BIG DATA

Chananel Perel
2024

Big Data - Definition

Big Data ("בִּיג דָּאטָה", לפי החלטת האקדמיה ללשון העברית: נְתוּנֵי עָתֵק)

הוא מונח המתייחס למאגר מידע הכולל נתונים מבוזרים, שאינם מאורגנים לפי שיטה כלשהי, שמגיעים ממקורות רבים, בכמויות גדולות, בפורמטים מגוונים, ובאיכויות שונות.

הגדרה מקובלת במחקר של המושג "נתוני עתק" היא זו המנוסחת בערך "big data" במילון אוקספורד: "נתוני עתק הם גוף נתונים גדול במידה כזו, שעיבודו, ארגונו וניתוחו מעמידים אתגר מיוחד."

ניתן לאפיין נתוני עתק לפי כמה מאפיינים (**חמשת ה-V-ים**) (יש כאלו שמונים: 10 V [קישור]):

- נפח (**volume**)
- גיוון (**variety**)
- מהירות (**velocity**)
- אי-אמינות (**veracity**)
- ערך (**value**)

מתוך ויקיפדיה (https://he.wikipedia.org/wiki/Big_data)

Big Data - Definition 2

האתגר בניהול נתוני עתק הפך תחום זה לעניין מרכזי בטכנולוגיית מידע. מסדי הנתונים היחסיים הקיימים אינם בנויים לאחסון ולניתוח כמויות מידע גדולות שרובן אינן מגיעות באופן מפורמט לפי תבניות אחידות וידועות מראש. העלות הזולה יחסית של אמצעי האחסון, מצד אחד, והכמות הגדולה של מידע המגיע משלל מקורות (אתרי אינטרנט, רשתות חברתיות, מכשירים סלולריים, מצלמות אבטחה, חיישנים ועוד), מצד שני, גורם לכך שמידע נאגר ללא מחיקה, ומאפשר יכולות ניתוח וזיהוי תבניות ומִתְאָמִים, הנדרשות בעולמות תוכן רבים.

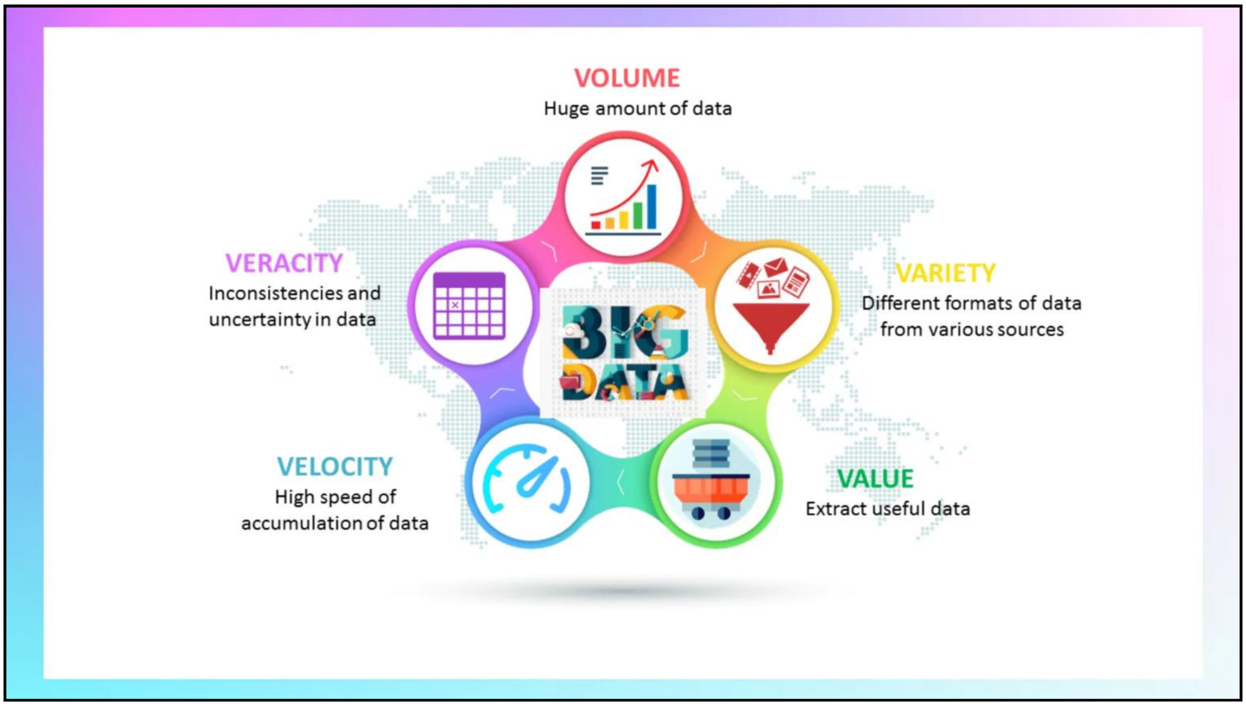
ביג דאטה משמש לתיאור היקף מאסיבי של אוסף נתונים מובנה ושאינו מובנה, שגודלו העצום -וקצב עדכונו המהיר, לא היו בנמצא עד לפני זמן קצר.

