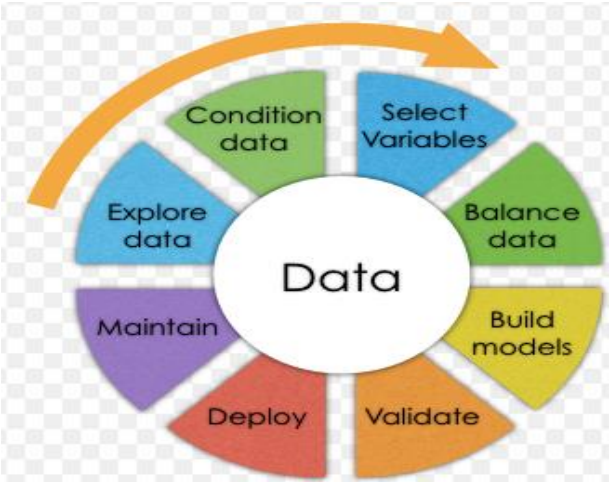
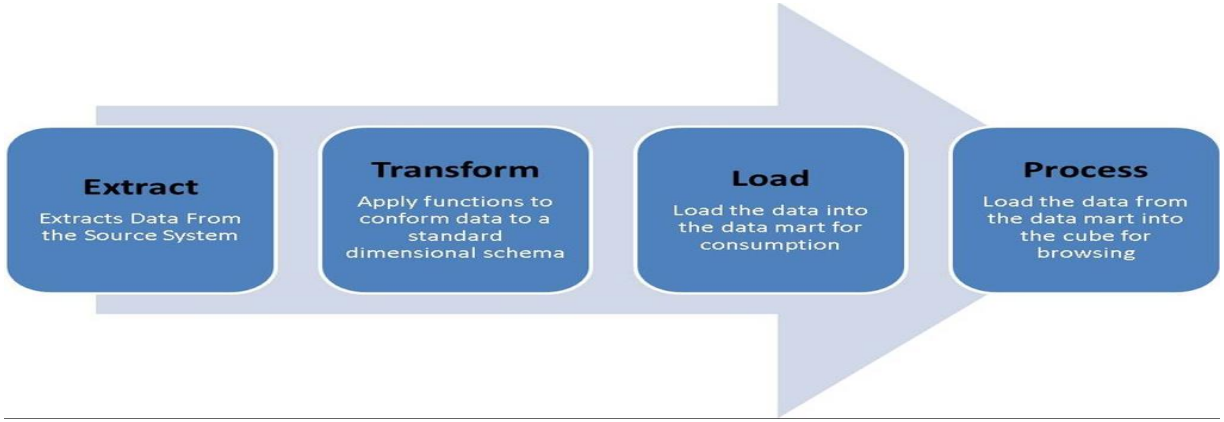
PACE UNIVERSITY

SEIDENBERG SCHOOL OF CSIS





Seidenberg School Of Computer Sciences | Pace University

Data Analytics Lifecycles - Data Science Workflows

The Evolution of Enterprise Analytics – Data Architecture

The **ETL/ELT** (Extraction, Load, Transform) Data Cycle

Prof. Tassos H. Sarbanes

CS675 --- Spring 2022

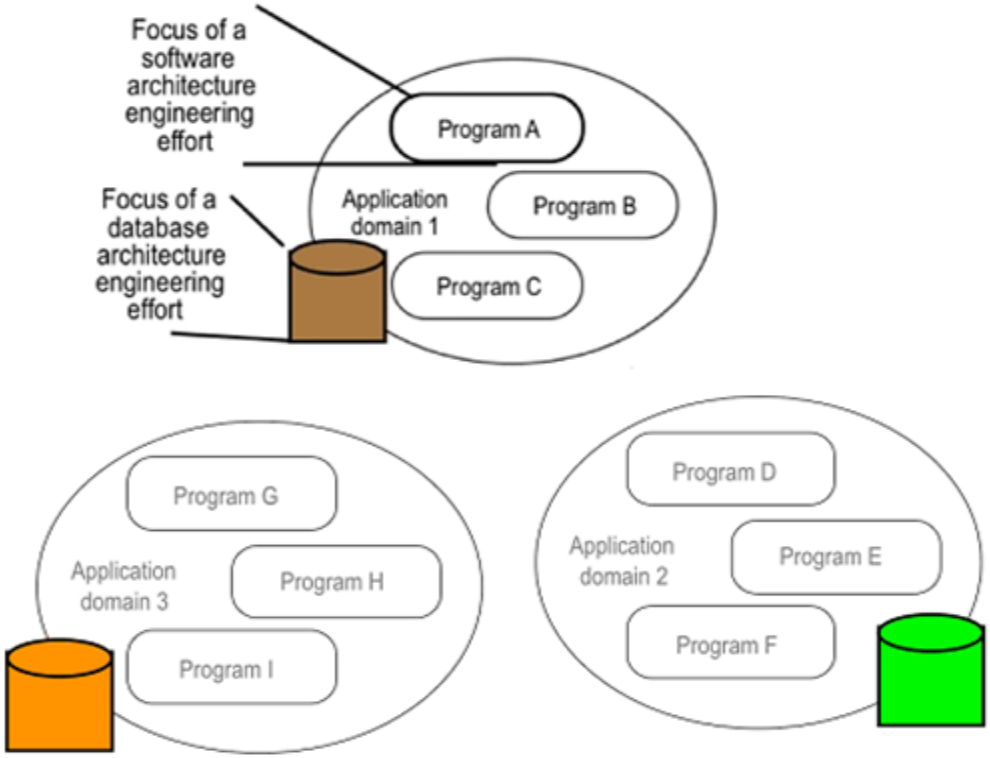
Typically Managed Organizational Architectures

- Process Architecture
  - Arrangement of inputs -> transformations = value -> outputs
  - Typical elements: Functions, activities, workflow, events, cycles, products, procedures
- Systems Architecture
  - Applications, software components, interfaces, projects
- Business Architecture
  - Goals, strategies, roles, organizational structure, location(s)
- Security Architecture
  - Arrangement of security controls relation to IT Architecture
- Technical Architecture/Tarchitecture
  - Relation of software capabilities/technology stack
  - Structure of the technology infrastructure of an enterprise, solution or system
  - Typical elements: Networks, hardware, software platforms, standards/protocols
- Data/Information Architecture
  - Arrangement of data assets supporting organizational strategy
  - Typical elements: specifications expressed as entities, relationships, attributes, definitions, values, vocabularies

Data Architecture - A Useful Definition

Common vocabulary expressing integrated requirements ensuring that **data assets** are stored, arranged, managed, and used in systems in **support of organizational strategy**.

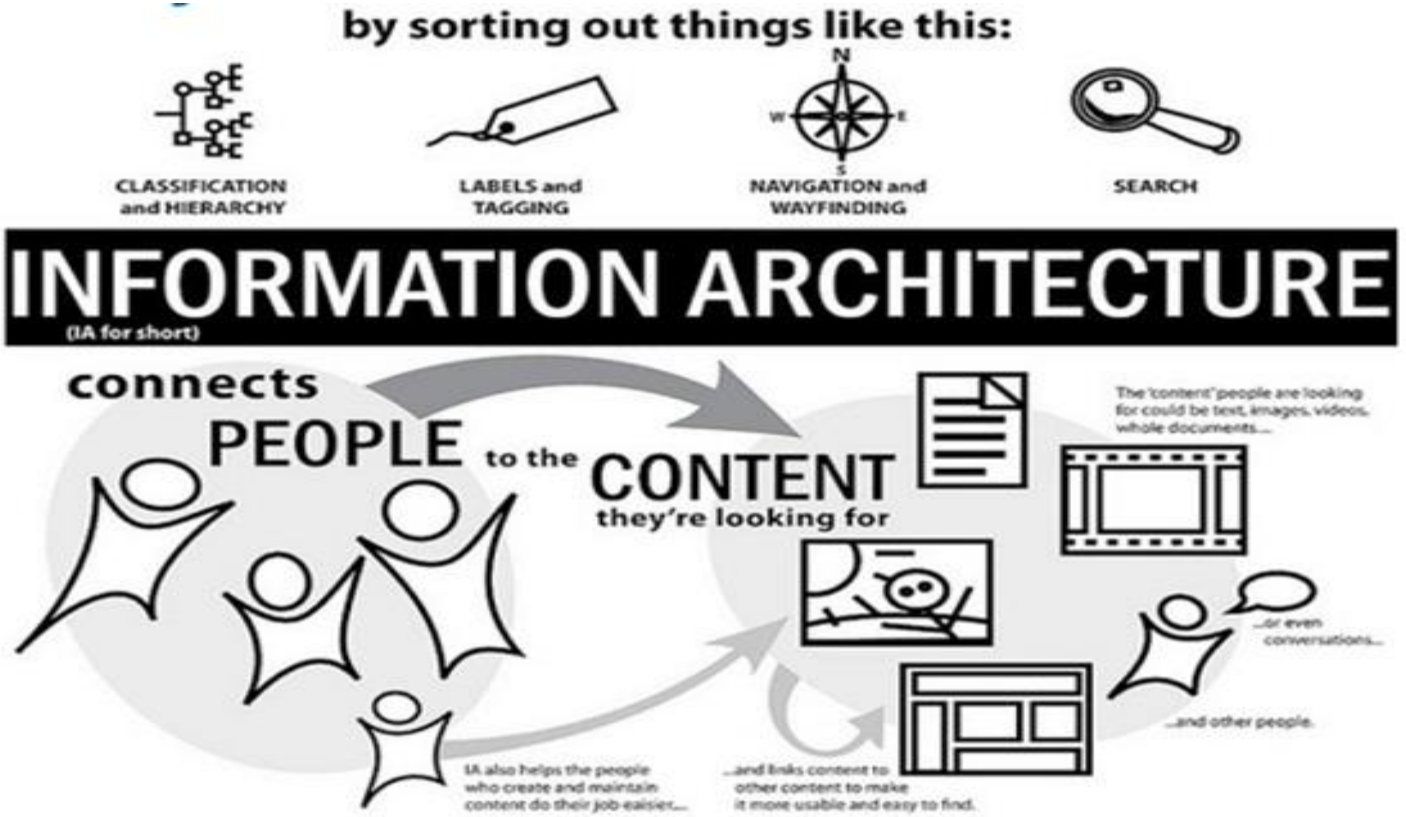
Database Architecture Focus



Database Architecture Focus has Greater Potential Business Value

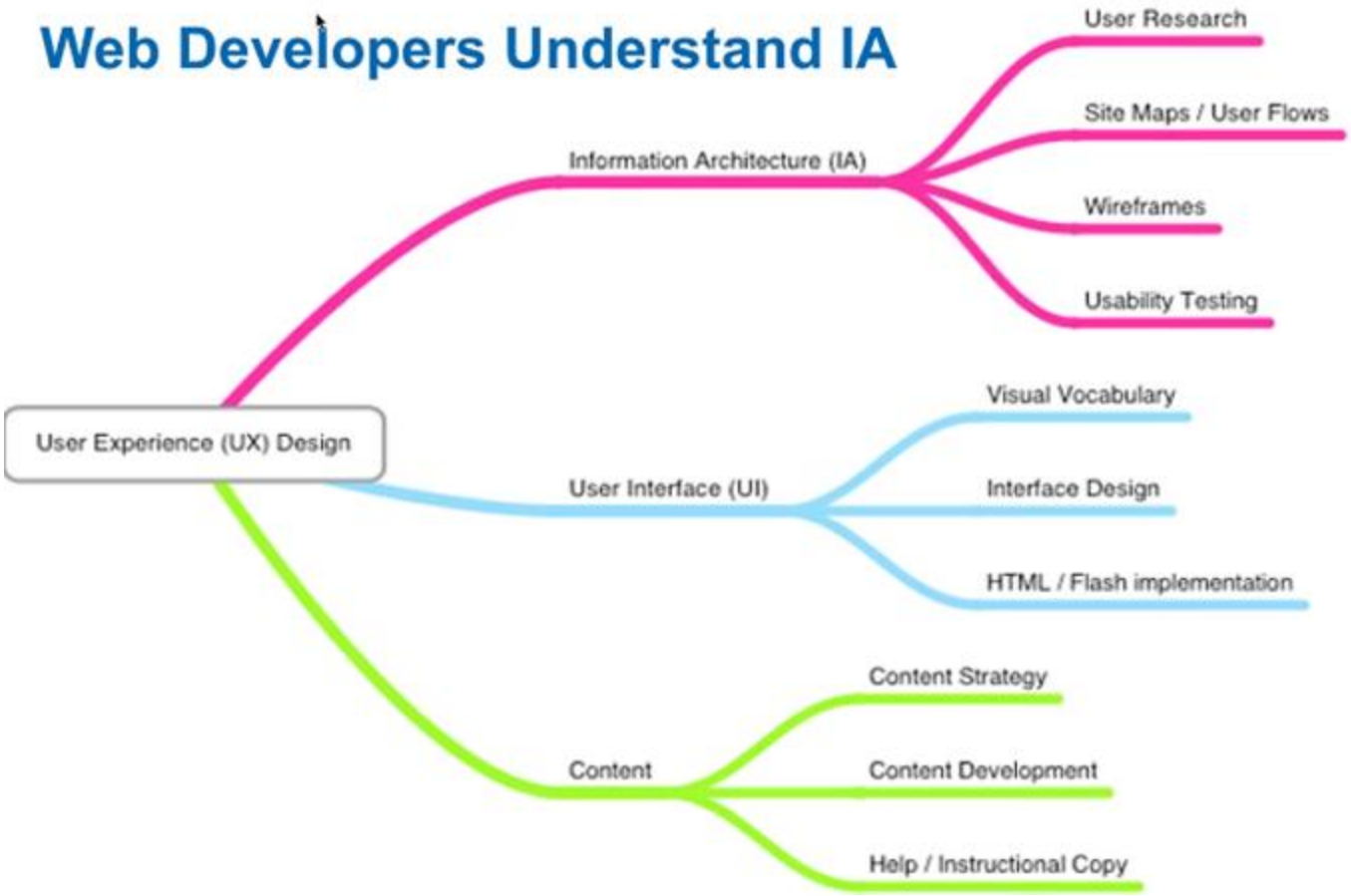
- Broader focus than either software architecture or database architecture
- Analysis scope is on the system wide use of data
- Problems caused by data exchange or interface problems
- Architectural goals more strategic than operational

What Do We Use a Data/Information Architecture (DA/IA) for?



Courtesy of Illustration by murdock23 @ http://designfestival.com/information-architecture-as-part-of-the-web-design-process/

Example of Information Architecture (IA)



Courtesy of http://www.jeffkerndesign.com

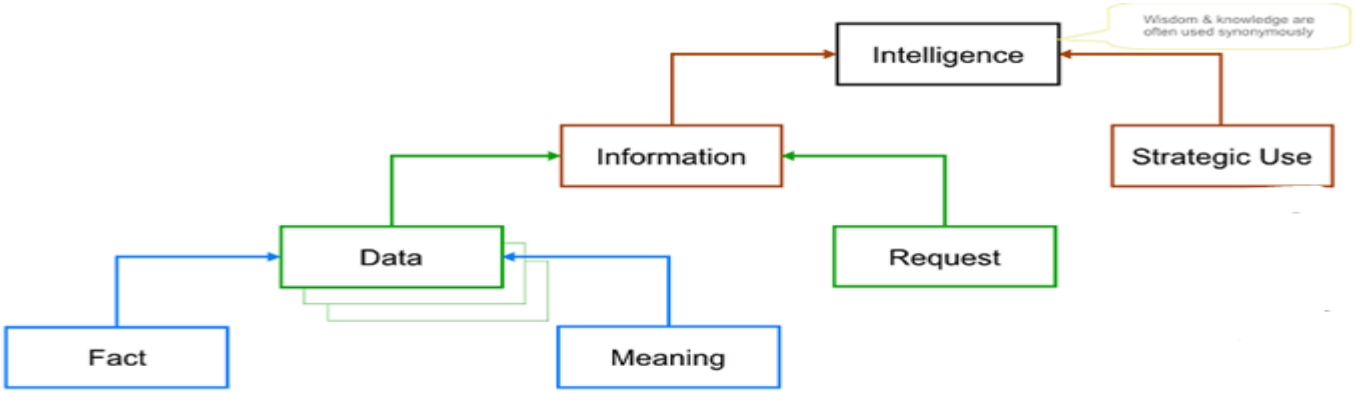
How are Data Models Expressed as Architectures?

- Attributes are organized into entities/objects
  - Attributes are characteristics of "things"
  - Entitles/objects are "things" whose information is managed in support of strategy
  - Examples
- Entities/objects are organized into models
  - Combinations of attributes and entities are structured to represent information requirements
  - Poorly structured data, constrains organizational information delivery capabilities
  - Examples
- Models are organized into architectures
  - When building new systems, architectures are used to plan development
  - More often, data managers do not know what existing architectures are and - therefore - cannot make use of them in support of strategy implementation
  - Why no examples?

Very Important: **Data must be Architected to Deliver Value**

(See next slide)

How Data Structures support Organizational Strategy



1. Each FACT combines with one or more MEANINGS
2. Each specific FACT and MEANING combination is referred to as a DATUM
3. An INFORMATION is one or more DATA that are returned in response to a specific REQUEST
4. INFORMATION REUSE is enable when one FACT is combined with more than one MEANING
5. INTELLIGENCE is INFORMATION associated with its STRATEGIC USES
6. DATA/INFORMATION must formally arranged into an ARCHITECTURE

Data → Information → Insight → Intelligence → Decision Making

This path is followed by Data Science and Data Analytics.

Data: An individual unit that contains **raw**, unprocessed material which does not carry any specific meaning. Data is the basic building block of research.

Information: A processed, organized group of **data** that collectively carry a logical meaning.

Insight: A significant **information** to an organization. It is the value obtained through the use of analytics. Actionable insights is the link for organizations (data-driven) that want to drive business outcomes from data.

