and your team.
- Getting data ready takes efforts, put deadlines for different deliverables.
- Spending time to plan the work will NOT slow you down
- Making sense of data is more an art than science!