Big Data - Definition 3

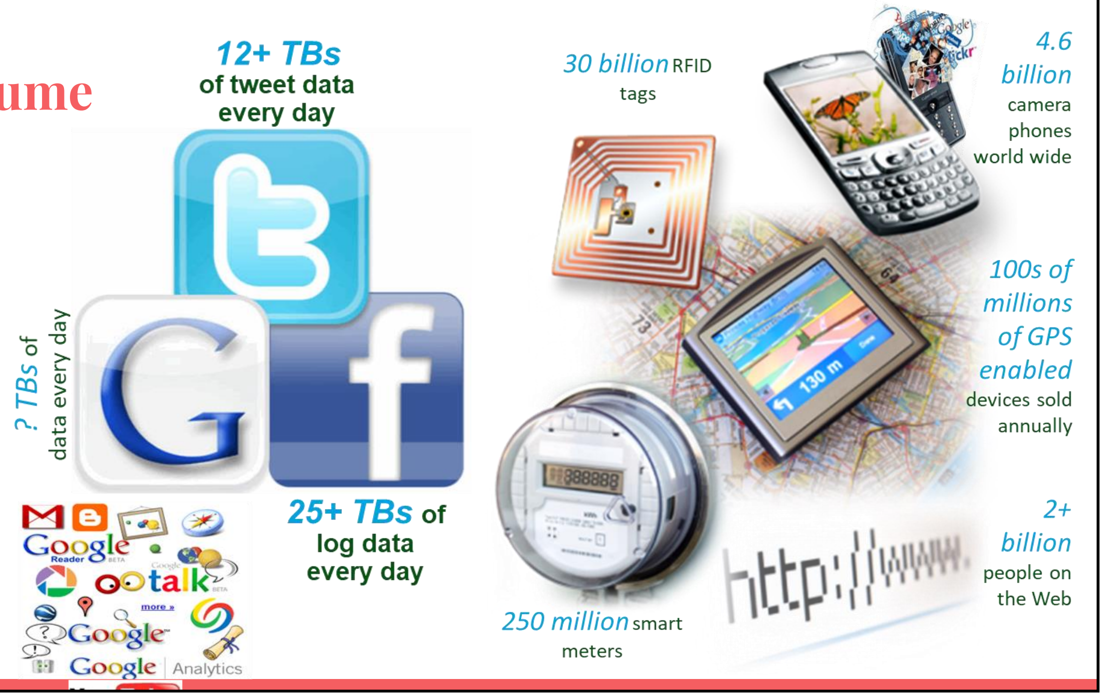
Big data primarily refers to data sets that are too large or complex to be dealt with by traditional data-processing application software. Big data has many entries or rows and it has many attributes or columns (higher complexity).

Big data analysis challenges include:

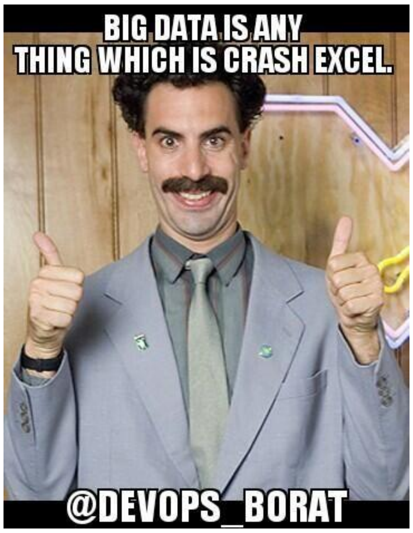
- capturing data
- data storage
- data analysis
- search
- sharing
- transfer
- visualization
- querying
- updating
- information privacy
- data source


VOLUME
Huge amount of data
VARIETY
Different formats of data from various sources
VALUE
Extract useful data
VELOCITY
High speed of accumulation of data
VERACITY
Inconsistencies and uncertainty in data
BIG DATA

Volume


12+ TBs of tweet data every day
? TBs of data every day
25+ TBs of log data every day
30 billion RFID tags
4.6 billion camera phones world wide
100s of millions of GPS enabled devices sold annually
250 million smart meters
2+ billion people on the Web
130 m
http://www.

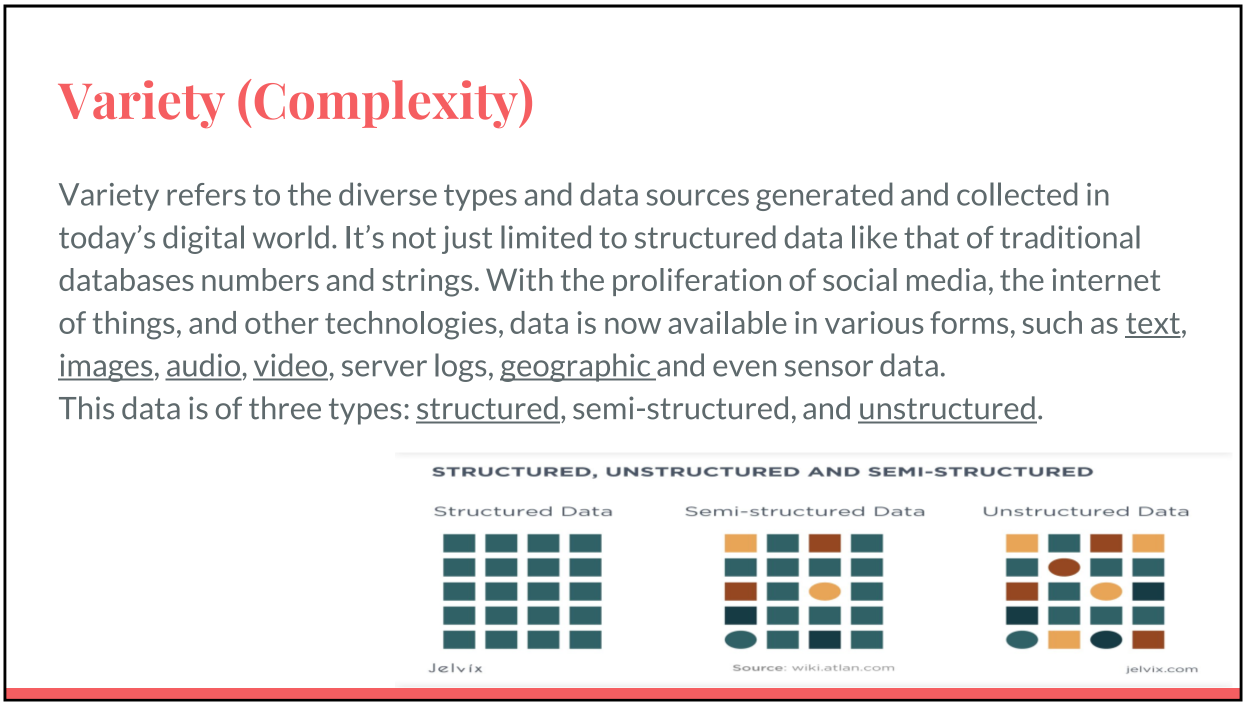
Volume



Volume (Scale)