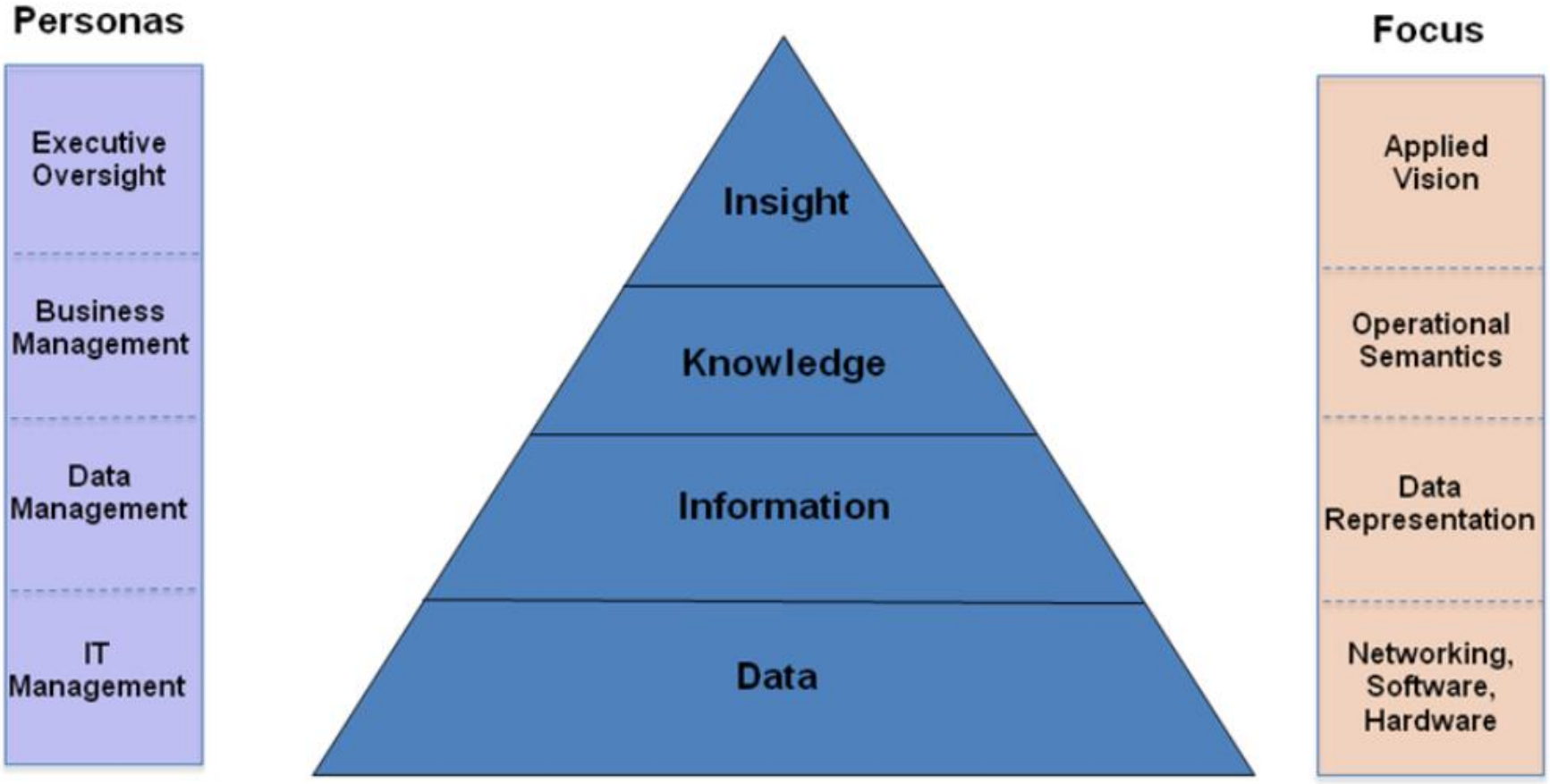
Intelligence: A deep-dive **study of data** to produce actionable insights..

Decision Making: An action taken by an organization based on **intelligence** gathering and actionable **insights**.

Example

Data: (255,0,0) (The RGB code for red color)
Information: Traffic light at the intersection Broadway & Wall Street has turned red.
Intelligence: Traffic light ahead of me (getting near the intersection) has turned red.
Decision Making: I must stop the car.

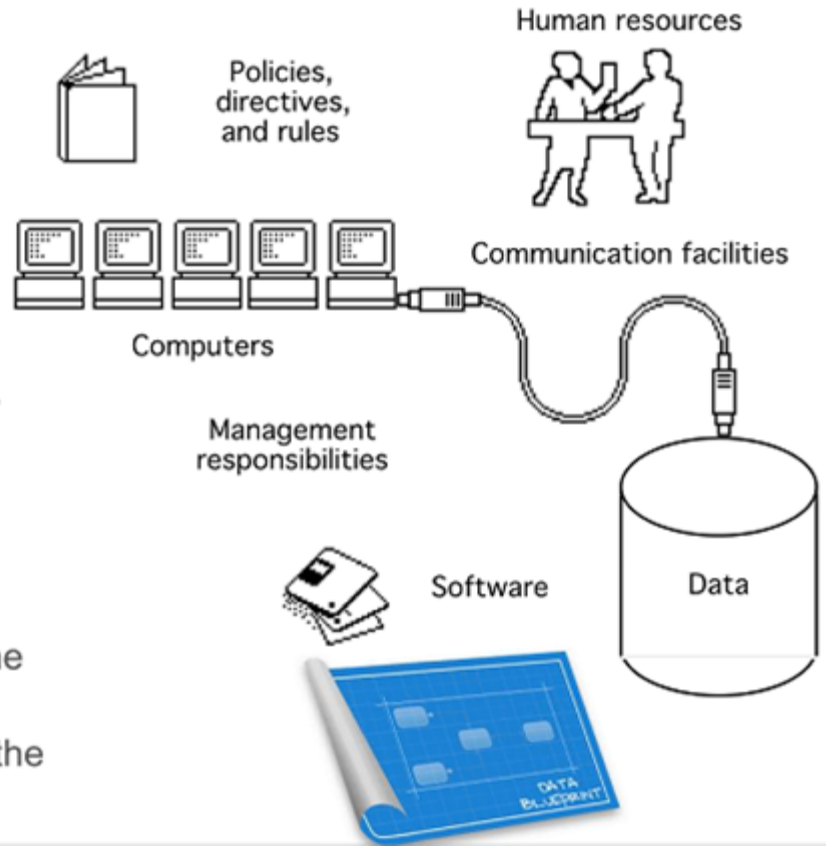
The Information Governance Maturity Model



- Source: https://ibmecmblog.wordpress.com

What Questions Can Data Architectures Address?

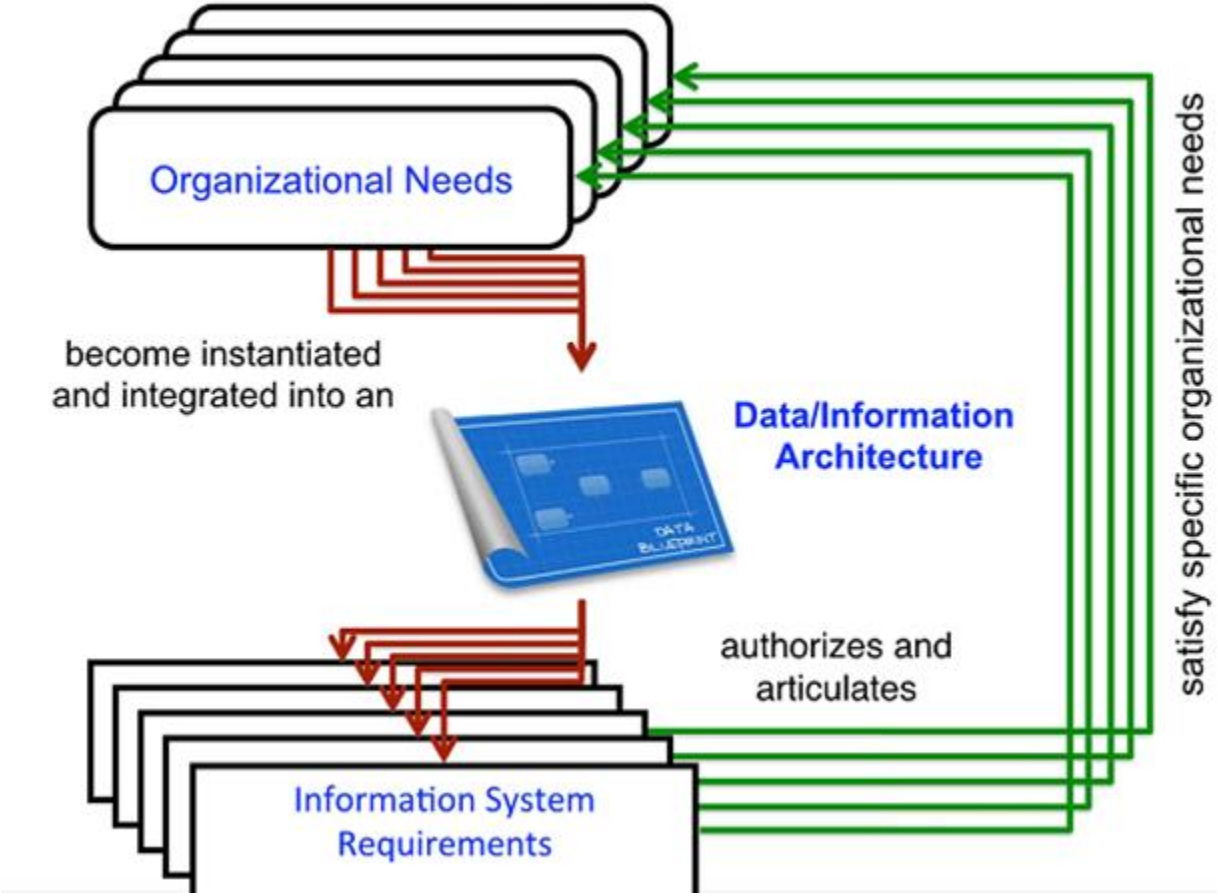
- How and why do the data components interact?
- Where do they go?
- When are they needed?
- Why and how will the changes be implemented?
- What should be managed organization-wide and what should be managed locally?
- What standards should be adopted?
- What vendors should be chosen?
- What rules should govern the decisions?
- What policies should guide the process?



Data Architectures produce and are made up of Information Models that are developed in response to Organization Needs.

(See next slide)

Data Architecture fulfill Organizational Needs



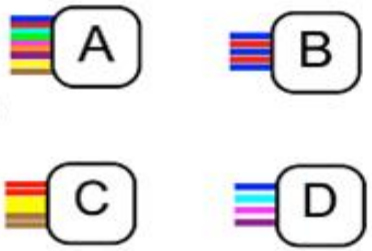
Data Leverage

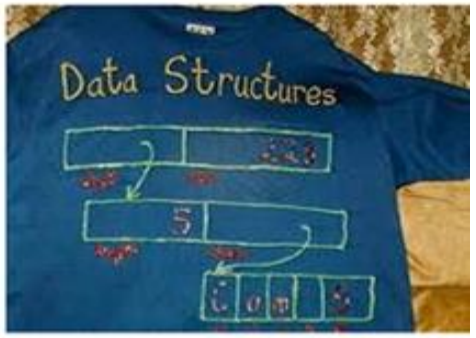
Permits organizations to **better manage** their sole non-'depletable', non-degrading, durable, strategic **asset-data**.

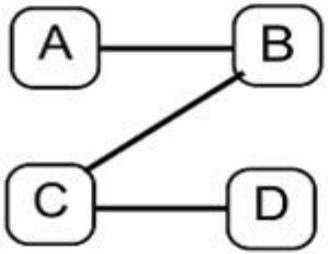
Treating data more asset-like simultaneously

How are Data Structures Expressed as Architectures?

- Details are organized into larger components
- Larger components are organized into models
- Models are organized into architectures



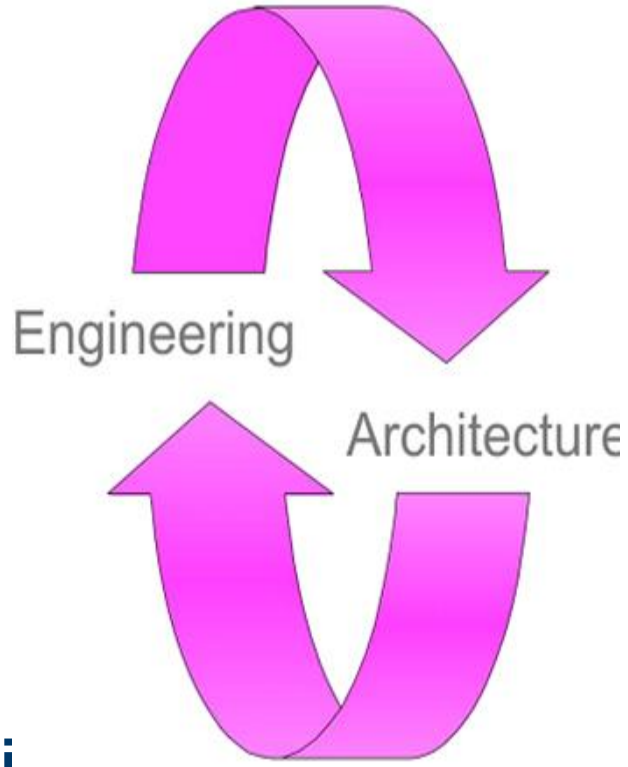




Details (Data) -→ Larger components (Data Structures) -→ Models -→ Architectures

Why Architect Data? (Need for Blue Prints!)

Would you build a house without an architecture sketch?	Model is the sketch of the system to be built in a project.
Would you like to have an estimate how much your new house is going to cost?	Your model gives you a very good idea of how demanding the implementation work is going to be!
If you hired a set of constructors from all over the world to build your house, would you like them to have a common language?	Model is the common language for the project team.
Would you like to verify the proposals of the construction team before the work gets started?	Models can be reviewed before thousands of hours of implementation work will be done.
If it was a great house, would you like to build something rather similar again, in another place?	It is possible to implement the system to various platforms using the same model.
Would you drill into a wall of your house without a map of the plumbing and electric lines?	Models document the system built in a project. This makes life easier for the support and maintenance!



Engineering / Architecting Relationship

- **Architecting** is used to create & build systems too complex to be treated by engineering analysis alone
- **Architects** require technical details as the exception
- **Engineers** develop the technical designs
- Example: Craftsman deliver components supervised by:
  - Building Contractor
  - Manufacturer

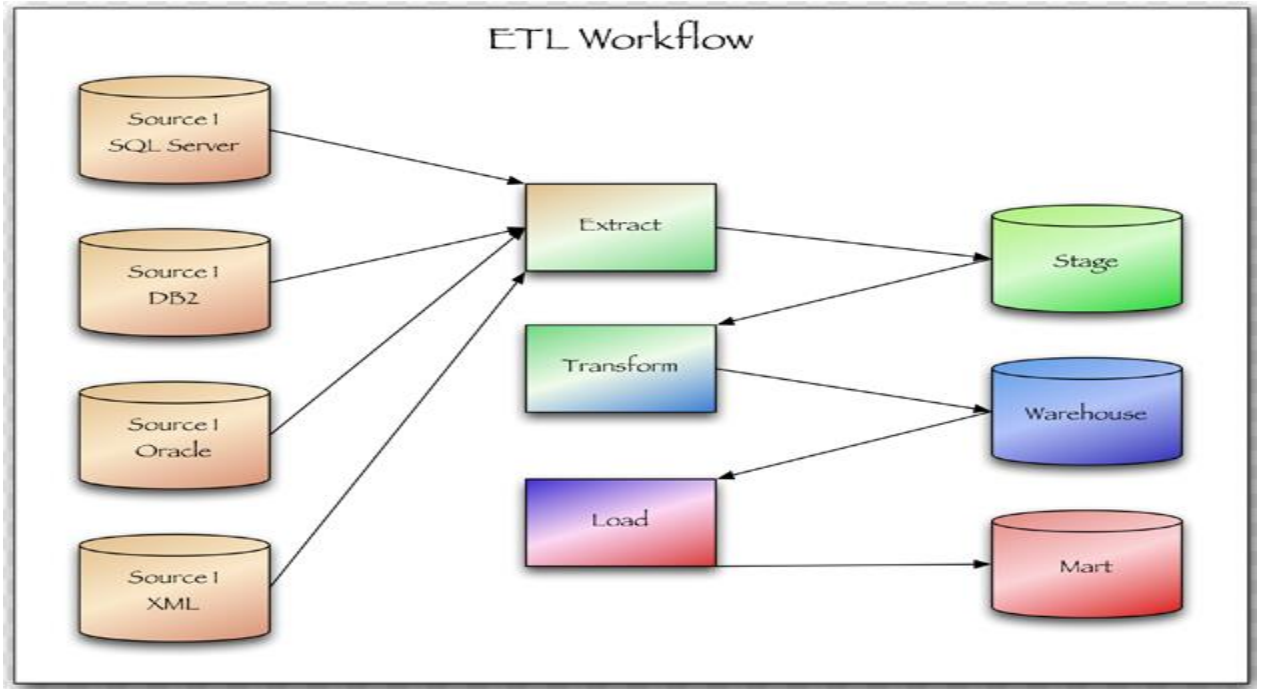
Data Architect – Data Engineer – Data Scientist

Data architect is a practitioner of data architecture, a data management discipline concerned with designing, creating, deploying and managing an organization's data architecture. Data architects define how the data will be stored, consumed, integrated and managed by different data entities and IT systems, as well as any applications using or processing that data in some way.

Data engineer gathers and collects the data, stores it, does batch processing or real-time processing on it, and serves it via an API to a data analyst/scientist who can easily query it. Data engineer provides the consolidated Big/Small/Fast data to the data analyst/scientist, so that the latter can analyze it.

Data Scientist is a scientist employed to analyze and interpret complex digital data, such as the usage statistics of a website, especially in order to assist an organization in its decision-making.

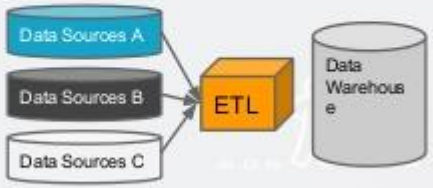
ETL (Extraction Transformation Load) Process



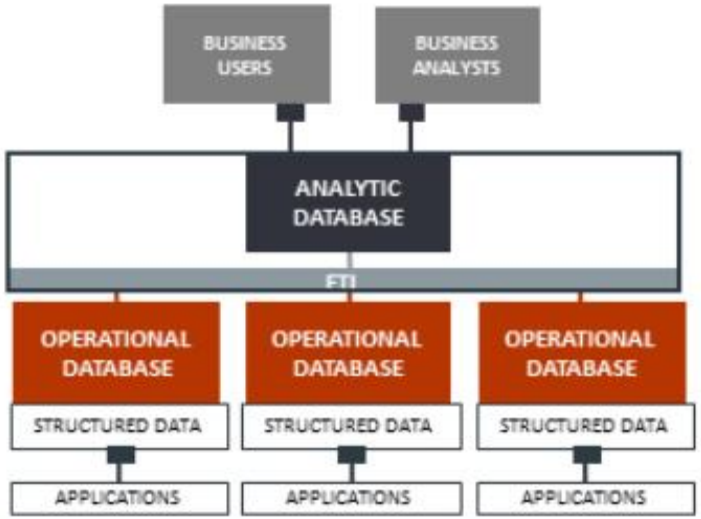
What is ETL?