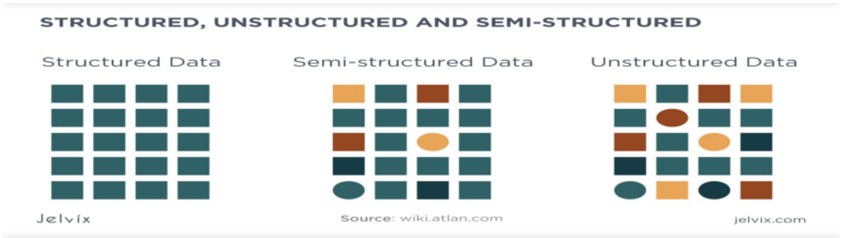
Today Terabyte size DB are common, so we may define big data as hundreds of Terabytes or Petabytes in size.

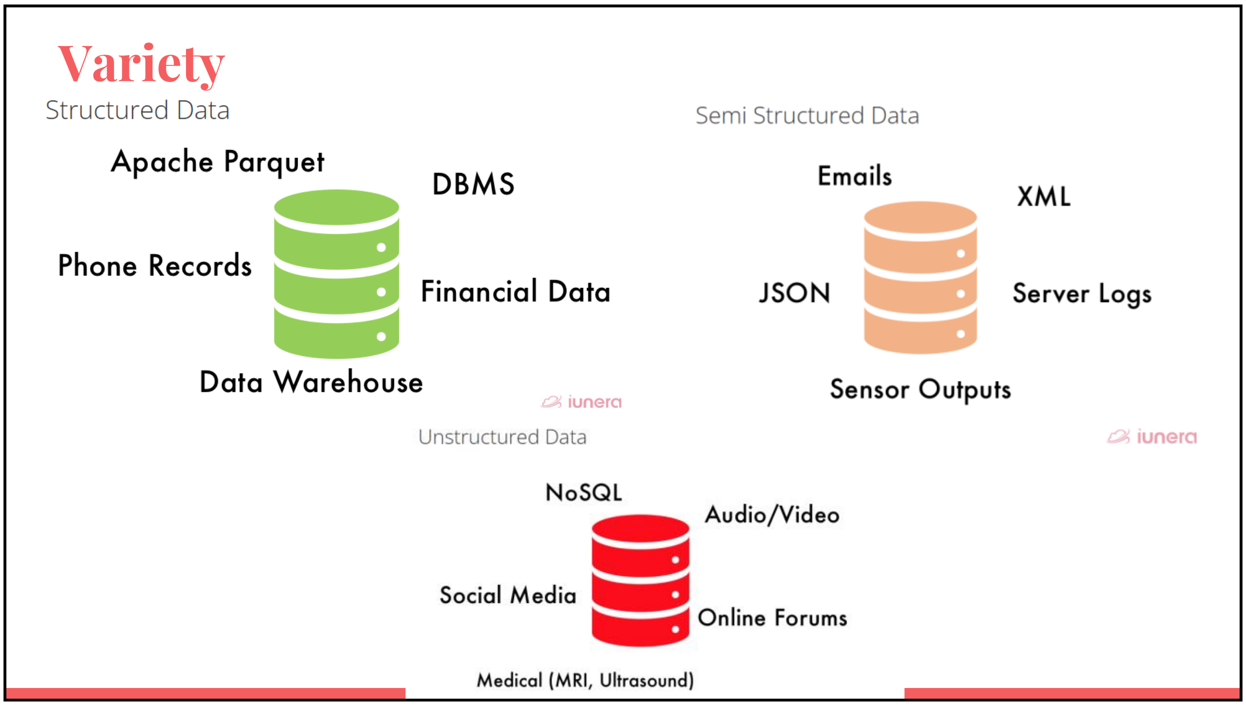
Byte =	8 Bits	1 bytes
Kilobyte =	1024 Bytes	1024 bytes
Megabyte =	1024 Kilobytes	1, 048, 576 bytes
Gigabyte =	1024 Megabytes	1, 073, 741, 824 bytes
Terabyte =	**1024 Gigabytes**	**1, 099, 511, 627, 776 bytes**
Petabyte =	**1024 Terabytes**	**1, 125, 899, 906, 842, 624 bytes**
Exabyte =	1024 Petabytes	1, 152, 921, 504, 606, 846, 976 bytes
Zettabyte =	1024 Exabytes	1, 180, 591, 620, 717, 411, 303, 424 bytes
Yottabyte =	1024 Zettabytes	1, 208, 925, 819, 614, 629, 174, 706, 176 bytes



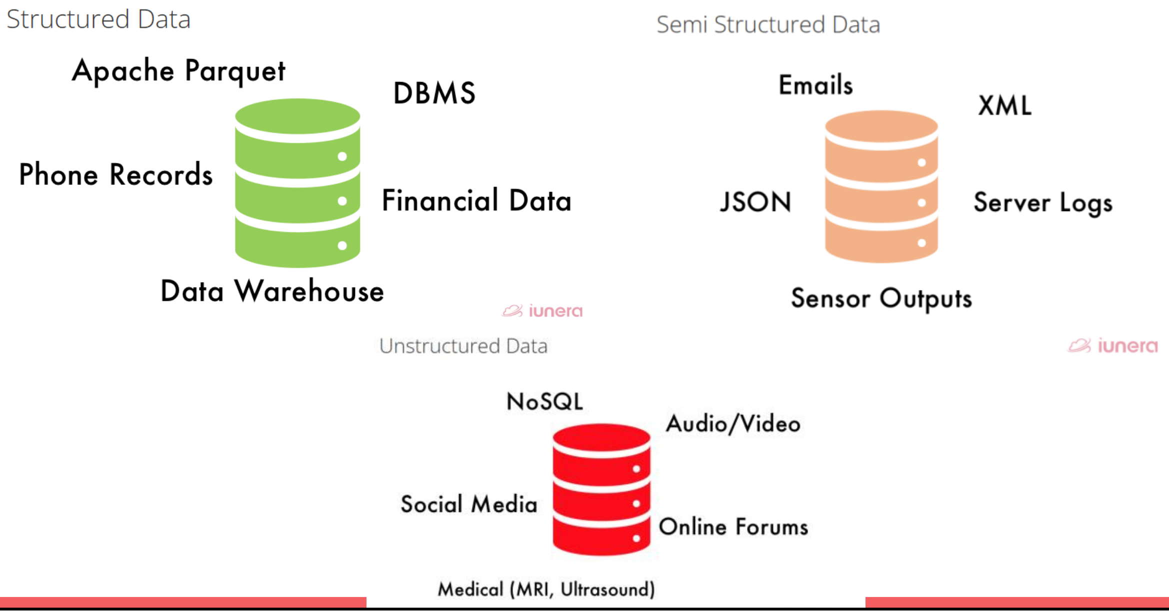
Variety (Complexity)

Variety refers to the diverse types and data sources generated and collected in today's digital world. It's not just limited to structured data like that of traditional databases numbers and strings. With the proliferation of social media, the internet of things, and other technologies, data is now available in various forms, such as text, images, audio, video, server logs, geographic and even sensor data.
This data is of three types: structured, semi-structured, and unstructured.





Variety



Variety

- Relational Data (Tables/Transaction/Legacy Data)
- Text Data (Web)
- Semi-structured Data (XML)
- Graph Data (Social Network, etc..)
- Streaming Data (You can only scan the data once)
- Big Public Data (online, weather, finance, etc)

To extract knowledge → all these types of data need to linked together

Velocity (Speed)

Data is being generated fast and need to be processed fast.

Need to do Online Data Analytics.

Late decisions missing opportunities.

Milliseconds to seconds response time.

Veracity

Data quality issues are particularly challenging in a big data context

Veracity refers to the quality, accuracy, integrity and credibility of data. Gathered data could have missing pieces, might be inaccurate or might not be able to provide real, valuable insight. Veracity, overall, refers to the level of trust there is in the collected data.

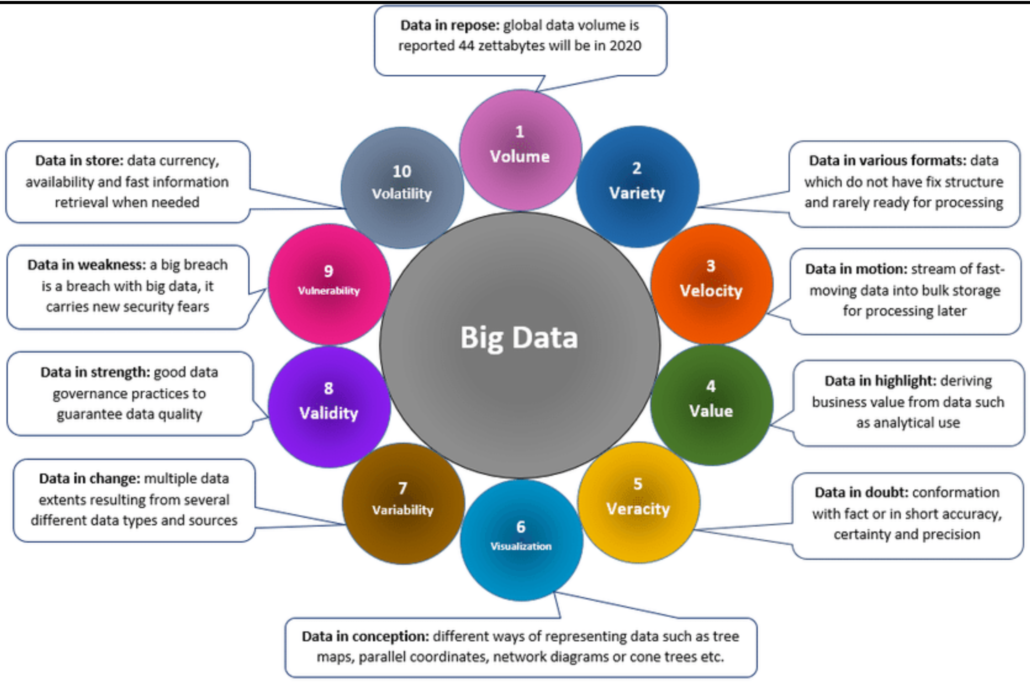
Data can sometimes become messy and difficult to use. A large amount of data can cause more confusion than insights if it's incomplete. For example, in the medical field, if data about what drugs a patient is taking is incomplete, the patient's life could be endangered.

Value

Provide value towards meaningful goals for the business.

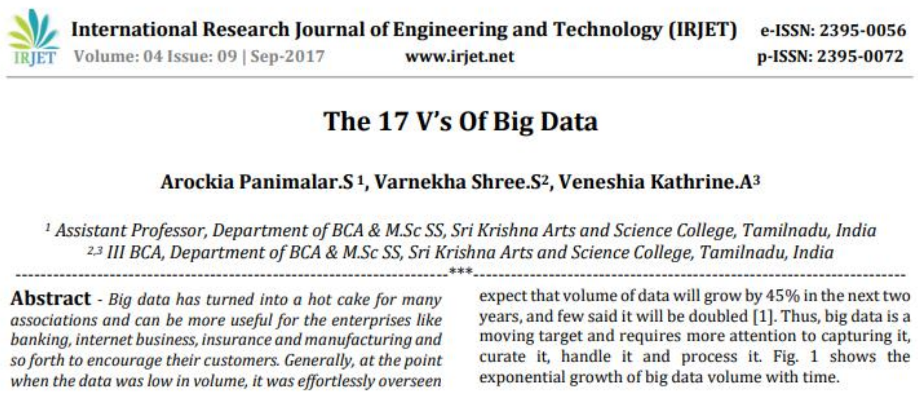
Value refers to the benefits that big data can provide, and it relates directly to what organizations can do with that collected data. Being able to pull value from big data is a requirement, as the value of big data increases significantly depending on the insights that can be gained from it.