In computing, **extract, transform, and load** (**ETL**) refers to a process in database usage and especially in data warehousing that:

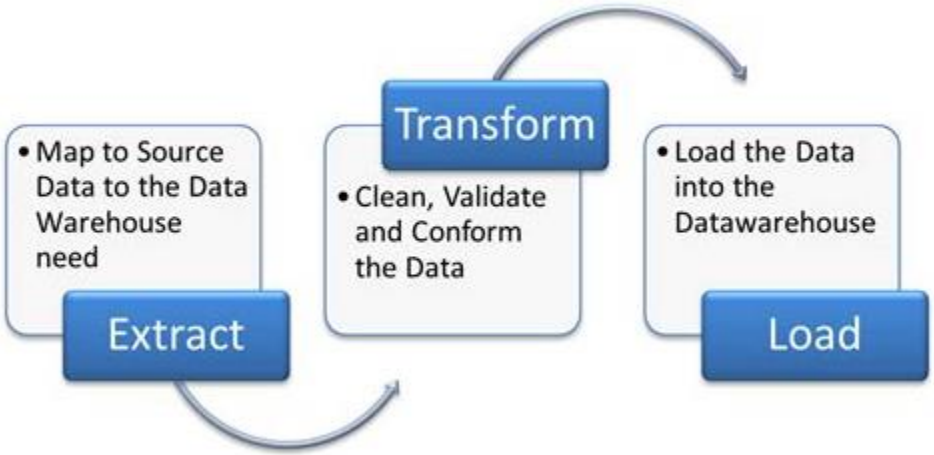
- Extracts data from outside sources
- Transforms it to fit operational needs, which can include quality levels
- Loads it into the end target (database, more specifically, operational data store, data mart, or data warehouse)



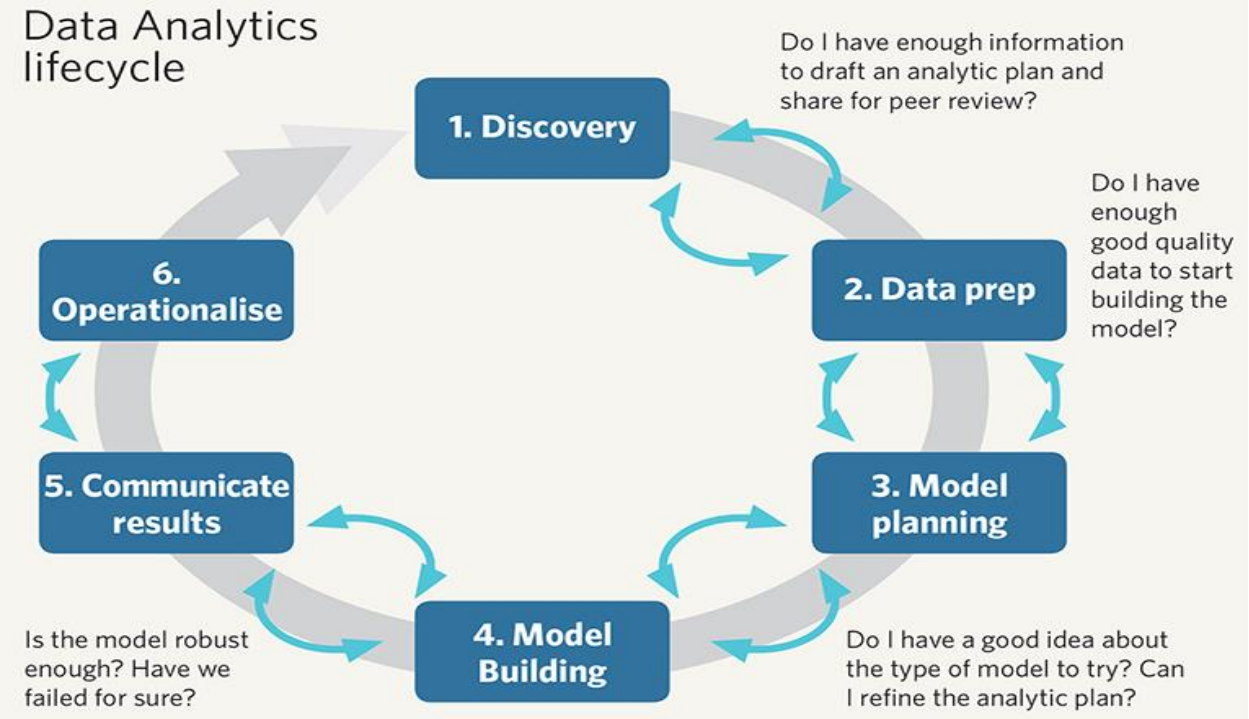
Data Extraction, Transformation, and Loading



- Traditionally, data integration technology served to enable the extraction, transformation and loading (ETL) of data from the operational database into the analytic database, as well as the integration with reporting and analysis tools
- Can be done in-database via loading and scripting tools
- More typically done via specialist ETL tools
- Often residing on the same physical server

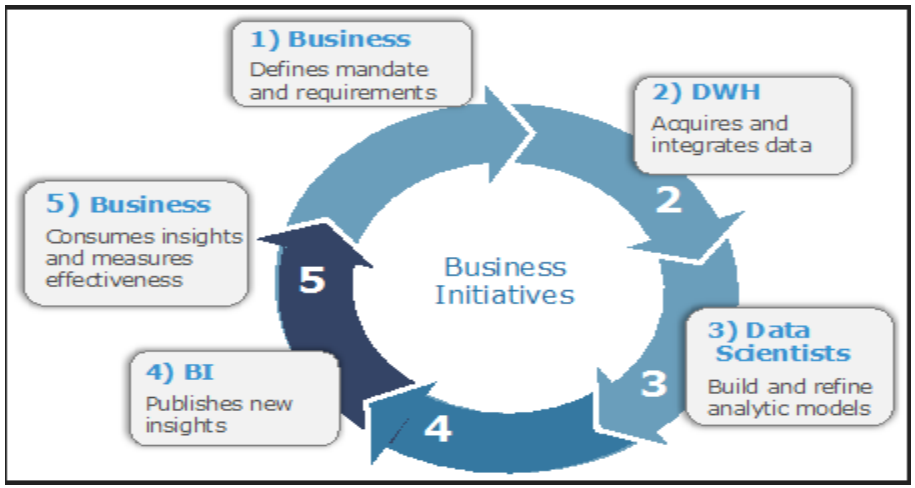


Data Analytics Lifecycle



From **IT** point of view

From **Business** point of view



What is Data Science - Definitions

- **Business Intelligence** – an umbrella term that includes the applications, infrastructure & tools, and best practices that enable access to and analysis of information to improve and optimize decisions & performance
- **Analytics** – it has emerged as a catch-all term for a variety of different BI and application-related initiatives. For some, it is the process of analyzing information from a particular domain, (e.g., website analytics). For others, is applying breadth of BI capabilities to a specific content area (sales, service, supply chain,...)
- **Data Science** – (grouped in w/ Advanced Analytics definition) the autonomous or semi-autonomous examination of **data** or content using sophisticated techniques and tools, beyond those of traditional BI, to discover deeper **insights**, make **predictions**, or generate **recommendations**

  Advanced analytic techniques include those such as data/text mining, ML, pattern matching, forecasting, predicting, visualization, semantic analysis, network and cluster analysis, multivariate statistics, graph analysis, simulation, complex event processing (CEP), neural networks.

Introduction to Data Science

More and more organizations these days **use their data**, a decision supporting tool to build data-intensive/focus **products** and **services**. Collection of skills required by organizations to support these functions grouped under term "**Data Sciences**". This course will cover the basic concepts of big data, methodologies for analyzing structured and unstructured data w/ emphasis on the **relationship between the Data Scientist and the Business needs**.

Data is an asset/value!

- Businesses need a mechanism in place that delivers insight.
- It is the analytics that get you there.
- Analytics require data (Big, Fast, Small, Smart)
- The more data we have the more interesting the insights can be.
- The Analytic Platforms today enable that.

What is a Data Science Workflow?

A **workflow is the definition, execution, and automation of business processes** toward the goal of **coordinating tasks and information between people and systems**.
In software development, standard processes like planning, development, testing, integration, and deployment, workflows that link them have evolved over decades. Data science is a young field so its processes are still in flux.

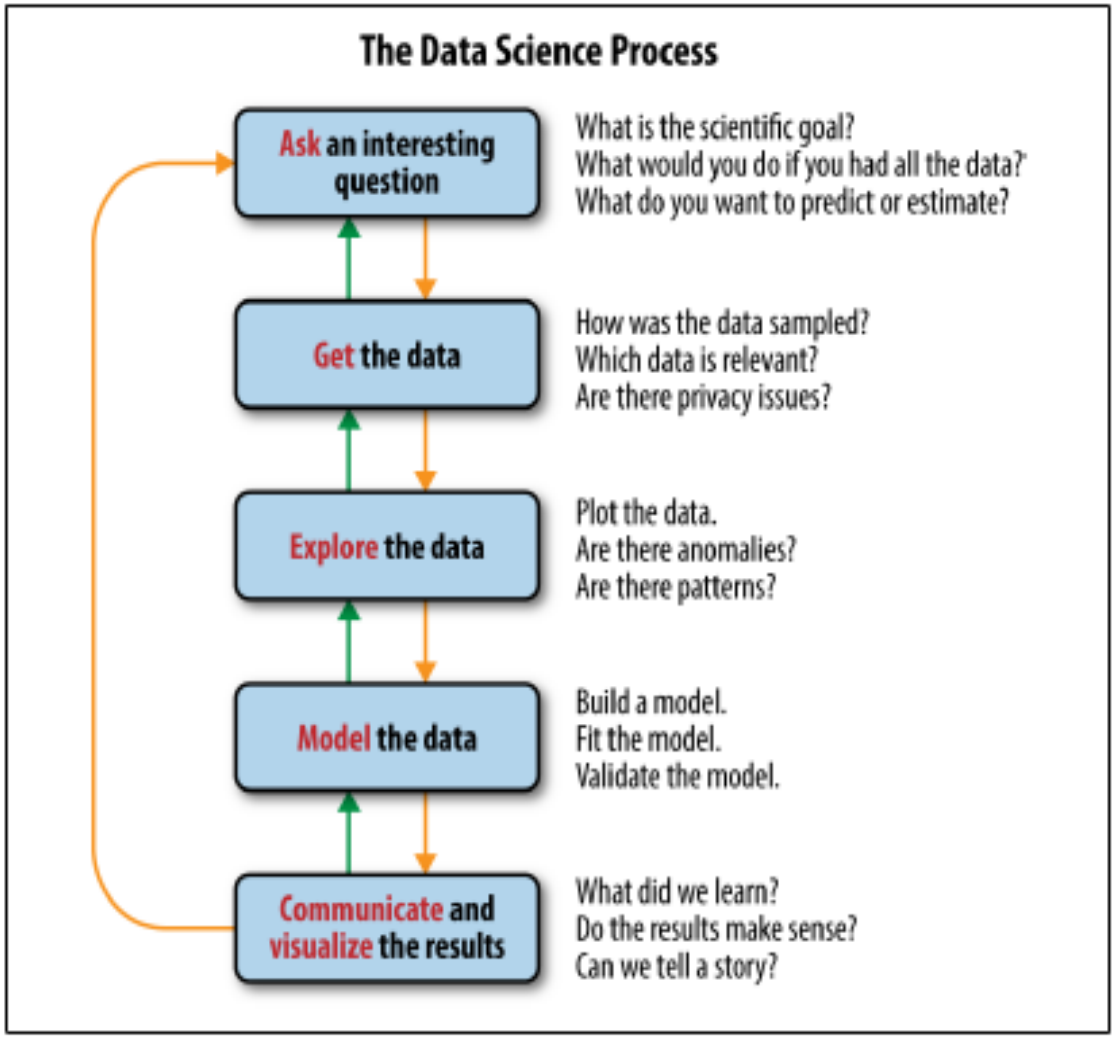
A **good workflow** for a particular team depends on the tasks, goals, and values of that team, whether they want to make their work faster, more efficient, correct, compliant, agile, transparent, or reproducible.

A tradeoff often exists between different goals and values—
- Do I want to get something done quickly (prototype) or
- Do I want to invest time now to make sure that it can be done quickly next time?

Some 'basic' **Metrics**
- Build a MVP (Minimum Viable Product) version
- Produce Results Fast (Scale)
- Reproduce and Reuse Results
- Audit Results

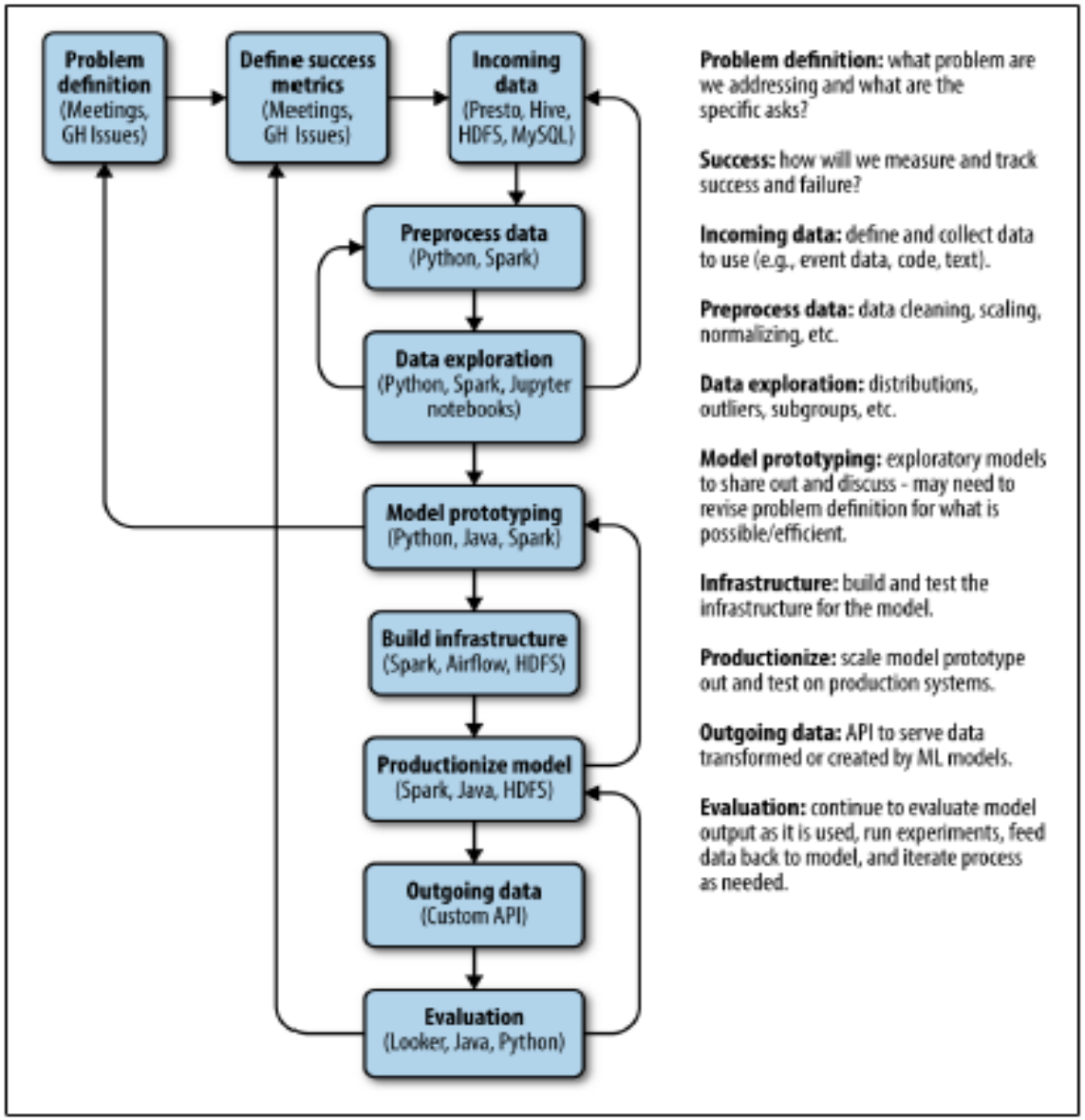
The Data Science Process / Workflow

There is no universally agreed upon data science process.
The introductory data science course at Harvard uses the following basic process:



A representation of the data science process (courtesy of Joe Blitzstein and Hanspeter Pfister, created for the Harvard data science course)

The Data Science Process at GitHub



At the beginning of every project, GitHub's machine learning team defines not just the problem or question it addresses, but what the **success metrics** should be.

Defining a **success measure** that makes sense to both the business and the data science team can be a challenge.

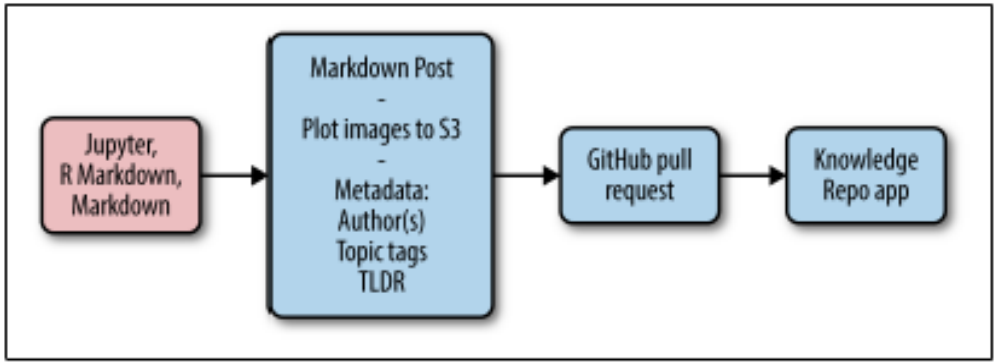
If that metric decouples from the business objective, the business feels that the data science team doesn't deliver, or the technical team creates models & reaches conclusions that might not be valid.

Data Science Workflow – Reuse Knowledge

As data science teams expand and mature, knowledge-sharing within data science teams and across the entire organization becomes a growing challenge.
In scientific research, examining previous relevant work on a topic is a basic prerequisite to doing new work.

Issue: No formal processes/tools exist within organizations to discover prior data science work

Airbnb's Knowledge Repo: attempt to make data science work discoverable, not just by the data science team, but by the entire company.
Posts written in JNBs, R markdown files, or in plain markdown are committed to a Git repository. A web app renders the repository's contents as an internal blog

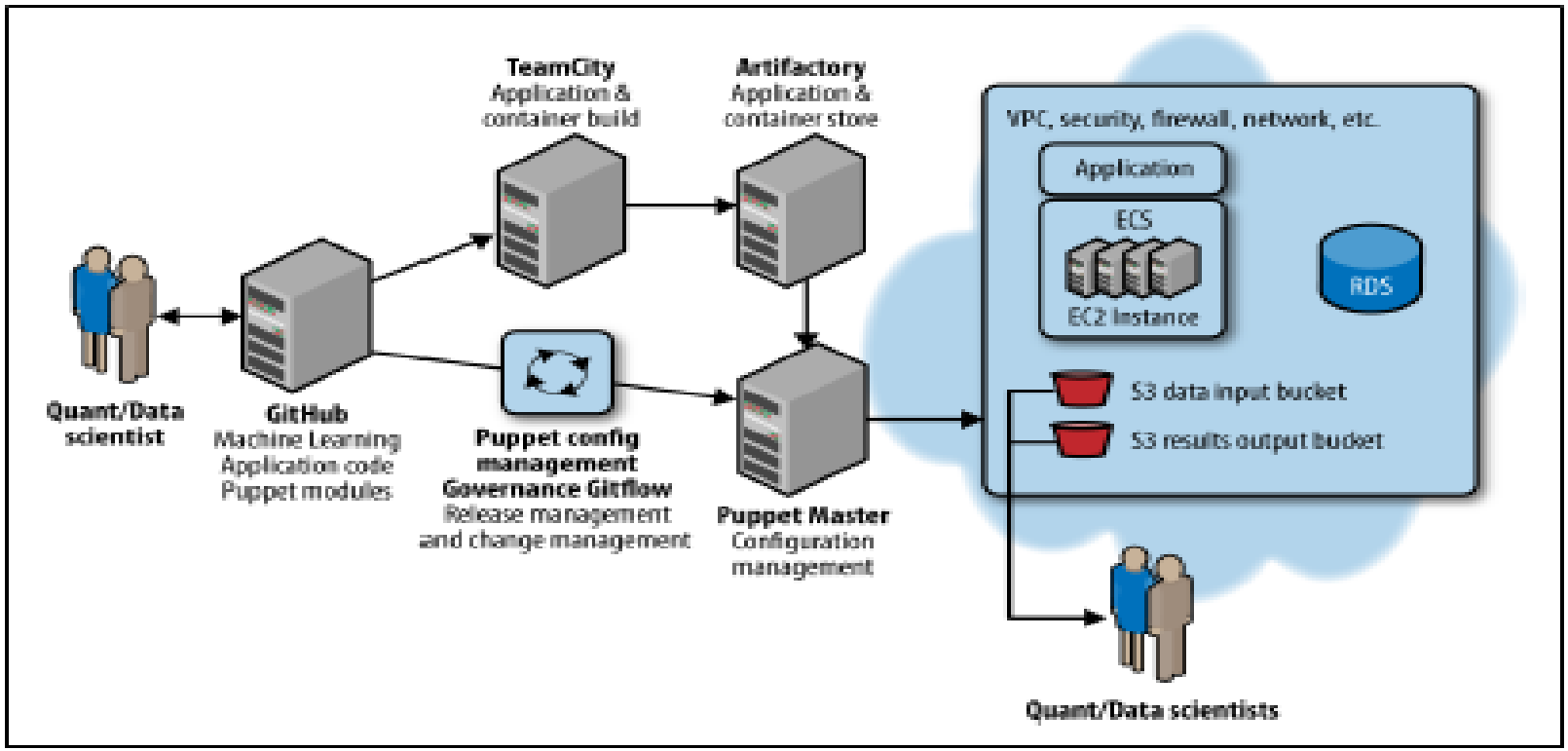


Airbnb's Knowledge Repo

Data science work at Airbnb is now discoverable via a full-text search over the synopsis, title, author, and tags.

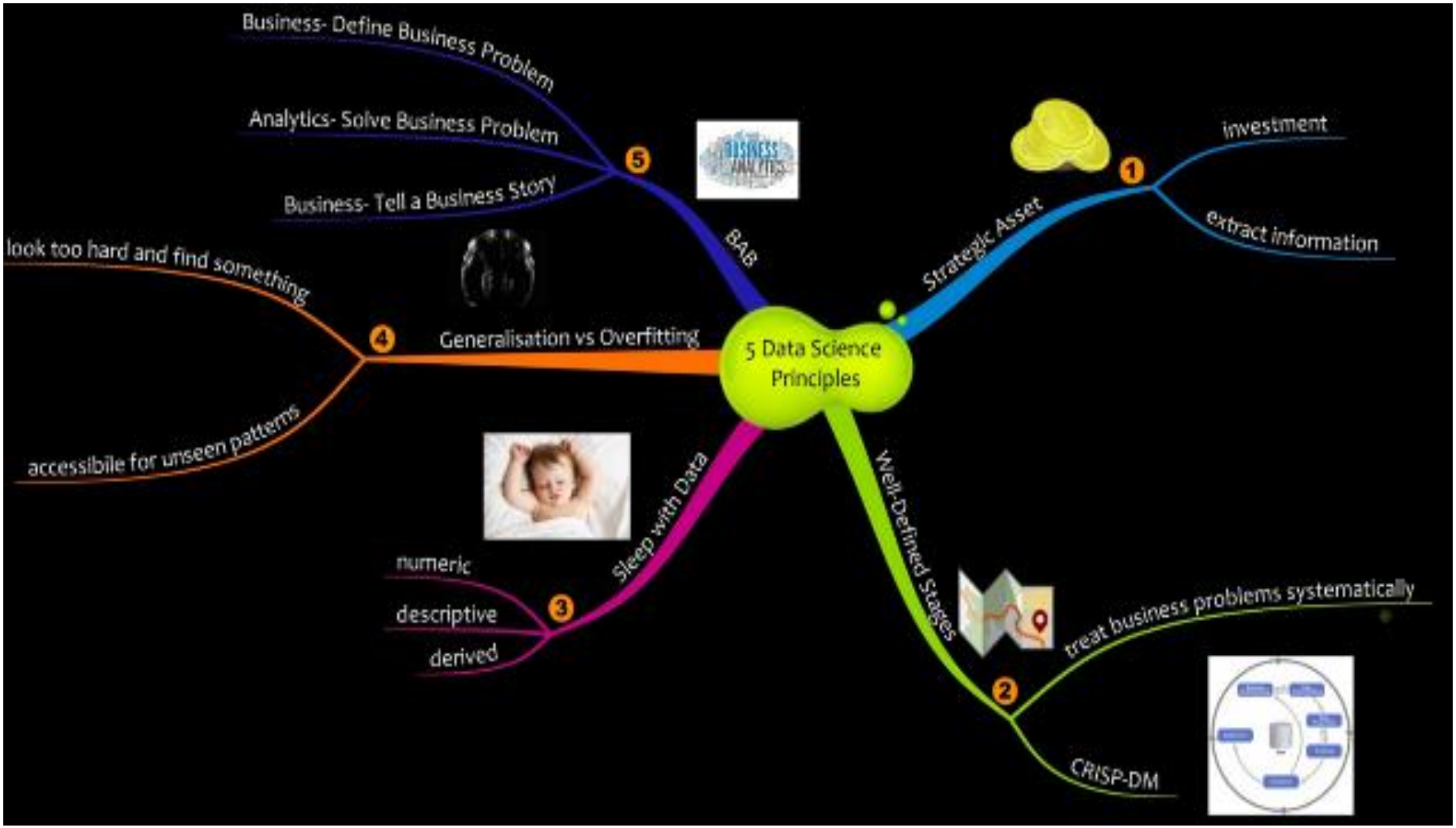
"Productionize" Data Science

Data Science's entire development and **deployment process must conform to strict compliance and security requirements and must be fully auditable**. At the same time, the process has to be extremely **efficient and easy to use** for data scientists as well as others!



Scotiabank's automated deployment system

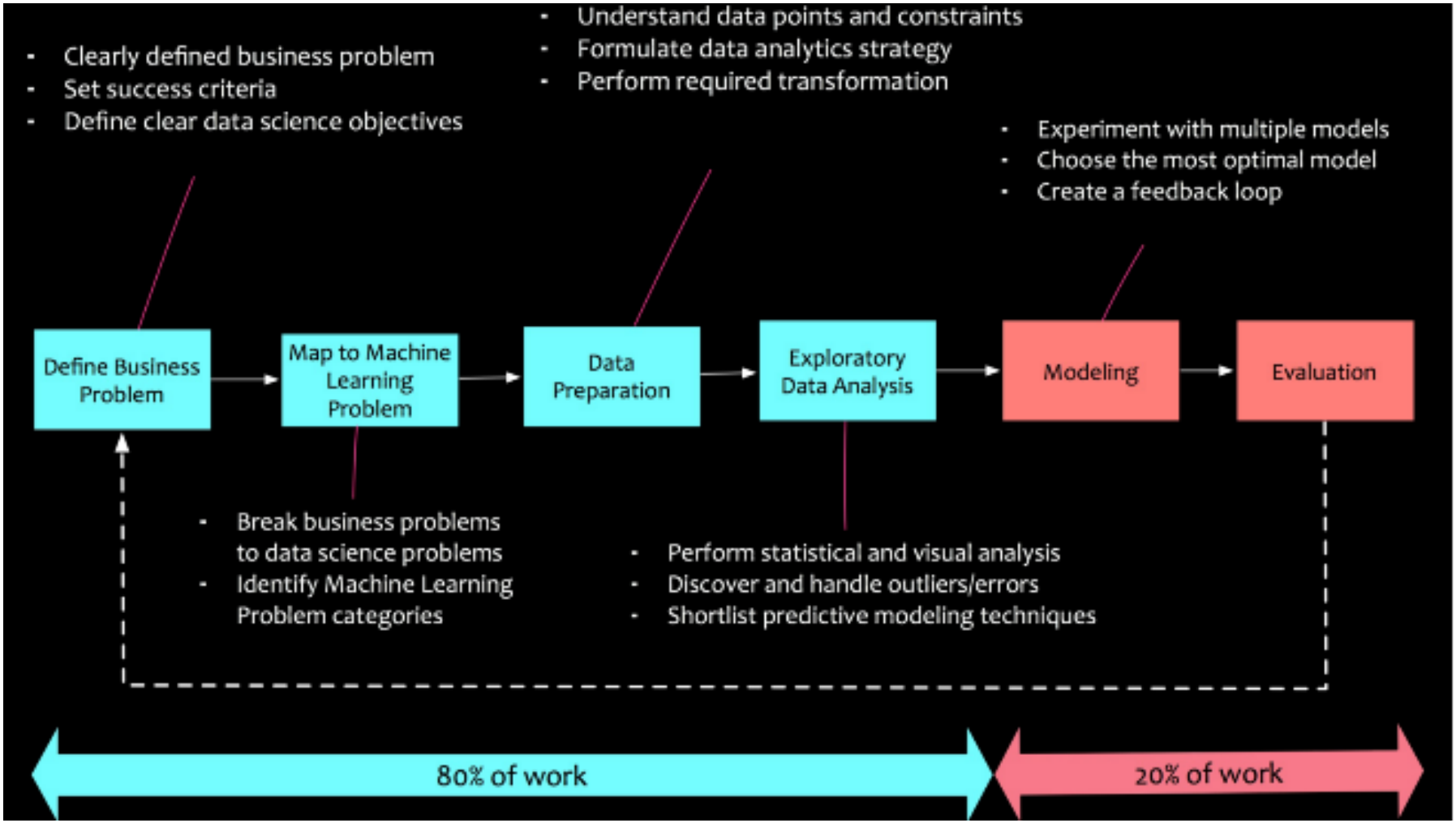
Data Science: Key Principles



Data Science: Key Principles … cont'd

- **Data is a strategic asset**: This concept is an organizational mindset. The question to ask is: "Are we using the all the data asset that we are collecting and storing? Are we able to extract meaningful insights from them?". I'm sure that the answers to these questions are "No". Companies that are cloud born are intrinsically data-driven. It is in their psyche to treat data as a strategic asset. This mindset is not valid for most of the organization.
- A systematic process for **knowledge extraction**: A methodical process needs to be in place for extracting insights from data. Process should have clear and distinct stages with clear deliverables. Cross Industry Standard Process for Data Mining (CRISP-DM) is such process.
- Sleeping with the data: Organizations need to invest in people who are passionate about data. **Transforming data into insight** is not alchemy. There are no alchemists. They need evangelists who **understand the value of data**. They need evangelists who are data literate and creative. They need folks who can **connect data, technology, and business**.
- Embracing uncertainty: Data Science is not a silver bullet. It is not a crystal ball. Like reports and KPIs, it is a decision enabler. Data Science is a tool and not a means to end. It is not in the realm of absolute. It is in the realm of probabilities. Managers and decision makers need to embrace this fact. They need to embrace ***quantified uncertainty*** in their decision-making process. Such uncertainty can only be entrenched if the organizational culture adopts a fail fast-learn fast approach. Only thrive if organizations choose a culture of experimentation.
- The **BAB principle**: I perceive this as the most important principle. The focus of a lot of Data Science literature is on models and algorithms. The equation is devoid of business context. **Business-Analytics-Business (BAB)** is the principle that emphasizes the business part of the equation. Putting them in a business context is pivotal. Define the business problem. Use analytics to solve it. Integrate the output into the business process. BAB.

Data Science: Process



Data Science: Process … cont'd (1)

Following are the stages of a typical data science project:

1. **Define Business Problem**

Albert Einstein once quoted "Everything should be made as simple as possible, but not simpler". This quote is the crux of defining the business problem. Problem statements need to be developed and framed. Clear success criteria need to be established. Business teams are too busy with their operational tasks at hand. It doesn't mean that they don't have challenges that need to be addressed. Brainstorming sessions, workshops, and interviews can help to uncover these challenges and develop hypotheses. Example: Let us assume that a telco company has seen a decline in their year-on-year revenue due to a reduction in their customer base. In this scenario, the business problem may be defined as:

"Company needs grow customer base by targeting new segments & reducing customer churn."

2. **Decompose To Machine Learning Tasks**

The business problem, once defined, needs to be decomposed to machine learning tasks. Let's elaborate on the example that we have set above. If the organization needs to grow our the customer base by targeting new segments and reducing customer churn, how can we decompose it into machine learning problems? Following is an example of decomposition:

- Reduce the customer churn by x %.
- Identify new customer segments for targeted marketing.

3. **Data Preparation**

Having 1 & 2, need to dive deeper into the data. Data understanding should be explicit to the problem at hand. It should help us with to develop right kind of strategies for analysis. Key things to note is the source of data, quality of data, data bias, etc.

Data Science: Process … cont'd (2)

4. **Exploratory Data Analytics (EDA)**

A cosmonaut traverses through the unknowns of the cosmos. Similarly, a data scientist traverses through the unknowns of the patterns in the data, peeks into the intrigues of its characteristics and formulates the unexplored. Exploratory data analysis (EDA) is an exciting task. We get to understand the data better, investigate the nuances, discover hidden patterns, develop new features (feature engineering) and formulate modeling strategies.

5. **Modeling**

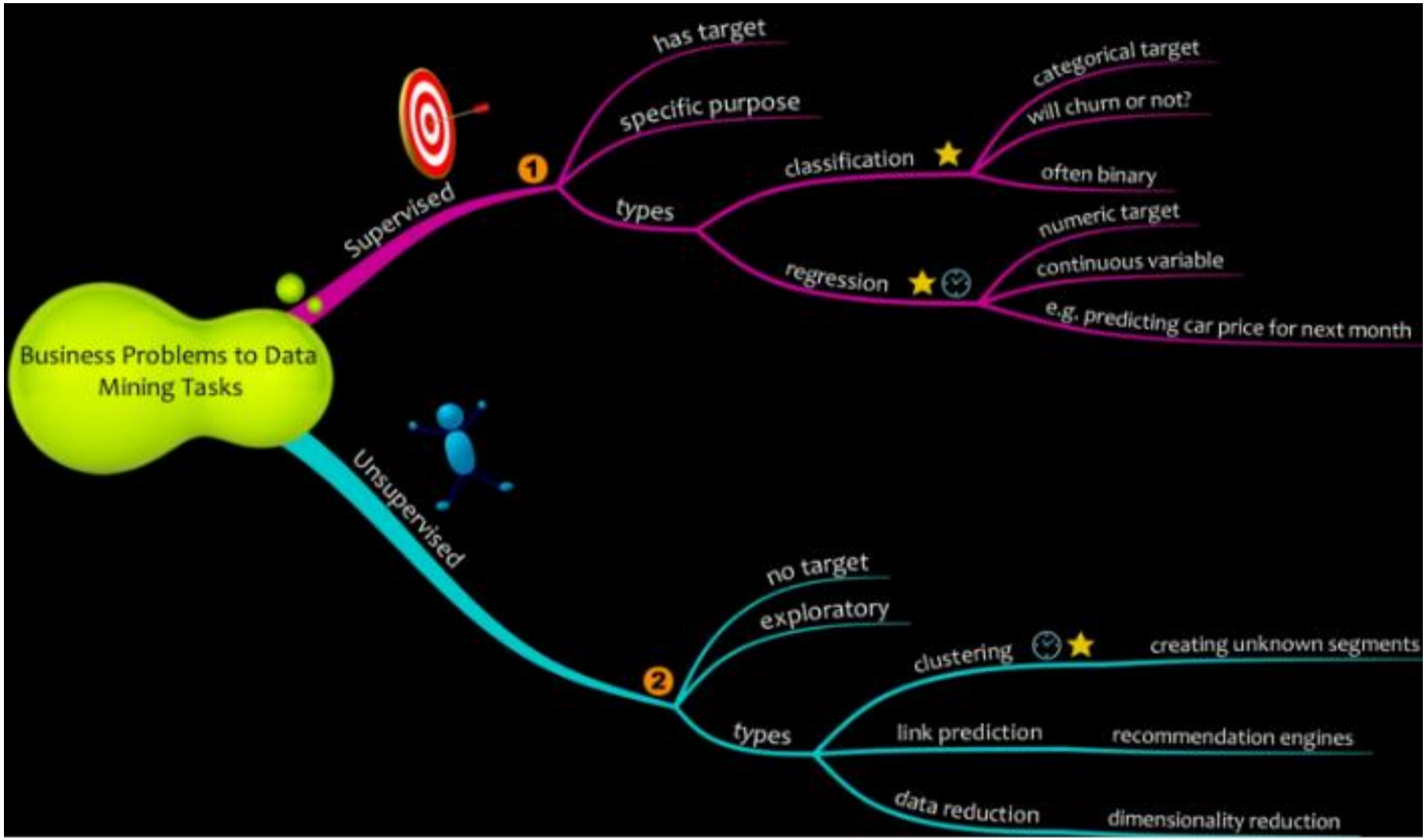
After EDA, we move on to the modeling phase. Here, based on our specific machine learning problems, we apply useful algorithms like regressions, decision trees, random forests, etc.