10 V


Data in repose: global data volume is reported 44 zettabytes will be in 2020
1 Volume
2 Variety
3 Velocity
4 Value
5 Veracity
6 Visualization
7 Variability
8 Validity
9 Vulnerability
10 Volatility
Big Data
Data in various formats: data which do not have fix structure and rarely ready for processing
Data in motion: stream of fast-moving data into bulk storage for processing later
Data in highlight: deriving business value from data such as analytical use
Data in doubt: conformation with fact or in short accuracy, certainty and precision
Data in conception: different ways of representing data such as tree maps, parallel coordinates, network diagrams or cone trees etc.
Data in change: multiple data extents resulting from several different data types and sources
Data in strength: good data governance practices to guarantee data quality
Data in weakness: a big breach is a breach with big data, it carries new security fears
Data in store: data currency, availability and fast information retrieval when needed

And even 17 v's :-)

https://www.irjet.net/archives/V4/i9/IRJET-V4I957.pdf



International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 09 | Sep-2017 www.irjet.net p-ISSN: 2395-0072

The 17 V's Of Big Data

Arockia Panimalar.S 1, Varnekha Shree.S2, Veneshia Kathrine.A3

1 Assistant Professor, Department of BCA & M.Sc SS, Sri Krishna Arts and Science College, Tamilnadu, India
2,3 III BCA, Department of BCA & M.Sc SS, Sri Krishna Arts and Science College, Tamilnadu, India

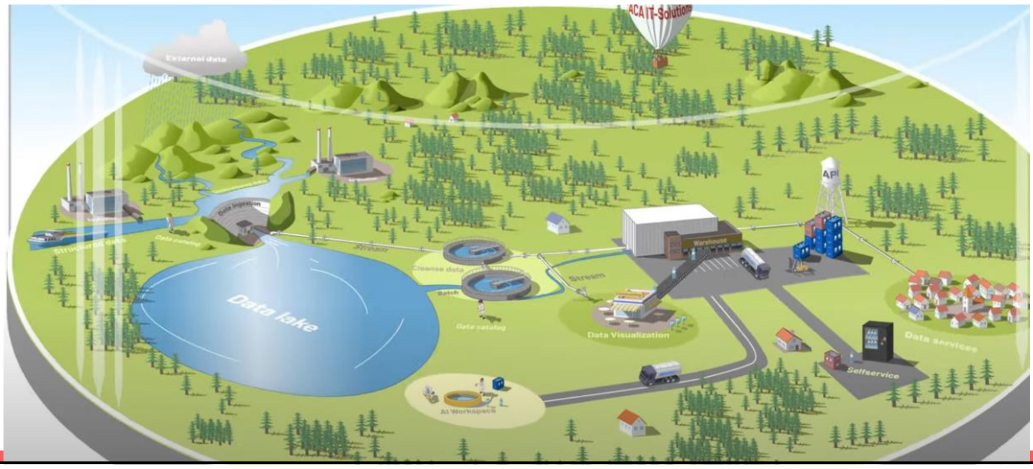
***

Abstract - *Big data has turned into a hot cake for many associations and can be more useful for the enterprises like banking, internet business, insurance and manufacturing and so forth to encourage their customers. Generally, at the point when the data was low in volume, it was effortlessly overseen*

expect that volume of data will grow by 45% in the next two years, and few said it will be doubled [1]. Thus, big data is a moving target and requires more attention to capturing it, curate it, handle it and process it. Fig. 1 shows the exponential growth of big data volume with time.

Youtube - What is a Data Architecture?

https://www.youtube.com/watch?v=64TpELBlqAk



Youtube - What is Data Mesh?

https://www.youtube.com/watch?v=PHKQ0i-nTOs

Data Lake

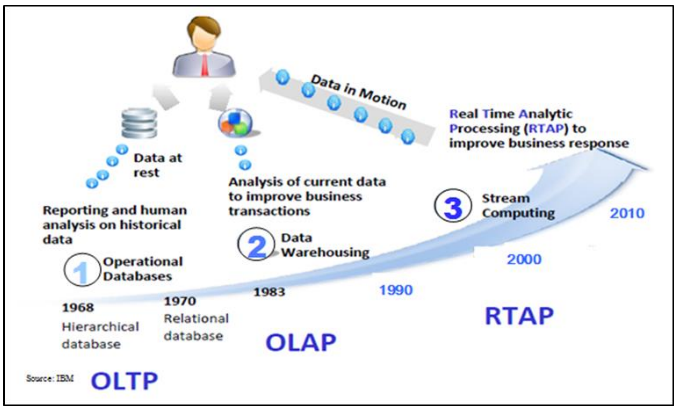
Implementation of “Big Data” may be using Data Lake:

A data lake is a system or repository of data stored in its natural/raw format, usually object blobs or files. A data lake is usually a single store of data including raw copies of source system data, sensor data, social data etc., and transformed data used for tasks such as reporting, visualization, advanced analytics and machine learning.

A data lake can include structured data from relational databases (rows and columns), semi-structured data (CSV, logs, XML, JSON), unstructured data (emails, documents, PDFs) and binary data (images, audio, video). A data lake can be established "on premises" (within an organization's data centers) or "in the cloud" (using cloud services from vendors such as Amazon, Microsoft, Oracle Cloud, or Google).

- Collect everything - all raw data is collected on write
- Flexible access - “Schema on read” allows adaptive/agile creation of connections

Analytics


Data in Motion
Real Time Analytic Processing (RTAP) to improve business response
Data at rest
Analysis of current data to improve business transactions
Reporting and human analysis on historical data
3 Stream Computing
2010
2 Data Warehousing
2000
1 Operational Databases
1983
1990
1968 Hierarchical database
1970 Relational database
RTAP
OLAP
Source: IBM
OLTP

- **OLTP**: Online Transaction Processing (DBMSs)
- **OLAP**: Online Analytical Processing (Data Warehousing)
- **RTAP**: Real-Time Analytics Processing (Big Data Architecture & technology)

MAD Landscape = Machine Learning, Artificial Intelligence & Data

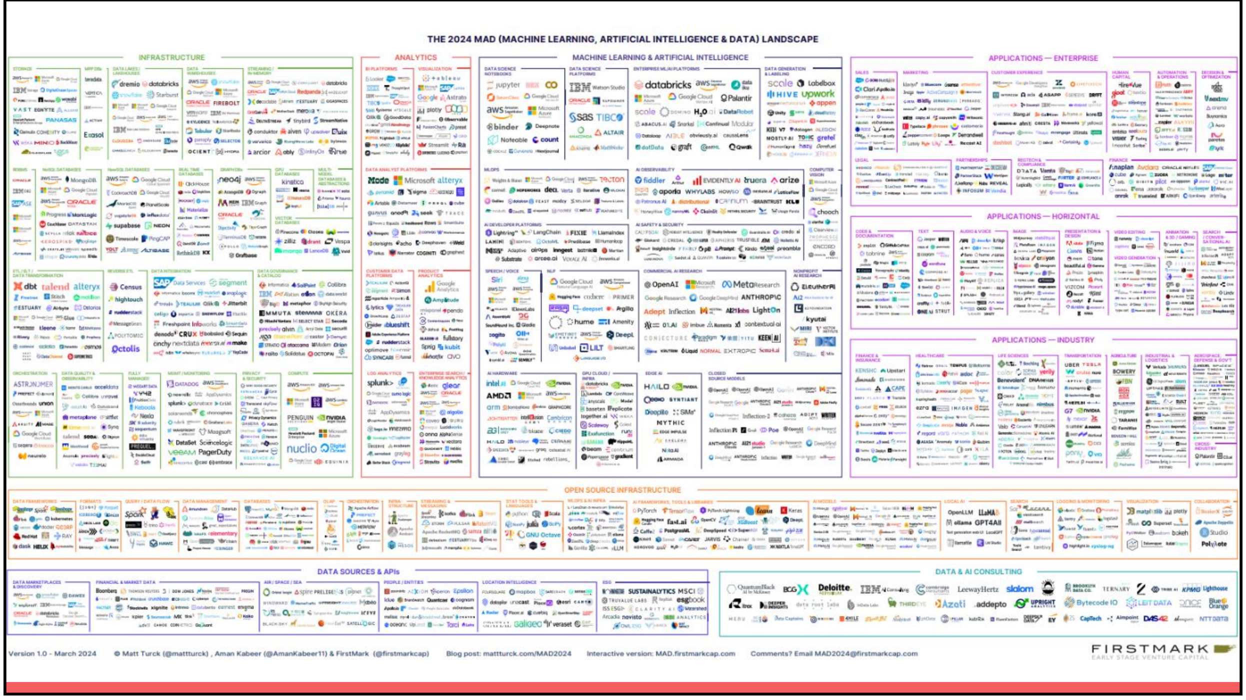
- https://mattturck.com/mad2024/
- https://mad.firstmark.com/
- https://mattturck.com/landscape/mad2024.pdf

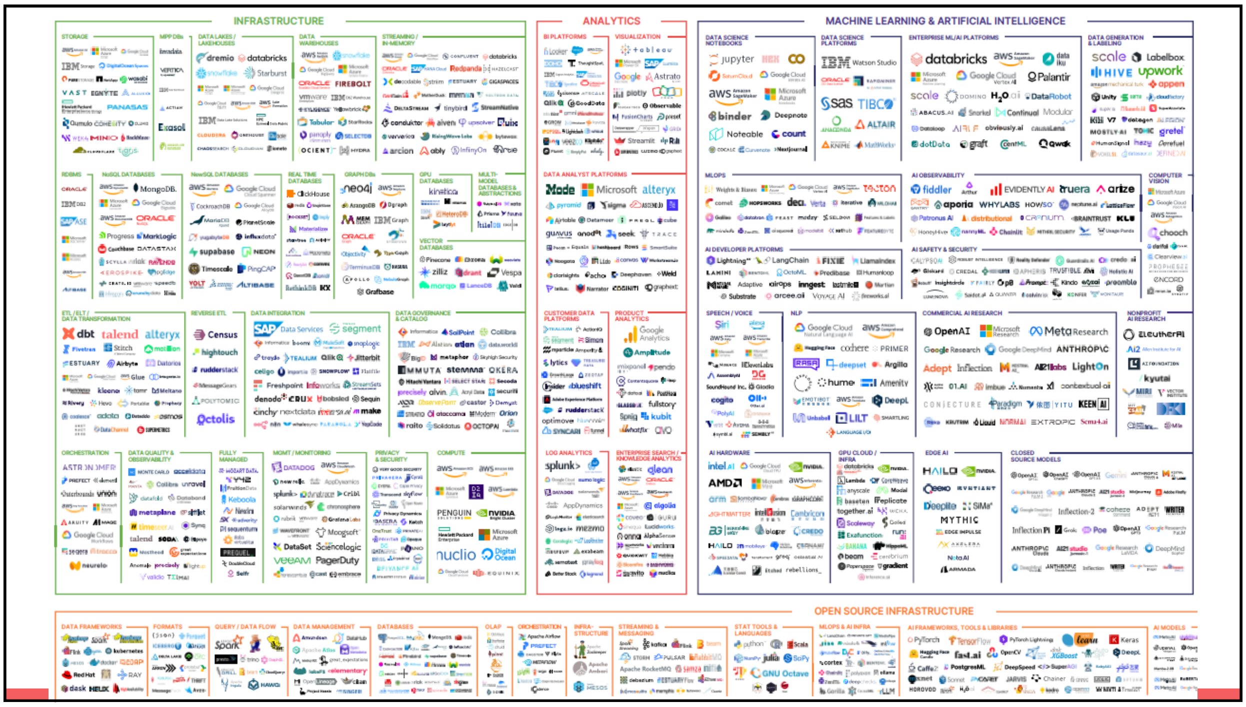
Infrastructure:
Storage | Data Lakes | Data Warehouses | Streaming |
RDBMs | NoSQL Databases | NewSQL Databases |
Real Time Databases | Graph DBs | ETL/ELT/Data Transformation | Compute