6. **Deployment and Evaluation**

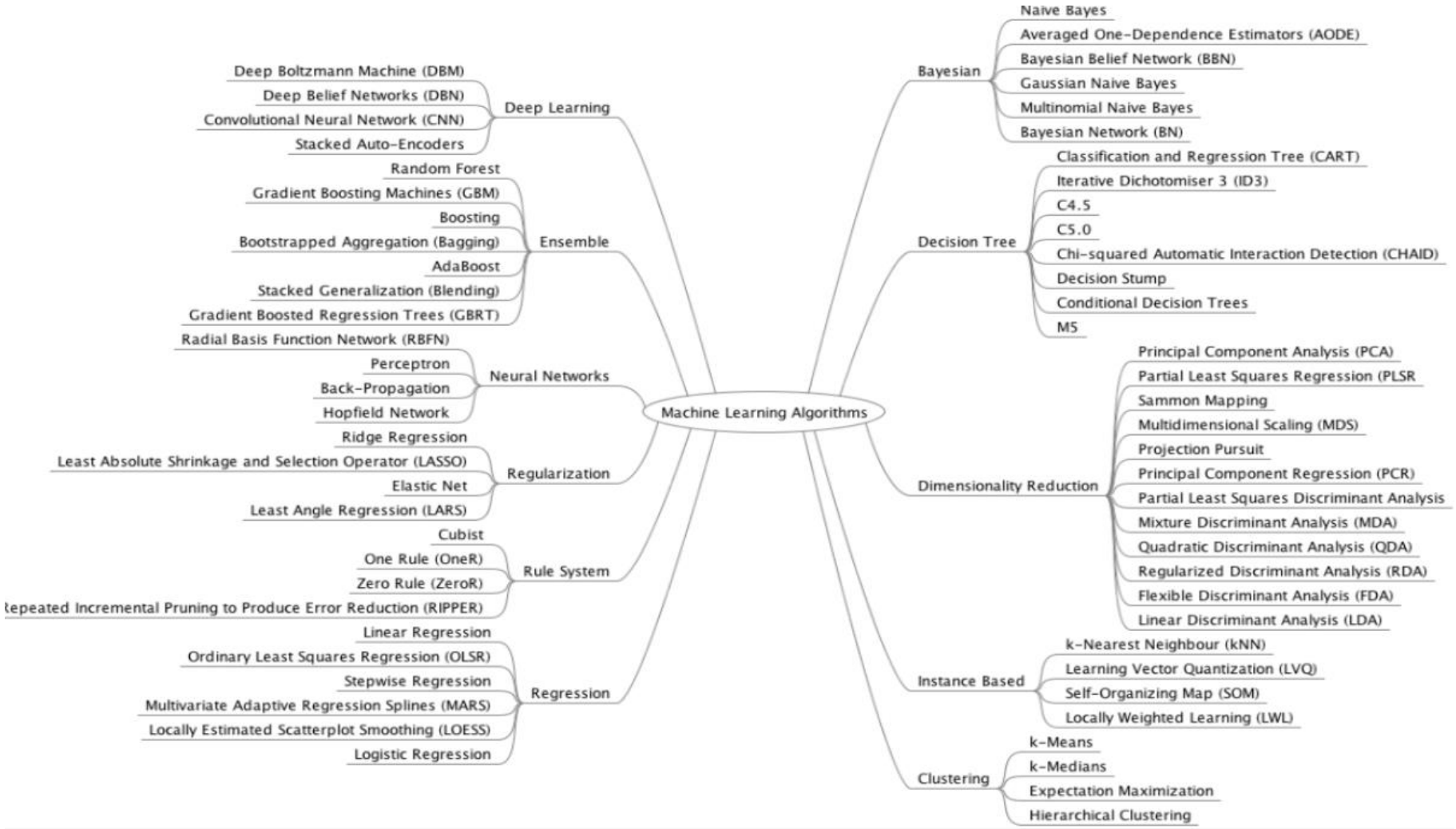
Finally, the developed models are deployed. They are continuously monitored to observe how they behaved in the real world and calibrated accordingly

Typically, the **modeling and deployment part (Data Science) is only 20%** of the work. **80% of the work (Data Engineering)** is getting your hands dirty with data, exploring the data and understanding it.

Machine Learning Problem Types



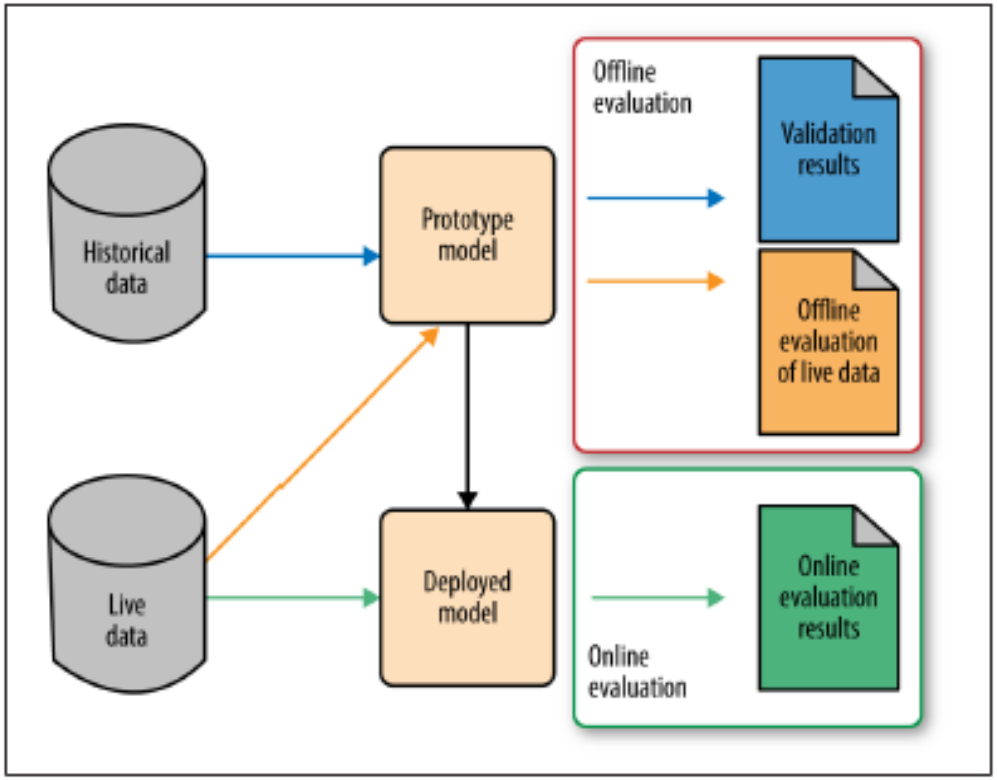
Machine Learning Algorithms



- We'll study many of them throughout the semester…

The Machine Learning Workflow

There are multiple **stages in developing a machine learning model** for use in a software application. It follows that there are multiple places where one needs to evaluate the model. The first phase involves ***prototyping***, where we try out different models to find the best one (***model selection***). Once satisfied with a prototype model, deploy it into production, where it will go through further testing on live data.



ML model development & evaluation workflow

Online evaluation measures live metrics of the deployed model on live data;

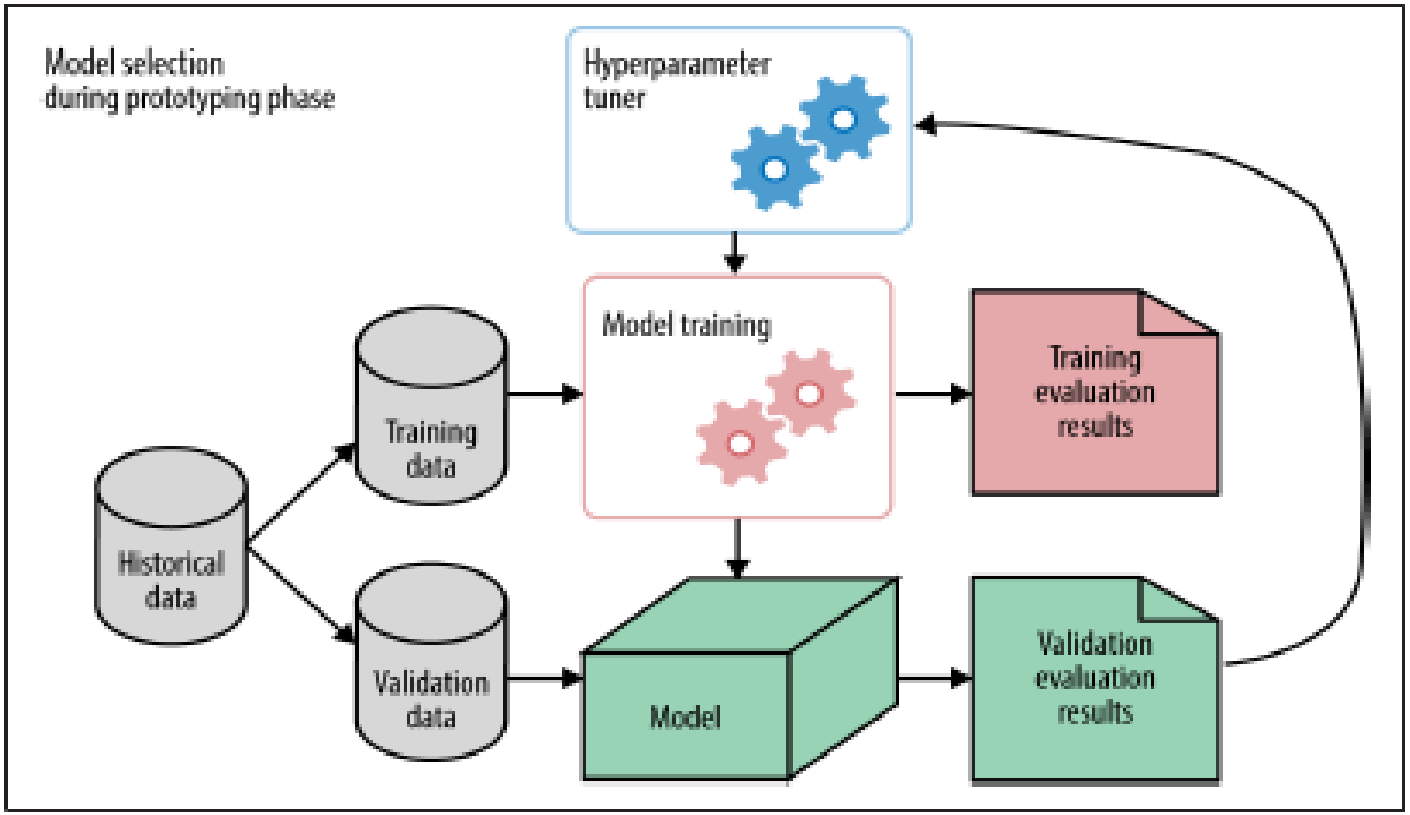
Offline evaluation measures offline metrics of the prototyped model on historical data (and sometimes on live data as well).

In other words, it's complicated.

The Prototyping Phase of Building a ML Model

This where we **tweak everything**: features, types of model, training methods

Each time we tweak something, we come up with a new model. **Model selection** refers to the process of selecting the right model (or type of model) that **fits the data**. This is done using validation results, not training results.



Hyperparameter Tuning Mechanism

Hyperparameter tuning is a meta-optimization task. Each trial of a particular hyperparameter setting involves training a model—an inner optimization process. The outcome of

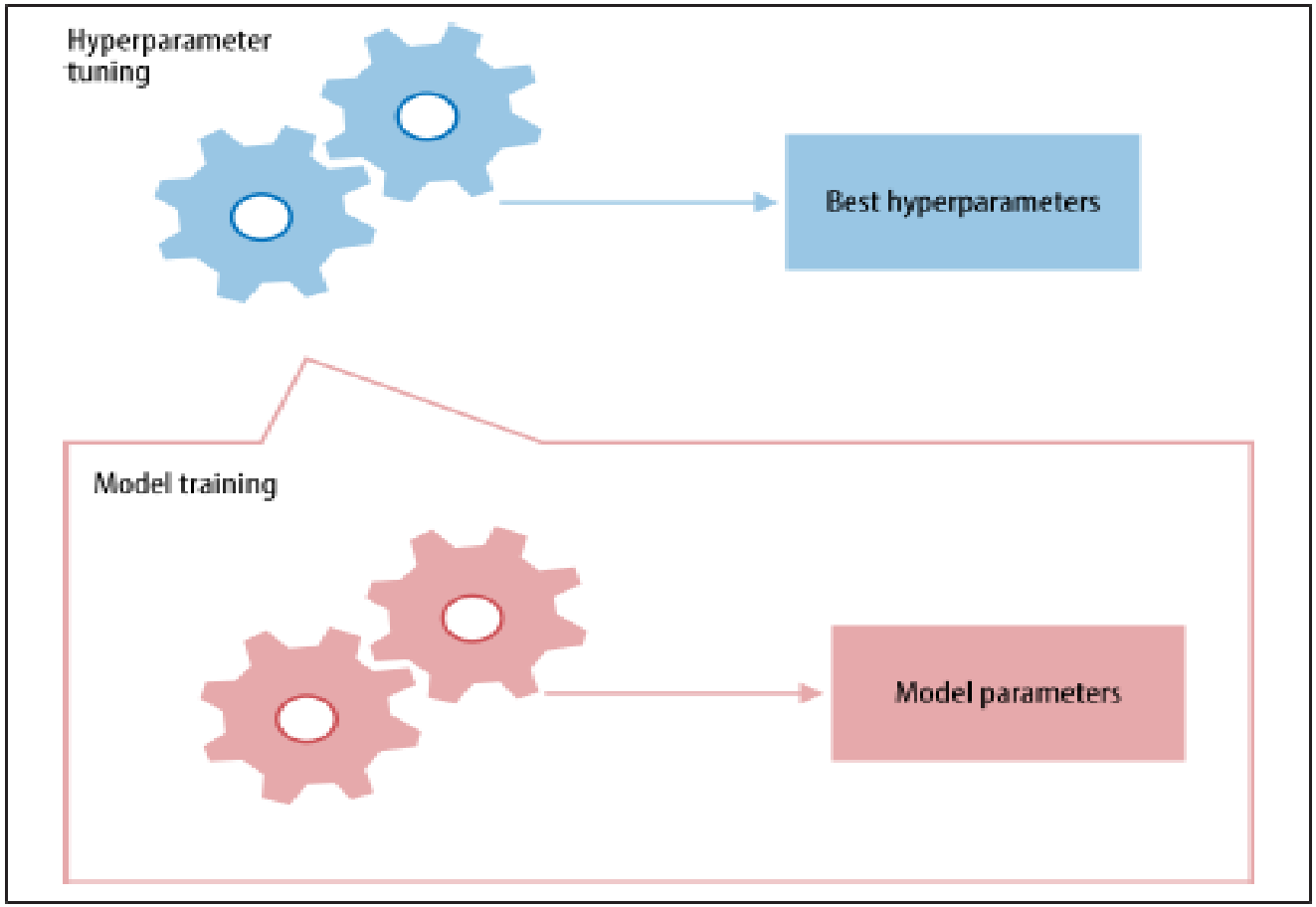
hyperparameter tuning is the best hyperparameter setting, and the outcome of model training is the best model parameter setting.

ML Components

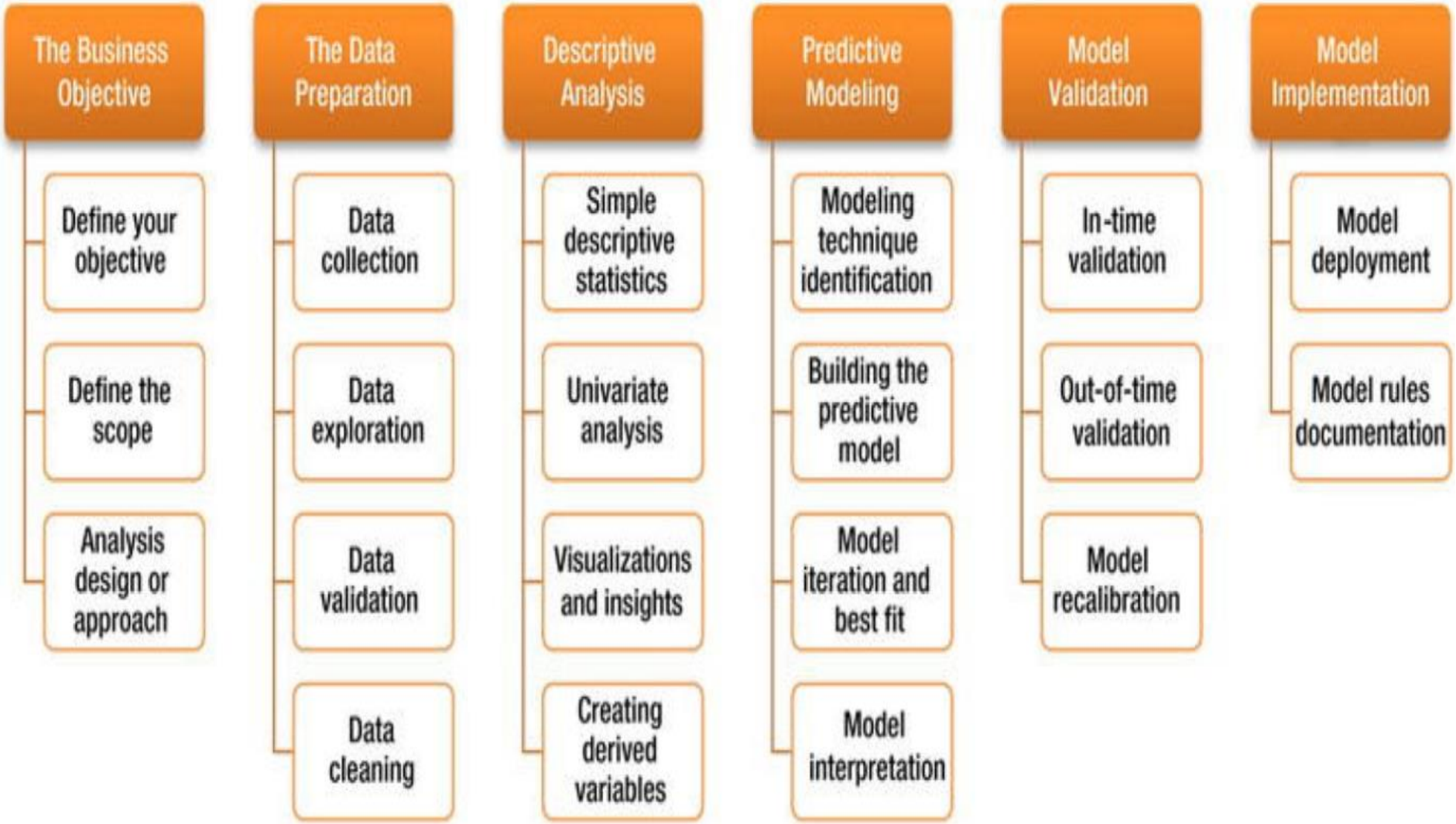
1_ Algo/Model

2_ Hyperparameters

3_ Data



An approach to Machine Learning & Data Analytics Lifecycle

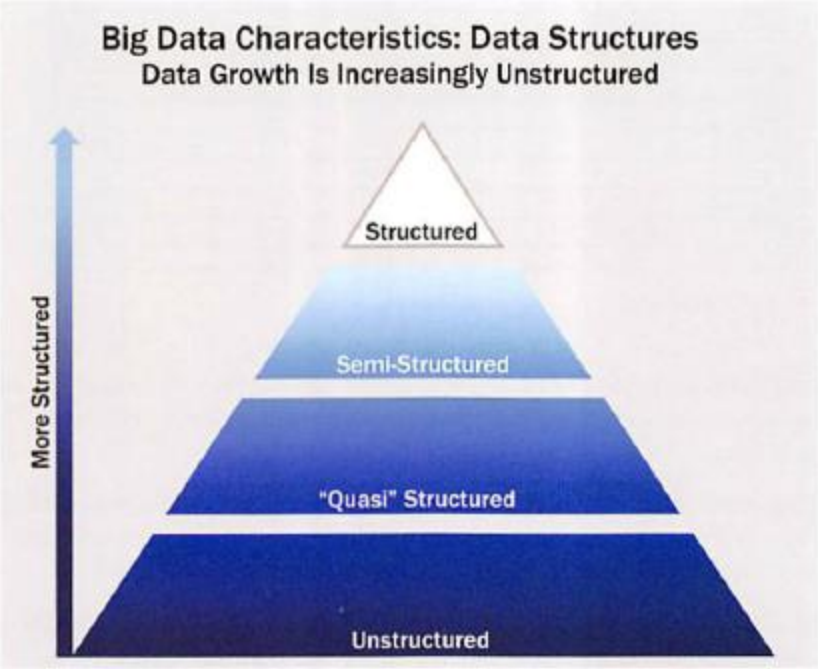


Courtesy: EMC Education (2015). *Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data* (1 ed.). Hoboken, New Jersey: Wiley.