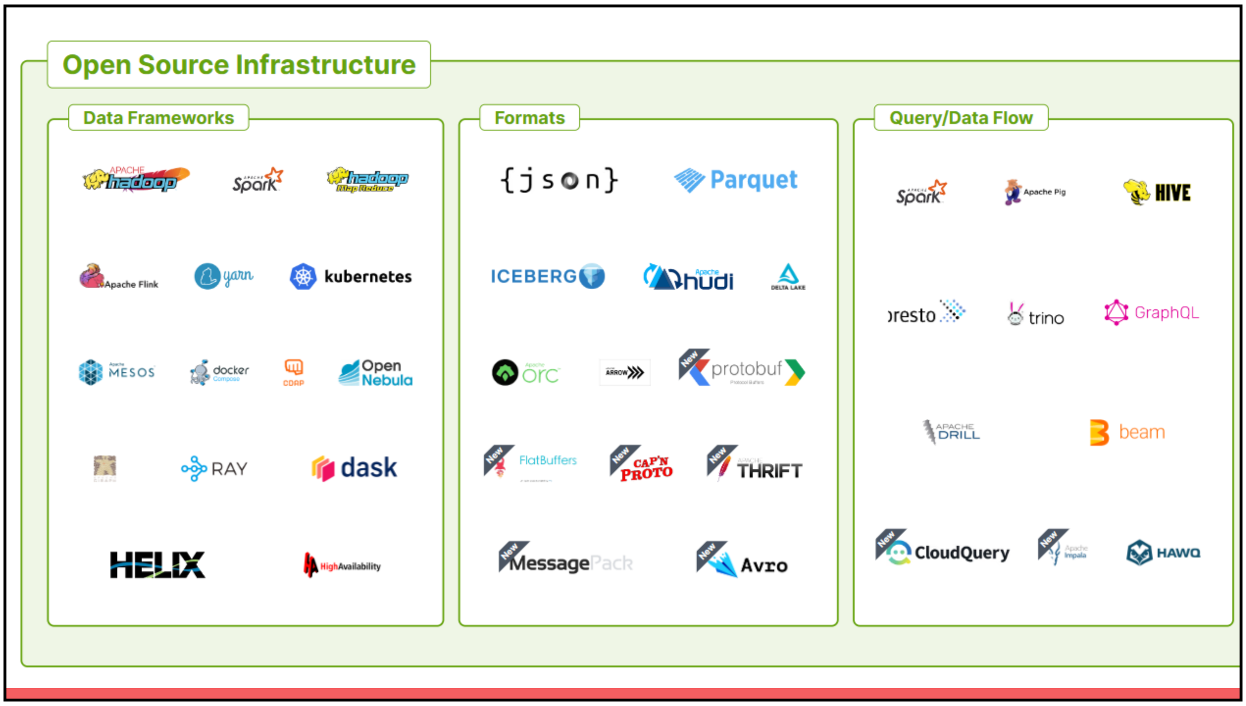
Analytics:
BI | Visualization | Data Analyst | Log Analytics | Enterprise Search

Machine Learning & Artificial Intelligence:
DS Notebooks | DS Platforms | Enterprise ML Platforms | MLOps | AI Developer Platforms |
Computer Vision | Speech | NLP | AI Research | AI Hardware | Models

Open Source Infrastructure:
Data Frameworks | File Formats | Query | Data Management | Databases | Orchestration |
Infrastructure | Streaming & Messaging | Stat Languages | MLOps & AI Infra |
AI Frameworks, Tools & Libraries | AI Models & Architectures | Search | Local AI |
Logging & Monitoring | Visualizations | Collaboration


THE 2024 MAD (MACHINE LEARNING, ARTIFICIAL INTELLIGENCE & DATA) LANDSCAPE
INFRASTRUCTURE
ANALYTICS
MACHINE LEARNING & ARTIFICIAL INTELLIGENCE
APPLICATIONS — ENTERPRISE
APPLICATIONS — HORIZONTAL
APPLICATIONS — INDUSTRY
OPEN SOURCE INFRASTRUCTURE
DATA SOURCES & APIs
DATA & AI CONSULTING
Version 1.0 - March 2024
© Matt Turck (@mattturck), Aman Kabeer (@AmanKabeer11) & FirstMark (@firstmarkcap)
Blog post: mattturck.com/MAD2024
Interactive version: MAD.firstmarkcap.com
Comments? Email MAD2024@firstmarkcap.com
FIRSTMARK
EARLY STAGE VENTURE CAPITAL