Big Data Growth is increasingly Unstructured

Data Structures

Big data can come in multiple forms, including structured and non-structured data such as financial data, text files, multimedia files, and genetic mappings



Structured data: Data containing a defined data type, format, and structure (that is, transaction data, online analytical processing [OLAP] data cubes, traditional RDBMS, CSV files, and even simple spreadsheets

Semi-structured data: Textual data files with a discernible pattern that enables parsing (such as Extensible Markup Language [XML] data files that are self-describing and defined by an XML schema).

Quasi-structured data: Textual data with erratic data formats that can be formatted with effort, tools, and time (for instance, web clickstream data that may contain inconsistencies in data values and formats).

Unstructured data: Data that has no inherent structure, which may include text documents, PDFs, images, and video.

Unstructured, Non-structured data are associated with HLP. (HLP: Human Level Performance)

Attribute and Data Types (Statistically speaking…)

Attributes can be categorized into four types:

- Nominal, **O**rdinal, **I**nterval, and **R**atio (**NOIR**)

Nominal and **Ordinal** attributes are considered **Categorical (Qualitative)** attributes,

whereas **Interval** and **Ratio** attributes are considered **Numeric (Quantitative)** attributes.

Also, it is useful to consider these attribute types during the following discussion on R data types.

Numeric, Character, and Logical (Boolean) Data Types

Then build on top of them structures such as

Vectors, Arrays and Matrices, Data Frames, Factors, Contingency Tables

Numeric and Categorical Data (1)

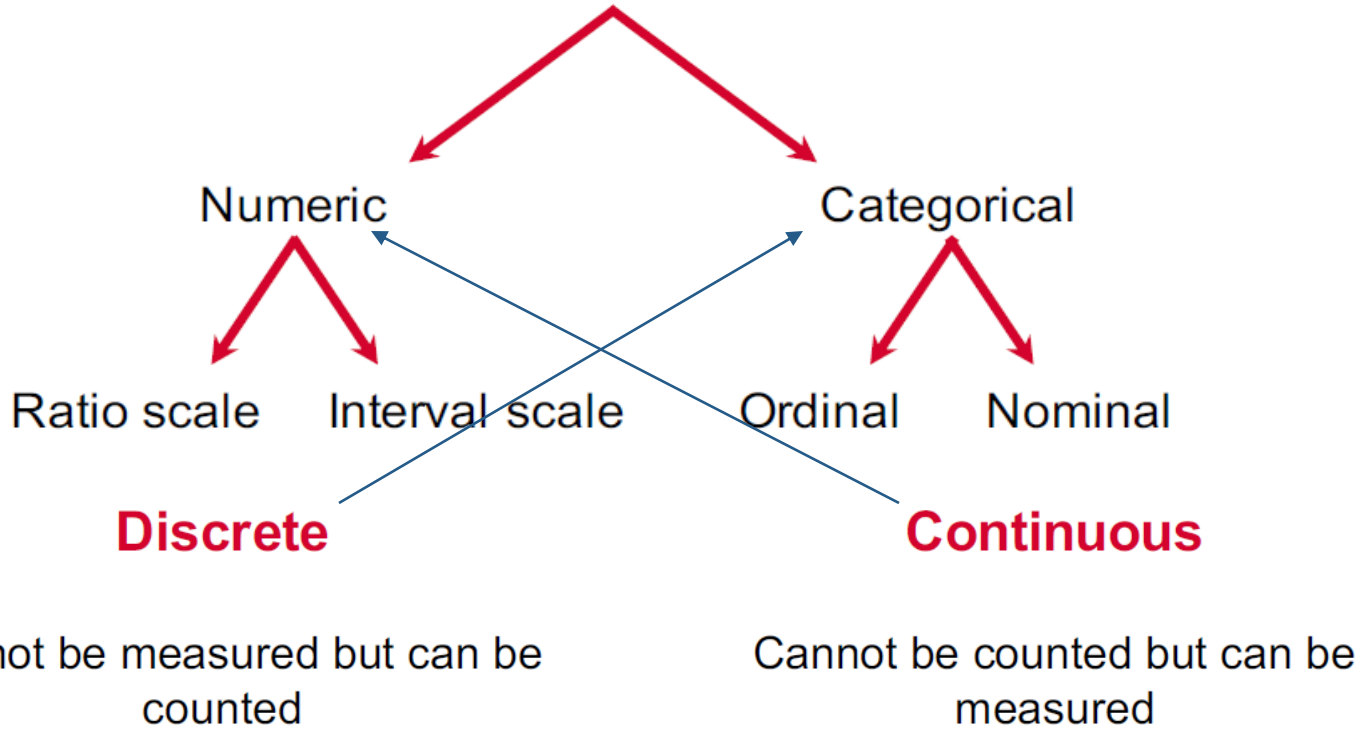
Numeric Data

- Prices of cars, heights of individuals
- Can take any value
- Predicted using **regression** models
- Always can be sorted on magnitude

Categorical Data

- Male/Female, True/False, Days of the week
- Finite set of permissible values
- Predicted using **classification** models
- Ordinal data can be sorted, nominal data cannot be sorted.

Numeric and Categorical Data (2)

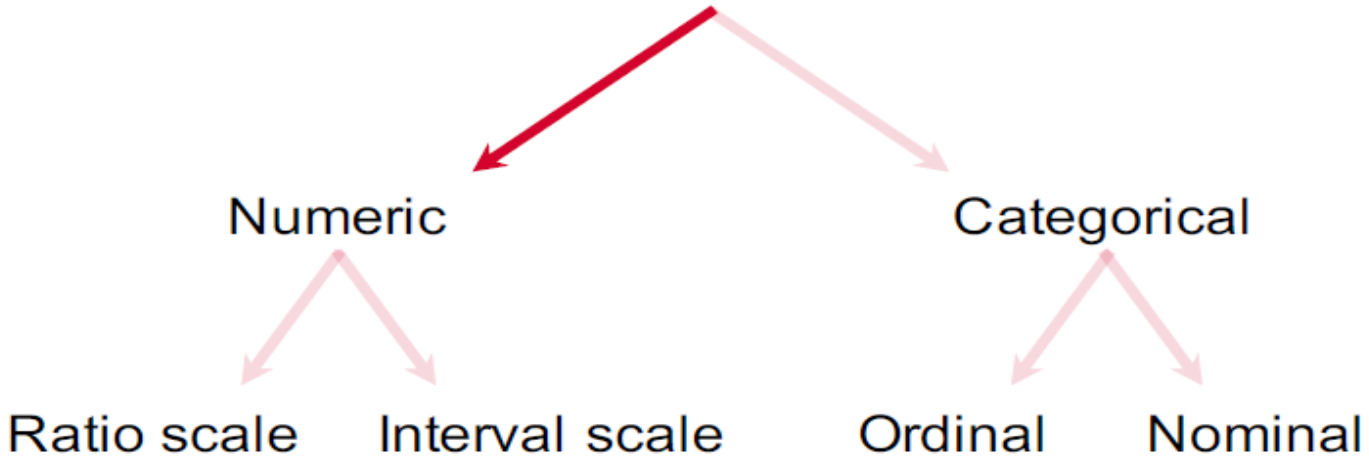


Examples:

Discrete: Number of visitors in a queue, number of children

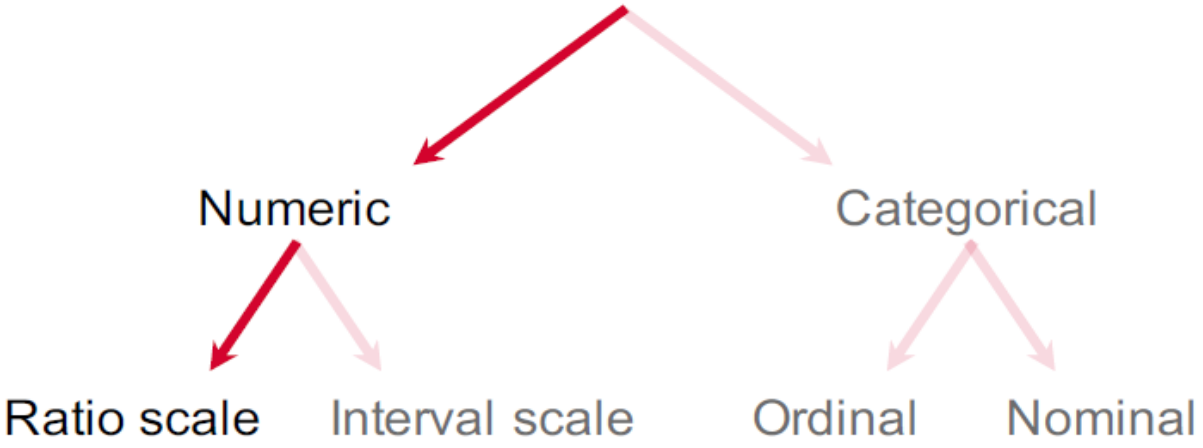
Continuous: Height of an individual, home prices, stock prices

Numeric Data (1)



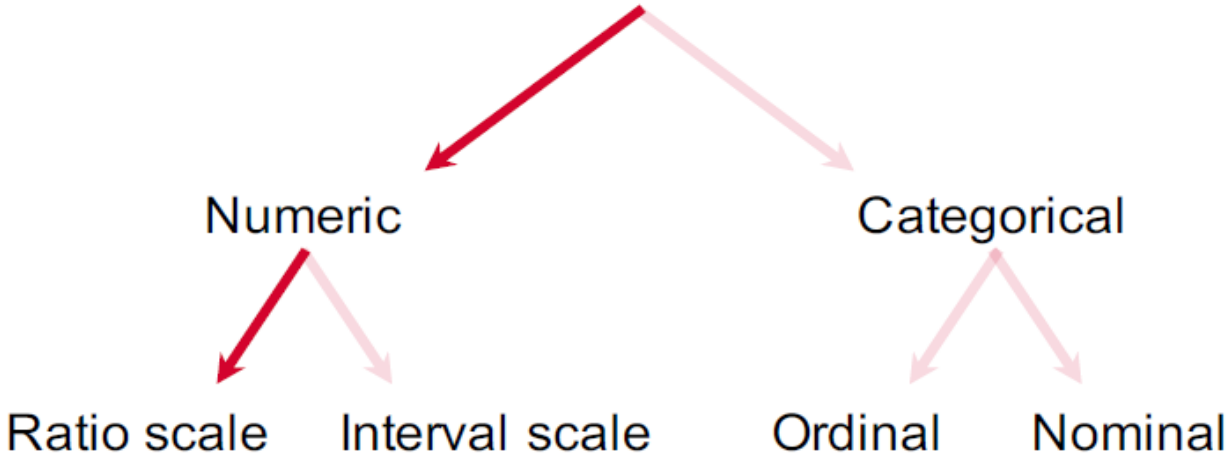
Associated with magnitude, can always be sorted i.e. comparable

Numeric Data (2)



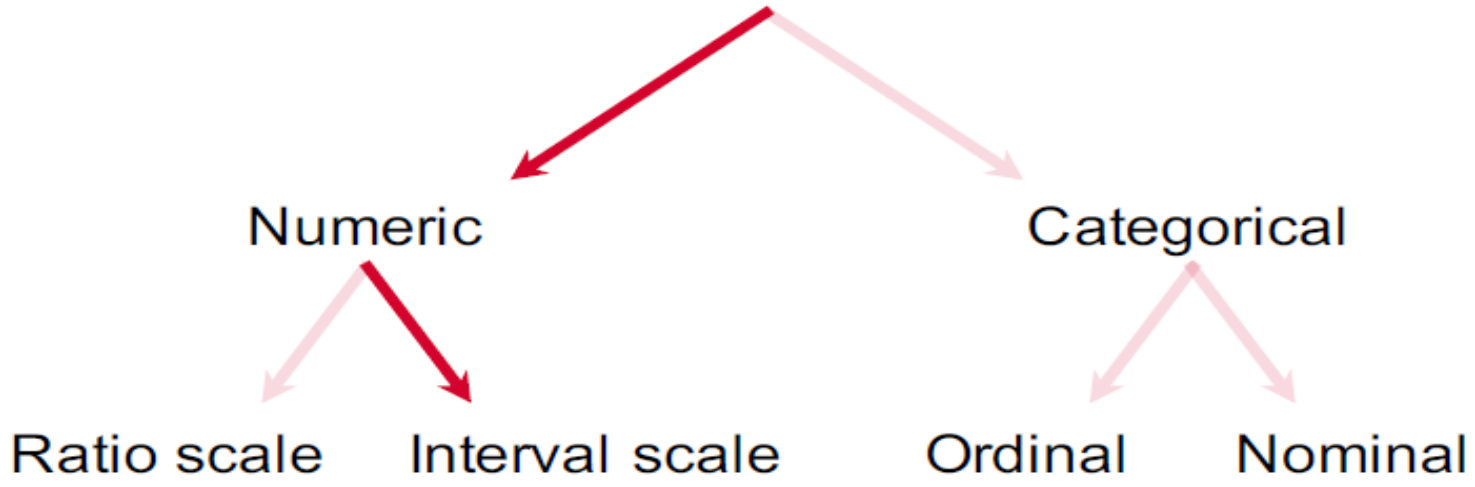
Can be expressed as a ratio of two numbers, arithmetic operations on the numbers make sense

Numeric Data (3)



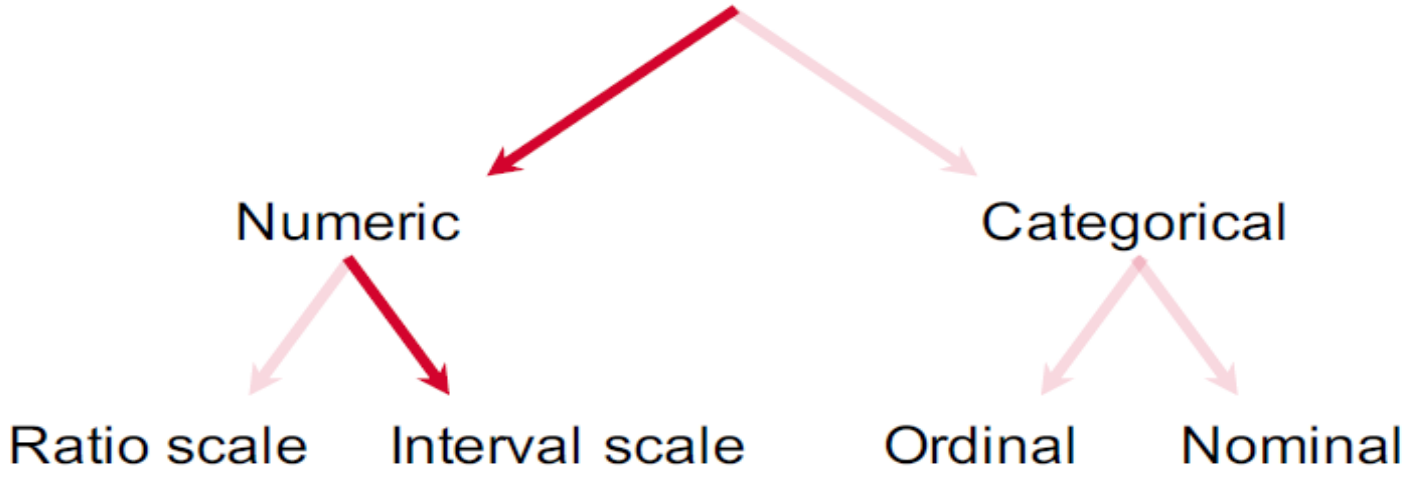
Ratio scale data has a meaningful zero point, zero is valid and has meaning

Numeric Data (4)



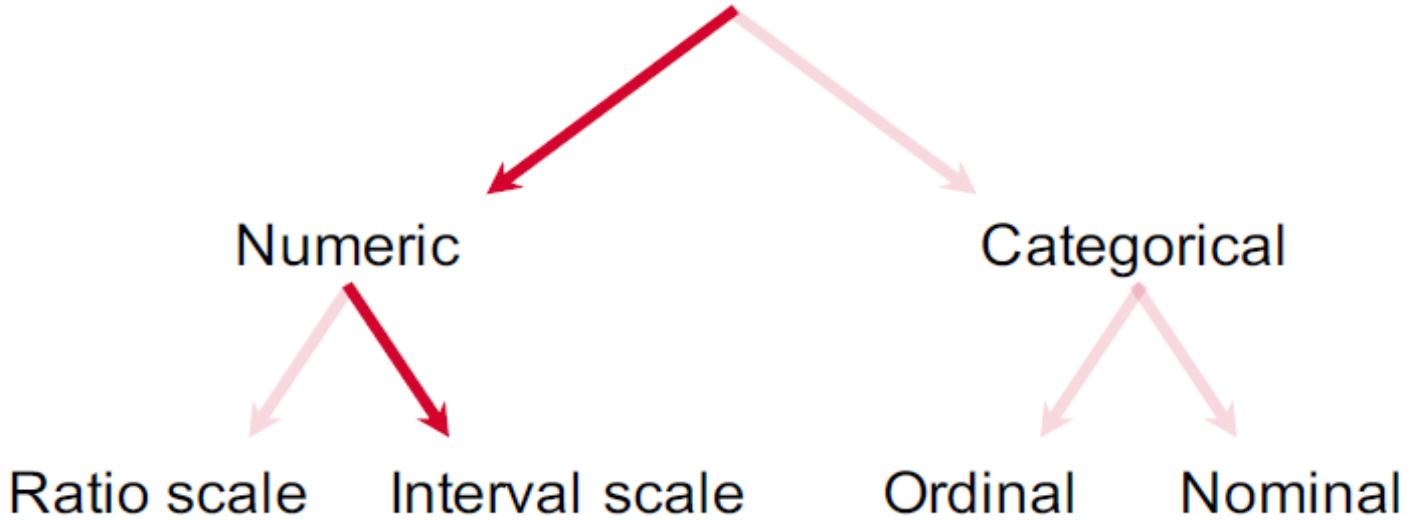
Ordered, numbered units with the same interval e.g. temperature in degrees

Numeric Data (5)



Arithmetic operations may no longer make sense i.e. adding temperatures

Numeric Data (6)



Zero does not mean “no value”

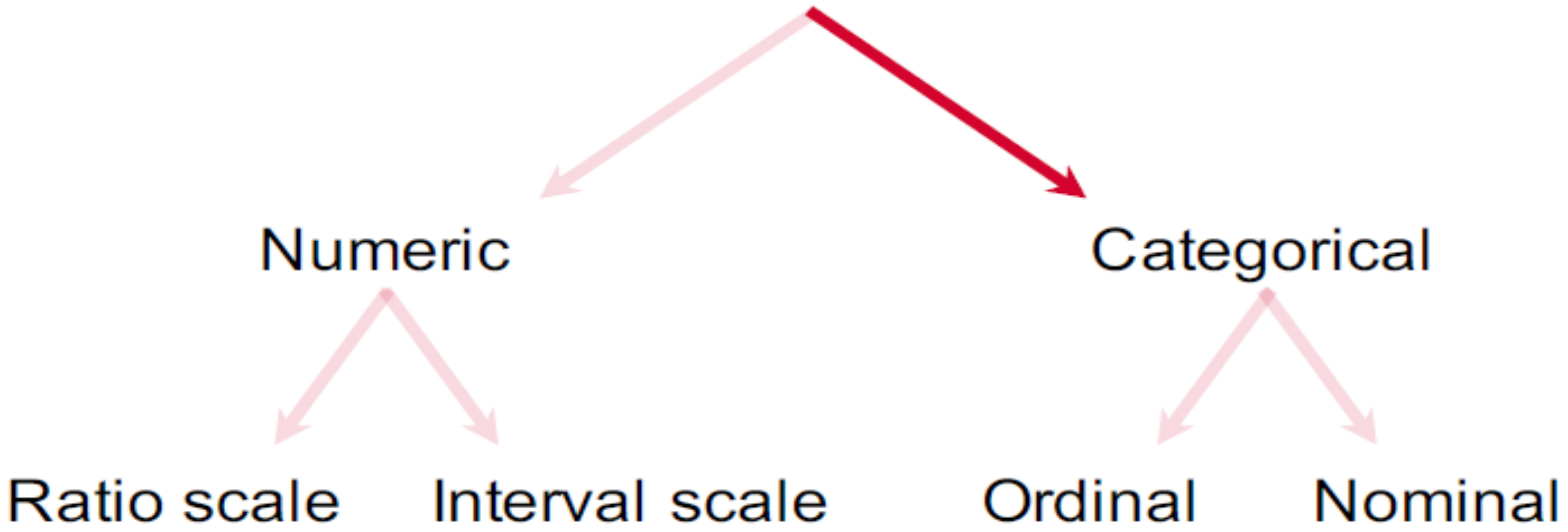
Numeric Data Pitfalls – ML Algorithms

This is **very important**, do remember it all the times:

If your numeric data is at **different** scales, machine learning algorithms may not work well with such data

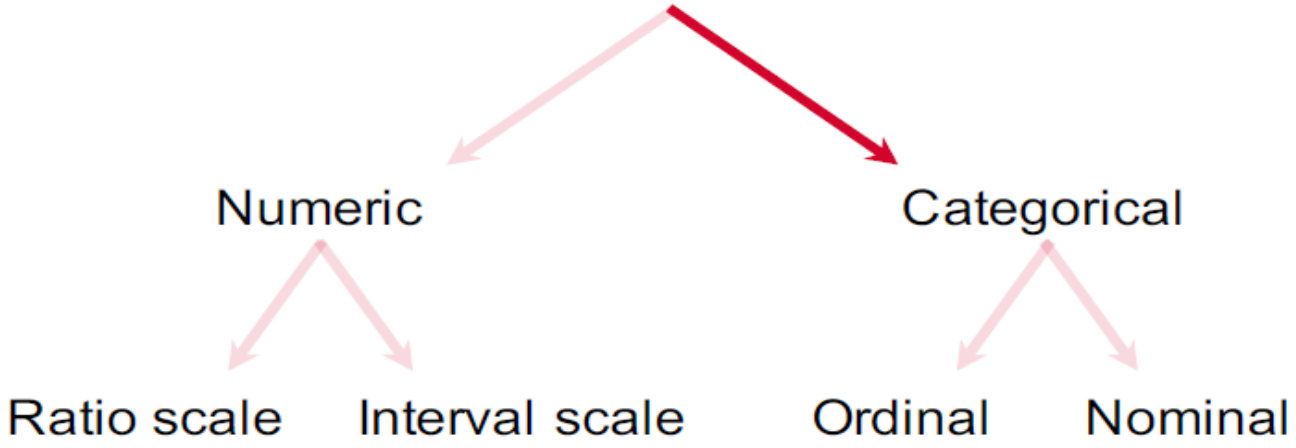
What is **the solution?**. See next slides…

Categorical Data (1)



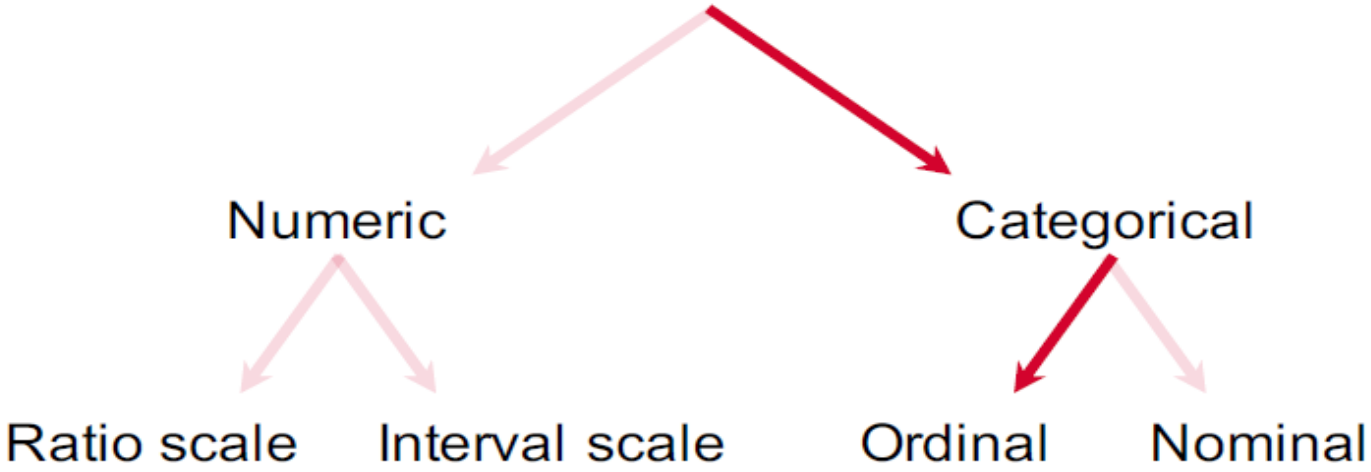
Discrete data which can only take on values from a specific set

Categorical Data (2)



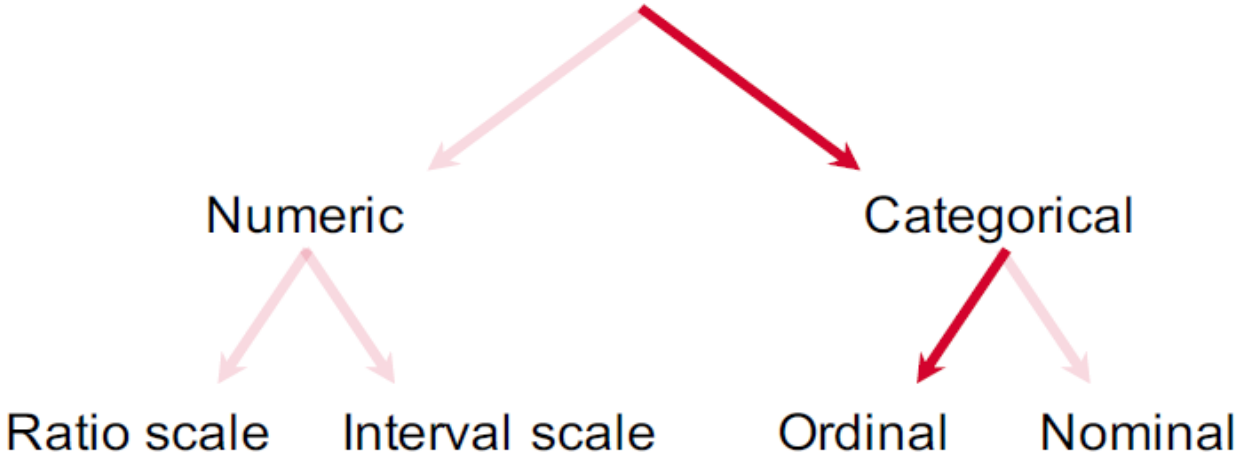
The values are categories, so arithmetic operations on the values will not make sense

Categorical Data (3)



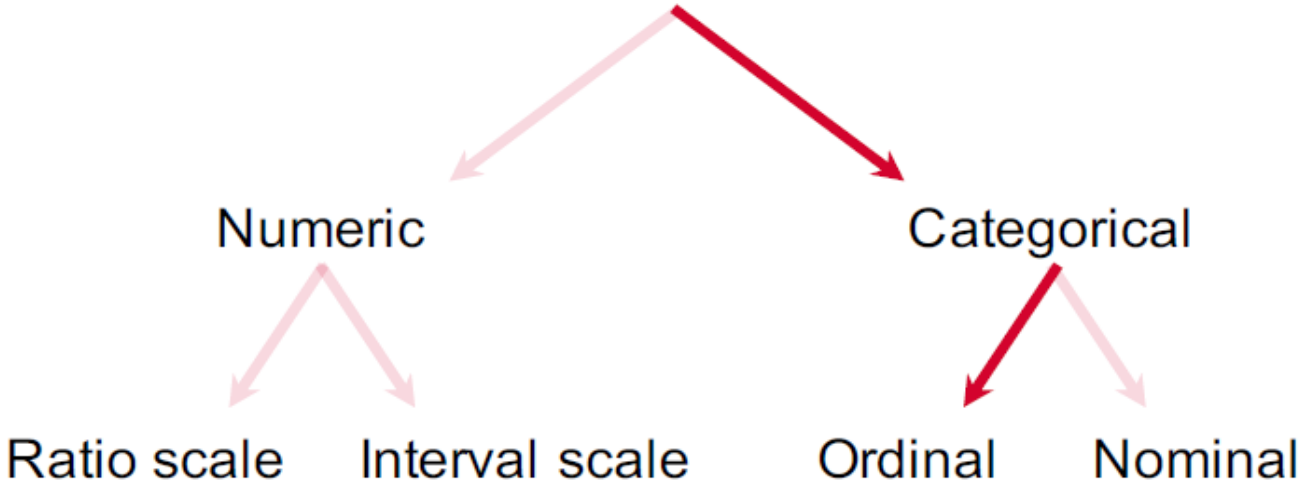
Categories have an intrinsic order

Categorical Data (4)



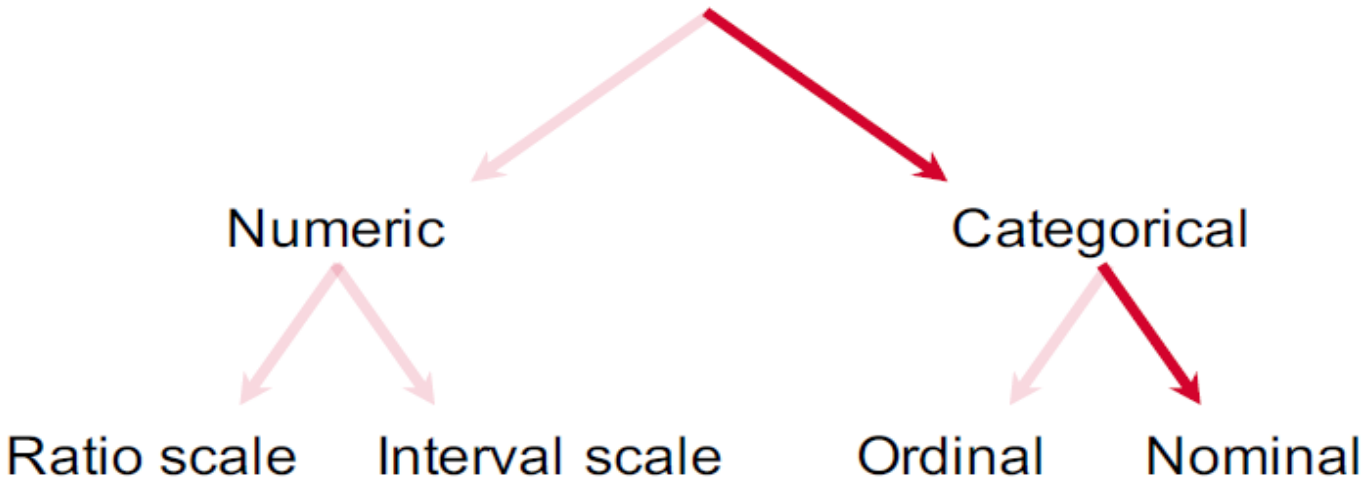
Days of the week, star ratings for movies

Categorical Data (5)



Differences in quality between star ratings are meaningful but the intervals may not be uniform

Categorical Data (6)



Nominal data cannot be ordered at all
e.g. true/false, male/female

Categorical Data and ML Models

This is **very important**, do remember it all the times:

Categorical data has to be **converted to numeric form** before it can be used in ML models

What is **the solution?**. **Encoding** text in numeric form.
See following slides…

Scaling and Standardization in ML

Numeric Features

- Can represent any kind of information
- The range of each feature will be different
- The average and dispersion of each of the features will also be different
- Comparing different features is hard (**Feature Engineering**)

Standardization **centers the data** so that every column has a mean of zero (0) and unit variance

Data is expressed in terms of **z-scores**

Statistics: Mean – Variance – Standard Deviation

In statistics, the **standard deviation** is a measure of the amount of variation or dispersion of a set of values. A **low standard deviation indicates that the values tend to be close to the mean** (also called the expected value) of the set, while a high standard deviation indicates that the values are spread out over a wider range.

$$\mu = \frac{\sum x_i}{N}$$

Mean (Average)

$$\sigma^2 = \frac{(x_1 - \mu)^2 + \cdots + (x_N - \mu)^2}{N}$$

Variance

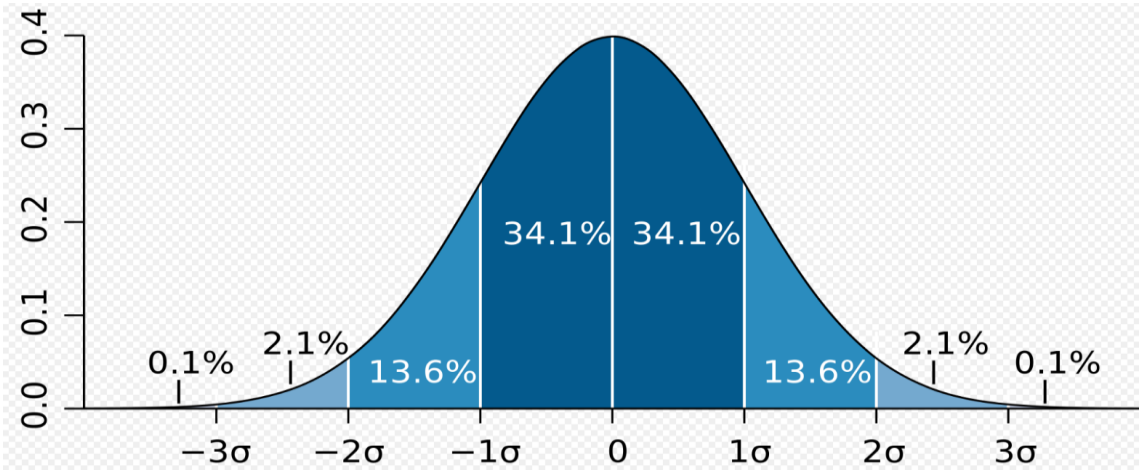
$$\sigma = \sqrt{\text{variance}} = \sqrt{\sigma^2}$$

Standard Deviation (SD)

Normal Distribution (Bell Curve)

SD: A measure used to quantify the amount of variation or dispersion of a set of data values.

- **1σ: 68.2% of data**
- **2σ: 95.4% of data**
- **3σ: 99.7% of data**



Standardizing Data (1)

$$\begin{bmatrix} X_{11} & \dots & X_{1k} \\ X_{21} & & X_{2k} \\ X_{31} & & X_{3k} \\ \dots & & \dots \\ X_{n1} & & X_{nk} \end{bmatrix}$$

$$\begin{matrix} avg(X_1) & \dots & avg(X_k) \\ stdev(X_1) & \dots & stdev(X_k) \end{matrix}$$

$$\begin{bmatrix} \frac{X_{11} - avg(X_1)}{stdev(X_1)} & \dots & \frac{X_{1k} - avg(X_k)}{stdev(X_k)} \\ \dots & & \dots \\ \frac{X_{n1} - avg(X_1)}{stdev(X_1)} & & \frac{X_{nk} - avg(X_k)}{stdev(X_k)} \end{bmatrix}$$

Each column of the standardized data has mean 0 and variance 1

Standardizing Data (2)

$$\begin{bmatrix} \frac{X_{11} - avg(X_1)}{stdev(X_1)} & \cdots & \frac{X_{1k} - avg(X_k)}{stdev(X_k)} \\ \cdots & & \cdots \\ \frac{X_{n1} - avg(X_1)}{stdev(X_1)} & & \frac{X_{nk} - avg(X_k)}{stdev(X_k)} \end{bmatrix}$$

Standardization is applied to features i.e. to all data in a single column

$$\begin{bmatrix} \frac{X_{11} - avg(X_1)}{stdev(X_1)} & \cdots & \frac{X_{1k} - avg(X_k)}{stdev(X_k)} \\ \cdots & & \cdots \\ \frac{X_{n1} - avg(X_1)}{stdev(X_1)} & & \frac{X_{nk} - avg(X_k)}{stdev(X_k)} \end{bmatrix}$$

Standardization is applied to features i.e. to all data in a single column

StandardScaler – Z-score values

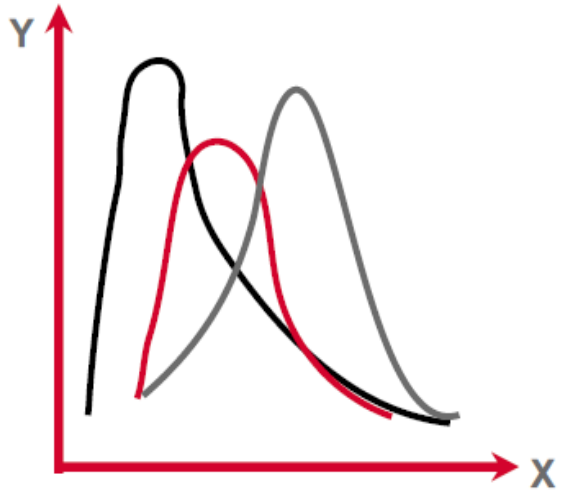
$$z = \frac{x_i - \text{mean}(x)}{\text{stdev}(x)}$$

Mean is a measure of central tendency and standard deviation is a measure of dispersion

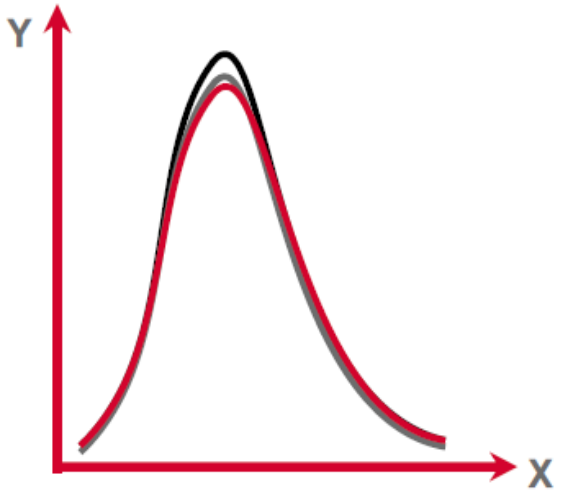
Scikit-Learn StandardScaler

- https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

StandardScaler



Before



After

StandardScaler & Outliers

This is **very important**, do remember it all the times:

Standardization is sensitive to the presence of outliers in the data

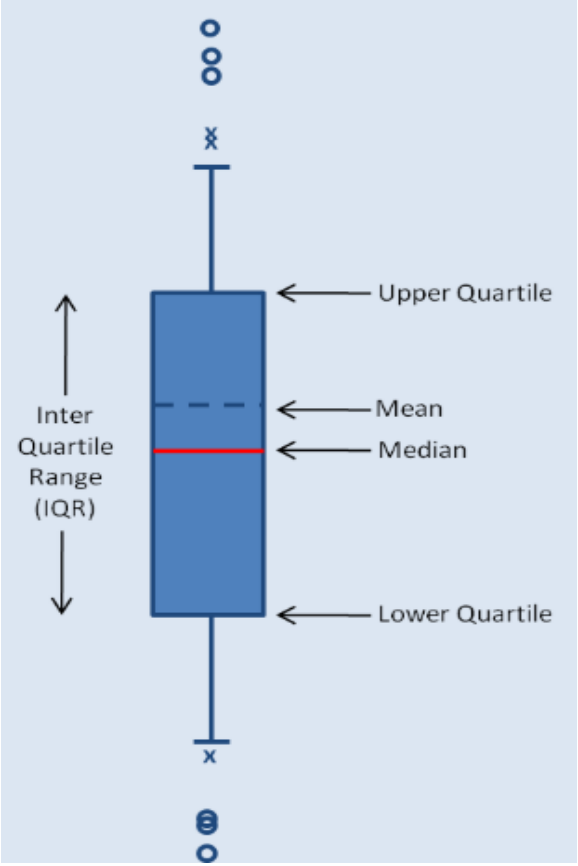
What is **the solution?**. See following slide…

RobustScaler – Inter-quartile Range (IQR)

$$z = \frac{x_i - \text{median}(x)}{\text{Inter-quartile Range}(x)}$$

RobustScaler is a scaler whose output does not change much due to outliers

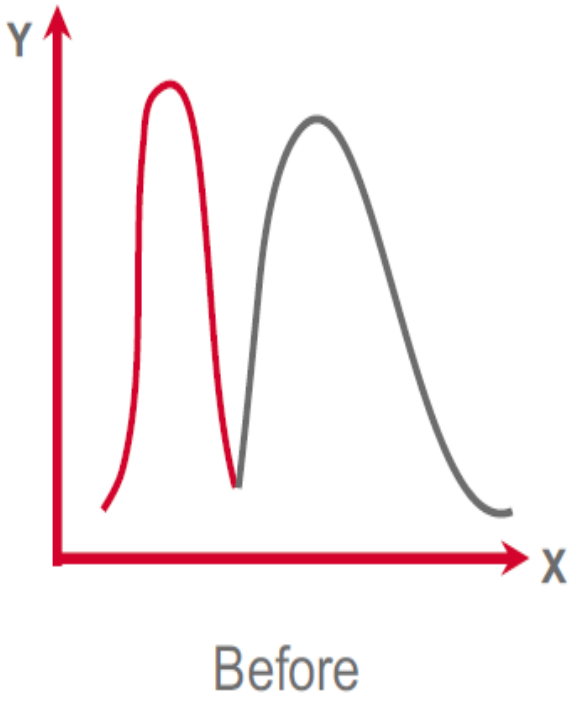
A Statistical Box Plot



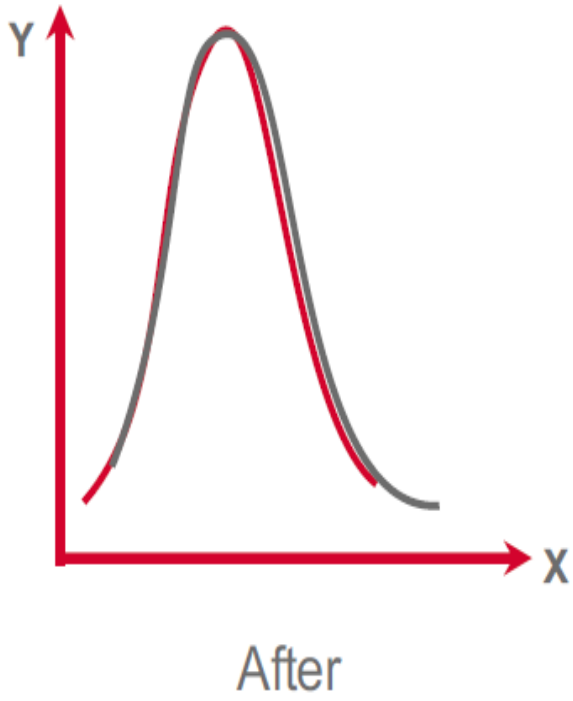
Scikit-Learn RobustScaler

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

RobustScaler



Before



After

Scaling – Normalization

Normalization: Process of scaling input vectors individually to unit norm (unit magnitude) i.e. magnitude of 1.

Often in order to simplify **cosine similarity** calculations.

Cosine similarity

Cosine similarity is a measure of similarity between two non-zero vectors, widely used in ML algorithms

Most often used in document modeling, clustering algorithms, NLP/NLG/NLU.

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \, \|B\|}$$

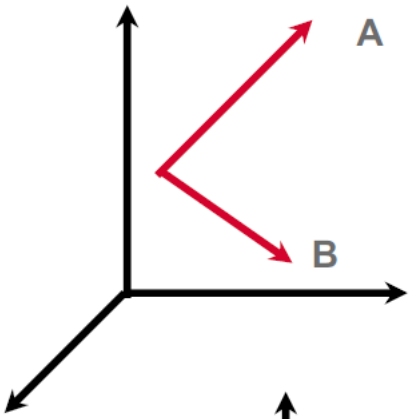
$$\|A\| = \sqrt{x_A^2 + y_A^2 + z_A^2}$$

$$\|B\| = \sqrt{x_B^2 + y_B^2 + z_B^2}$$

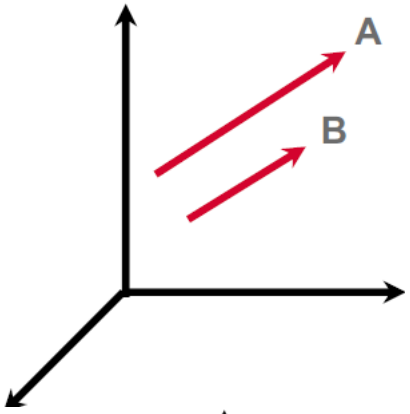
$$A.B = x_A x_B + y_A y_B + z_A z_B$$

Cosine Similarity – Data Similarity

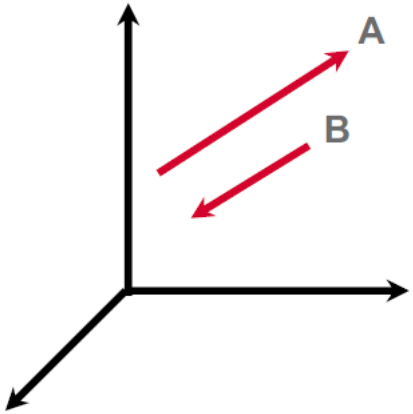
- **Uncorrelated Data**
- Vectors A & B are at 90 degrees
- Orthogonal vectors == uncorrelated data
- $Cos(90^0) = 0$



- **Fully Correlated Data**
- Vectors A & B are parallel
- Angle between them is zero (0) degrees
- Correlation (statistically) of 1
- $Cos(0^0) = 1$



- **Fully Negatively Correlated Data**
- Vectors A & B point in opposite (parallel) directions
- Angle between them is 180 degrees
- Correlation (statistically) of -1
- $Cos(180^0) = -1$



Normalization vs Cosine Similarity

Normalization

Pre-convert A and B to unit norm vectors to simplify calculation

$$a = \frac{A}{||A||} = \frac{(x_A, y_A, z_A)}{sqrt(x_A^2 + y_A^2 + z_A^2)}$$

$$b = \frac{B}{||B||} = \frac{(x_B, y_B, z_B)}{sqrt(x_B^2 + y_B^2 + z_B^2)}$$

Cosine Similarity

$$\cos(\theta) = \frac{A \cdot B}{||A||\ ||B||}$$

$$||A|| = 1$$

$$||B|| = 1$$

$$A.B = x_A x_B + y_A y_B + z_A z_B$$

Normalization vs Standardization

This is **very important**, do remember it all the times:

Normalization acts on vectors i.e. rows unlike standardization which acts on features i.e. columns

Normalization Types

Different Norms

L1

Sum of absolute values of components of vector

$$X_{new} = \frac{(x, y, z)}{|x| + |y| + |z|}$$

L2

Traditional definition of vector magnitude

$$X_{new} = \frac{(x, y, z)}{\text{sqrt}(x^2 + y^2 + z^2)}$$

max

Largest absolute value of elements of vector

$$X_{new} = \frac{(x, y, z)}{\max(\text{abs}(x, y, z))}$$

Data Types – Use Cases (Statistics point of view)

- **Time Series** (Time-ordered) Data
- **Cross Section** (Cross-sectional) Data
- **Longitudinal** (Panel) Data
- **Unstructured** Text Data

Data Types – Time Series Data

Time series data: series of numeric data points of some particular metric over time.

As name suggests, involves working on time (years, days, hours, min) based data,

Time Series Modeling is very useful in ML techniques such as prediction and forecasting, specifically for models that deal with serially correlated data.

Many businesses work on time series data to analyze sales number for the next year, website traffic, financial data (forecast movements in stock markets, DowJones)

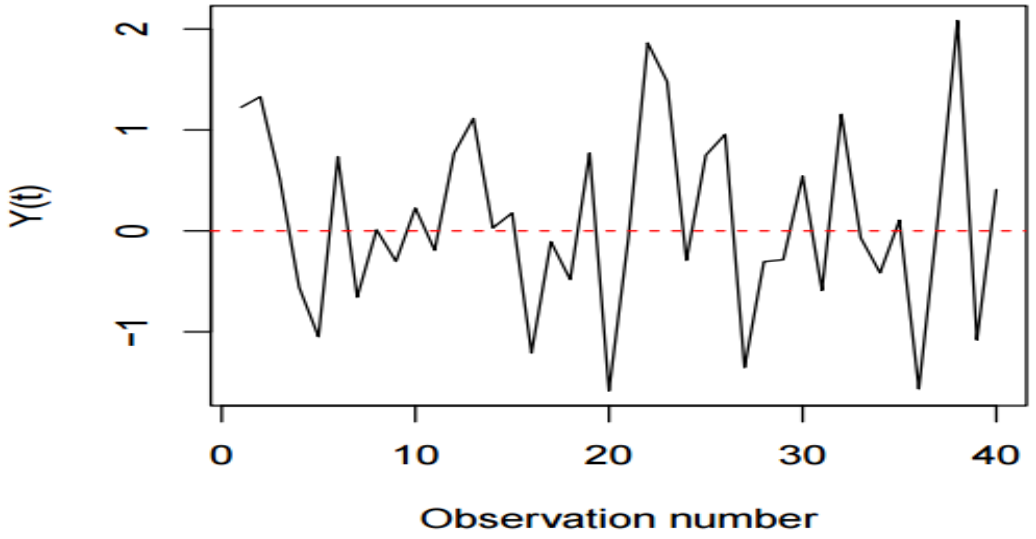
In Statistics, the same **variable** or variables **observed** and **measured** at **consecutive points of time**. Usually but not necessarily, the points of time are equally spaced.

Time-ordered data are very often pertinent for total quality.

The simplest time series model is th

$$Y(t) = \mu + \epsilon(t)$$

Here, **μ** is a constant,
and **ε(t)** is the residual (or error) term.



Data Types – Multivariate Time Series (MTS) Data

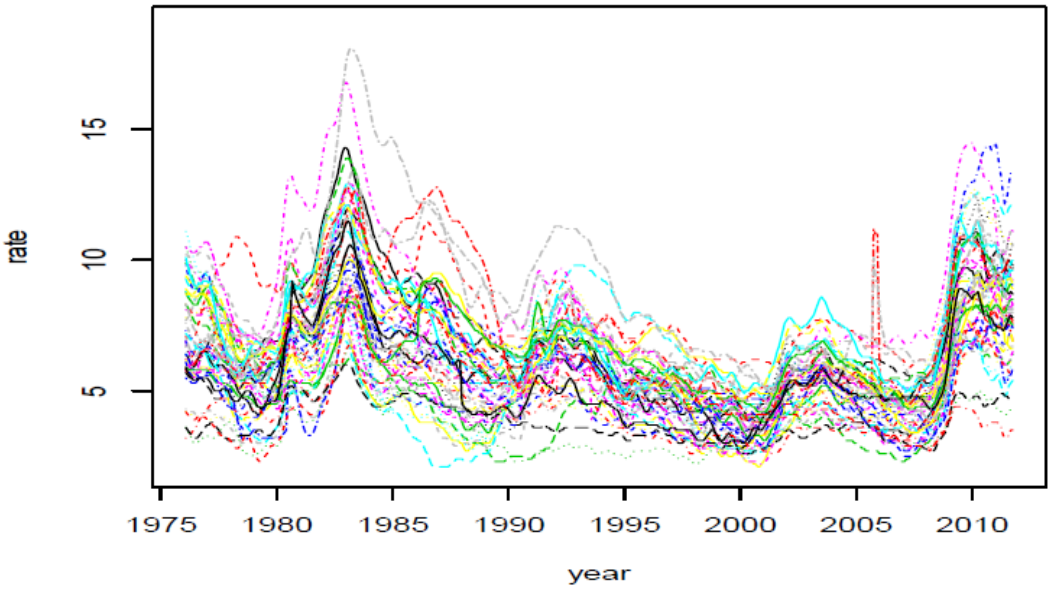
Multivariate time series (MTS) analysis is used when one wants to model and explain the interactions and co-movements among a group of time series variables:

- Consumption and income
- Stock prices and dividends
- Forward and spot exchange rates
- Interest rates, money growth, income, inflati
- Rate of unemployment (per region)

Mathematical Definition

Consider n time series variables $\{y_{1t}\}, \ldots, \{y_{nt}\}$. A *multivariate time series* is the $(n \times 1)$ vector time series $\{\mathbf{Y}_t\}$ where the i^{th} row of $\{\mathbf{Y}_t\}$ is $\{y_{it}\}$. That is, for any time t, $\mathbf{Y}_t = (y_{1t}, \ldots, y_{nt})'$.

Time plots of the **monthly unemployment rates** of the **50 States** in the U.S. from January 1976 to September 2011.



See **Python's Statsmodel** package.
http://statsmodels.sourceforge.net/stable/tsa.html

See **R's MTS** package. https://cran.r-project.org/web/packages/MTS/MTS.pdf

Data Types – Cross Section Data

Not all data are time-ordered!

There is a type of data (structure) called **cross-section data**, where we are dealing with information about different individuals (or aggregates such as work teams, sales territories, stores, etc,) **at the same point of time or during the same time period**.

Many things can be learned from cross-sectional data especially in processes where **statistical control** is required. For instance, we might need to evaluate the effectiveness of interventions which aim to **improve processes and controls** (see Six Sigma data-driven methodology for eliminating defects), and to assure that gains being hold from effective interventions from the past.

Another example, we might have data on total accidents per worker over the course of the last calendar year for all the workers in a given plant, or we might have questionnaire data on customer satisfaction for a sample of customers last month.

Analysis of cross-sectional data usually consists of comparing the differences among the subjects. This is of huge importance in **Health Care** vertical since studies involve data collected at a defined time. They are often used to assess the prevalence of acute or chronic conditions, or to answer questions about the causes of disease or the results of intervention.

An example of cross-sectional analysis in **Economics** is the regression of money demand—the amounts that various people hold in highly liquid financial assets—at a particular time upon their income, total financial wealth, and various demographic factors.

What are Longitudinal (Panel) Data? --- Statistical Analysis

A dataset is **longitudinal** if it tracks the same type of information on the same subjects at multiple points in time.

The primary advantage of longitudinal databases is that they can measure *change*.

The longitudinal data extend into the **past** as well as the **present**.

Example of Longitudinal Dataset: Students and their test scores in successive years

Here we can estimate, the effect of various factors on improvement in student achievement. Can also estimate overall effectiveness of individual teachers by examining the performance of successive classes of students they teach, also examine the extent to which teacher effectiveness changes with experience or the composition of their class.

Also, evaluate the effect of a specific policy by looking at, say, student performance or teacher turnover before as well as after the policy was introduced.

Student Name	Grade 1 (2001) Raw Score	Grade 2 (2002) Raw Score	Grade 3 (2003) Raw Score	Grade 4 (2004) Raw Score	Grade 5 (2005) Raw Score	Grade 6 (2006) Raw Score
Mike	339	350	361	366	381	390
Jasmine	332	343	350	351	351	355
Thomas	360	380	400	420	430	438

Data Types – Unstructured Text Data

Content is everywhere. **Unstructured textual data** is being constantly generated:

- Call Center logs
- Emails
- Documents on the Web
- Blogs
- Tweets
- Customer comments
- Customer reviews
- Machine-to-Machine data

While the amount of textual data is increasing rapidly, businesses' ability to summarize, understand, and make sense of such data for making better business decisions remain challenging.

As per an IDC survey, **unstructured data** takes a lion's share in digital space and approximately occupies **80%** by volume compared to only **20%** for **structured data**.

Methods for **analyzing unstructured data**, historically, come out of technical areas such as **Natural Language Processing (NLP), knowledge discovery, data mining, information retrieval, and statistics.**

Text analytics is the process of analyzing unstructured text, extracting relevant information, and transforming it into structured information that can then be leveraged in various ways.

Data Types – Unstructured Text Data … cnt'd

Search is about retrieving a document based on what we already know they are looking for. **Text analytics** is about **discovering information**. While text analytics differs from search, it augments search techniques.

Using **text analytics algorithms**, a trend may emerge from the unstructured data.

That's what makes text analytics so powerful!

There are four **technologies** used for **data retrieval**: query, data mining, search, and text analytics.

Retrieval	Insight	
Structured	Query: Returns data	Data mining: Insight from structured data
Unstructured	Search: Returns documents	Text analytics: Insight from text

Text Analytics Use Case ==➔ Sentiment Analysis

A **Sentiment Analysis** solution needs to provide a rich set of **contextual information** that helps you **understand what is really being said** about you, your products, or your brand and to what extent, and through which channels are impacting you and what you can do about it.

Python's NLTK Sentiment Analysis Package (see also **SpaCy**):
http://www.nltk.org/api/nltk.sentiment.html

R's Sentiment Analysis package:
https://github.com/timjurka/sentiment

R's Natural Language Processing (NLP) libraries:
https://cran.r-project.org/web/views/NaturalLanguageProcessing.html

Questions?