Open Source Infrastructure

Data Frameworks

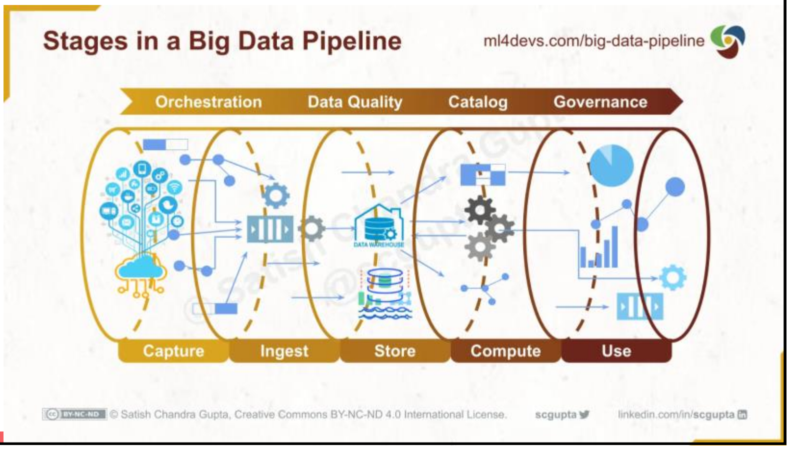
Formats

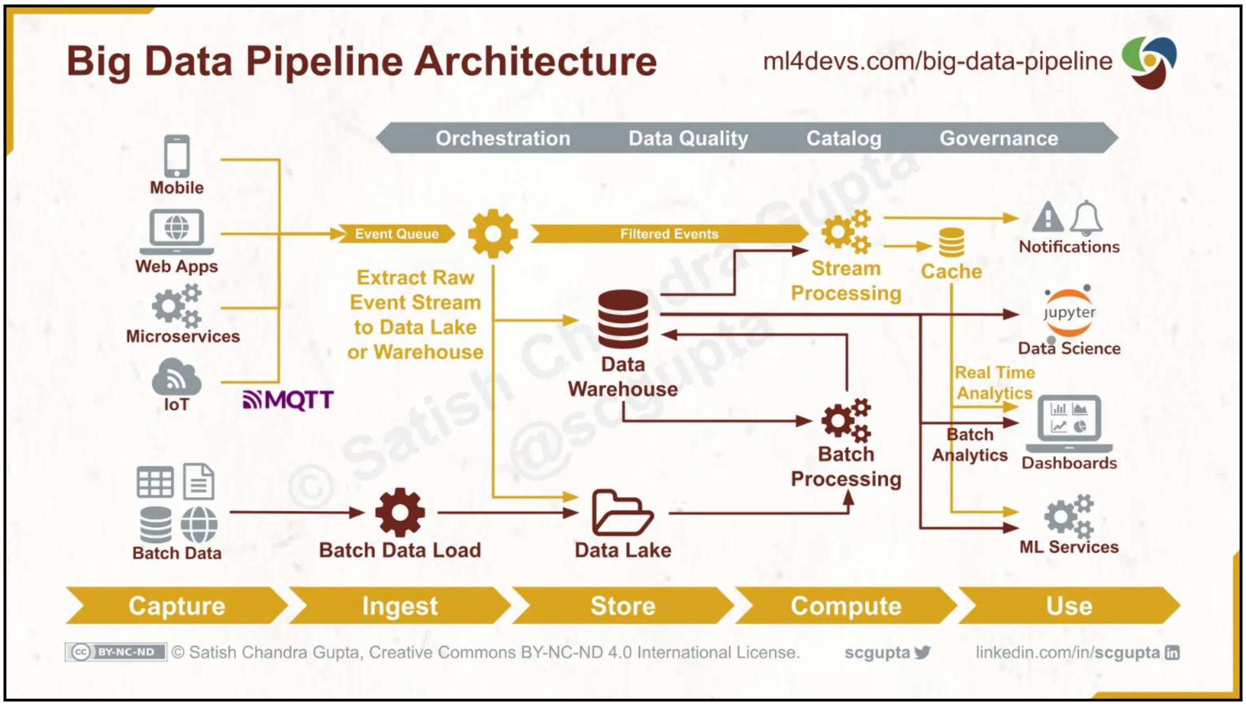
Query/Data Flow

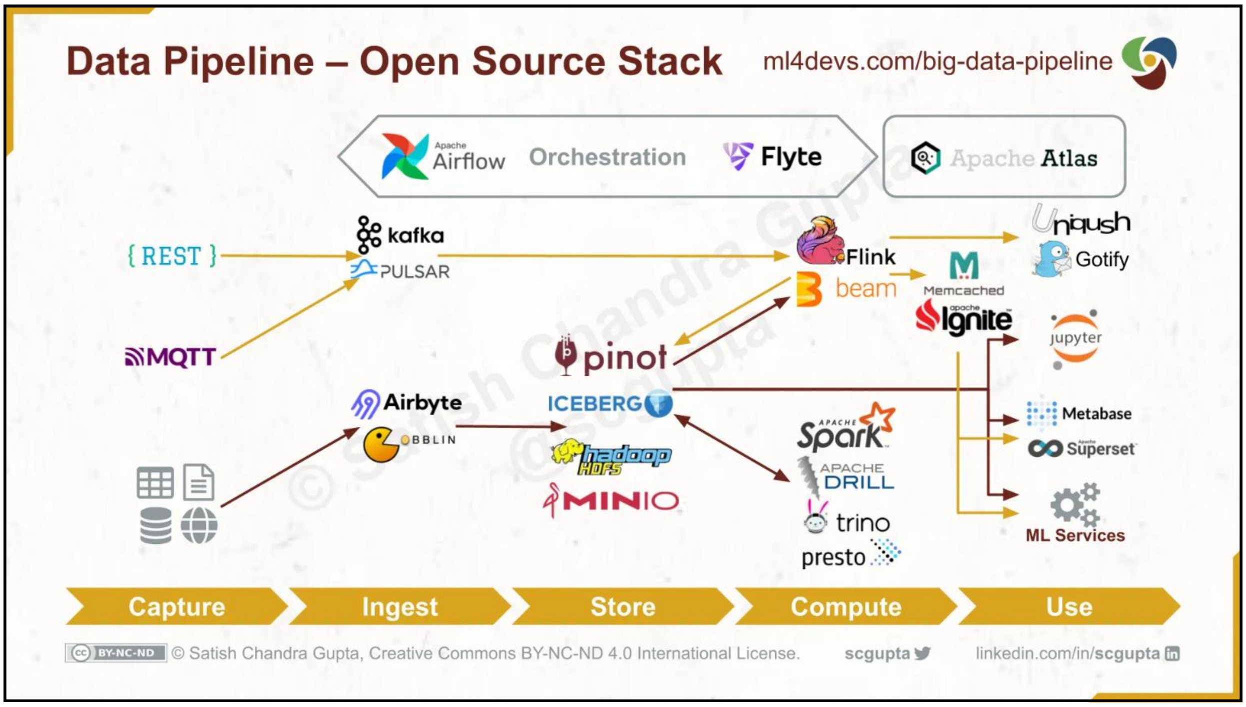
Data Pipe

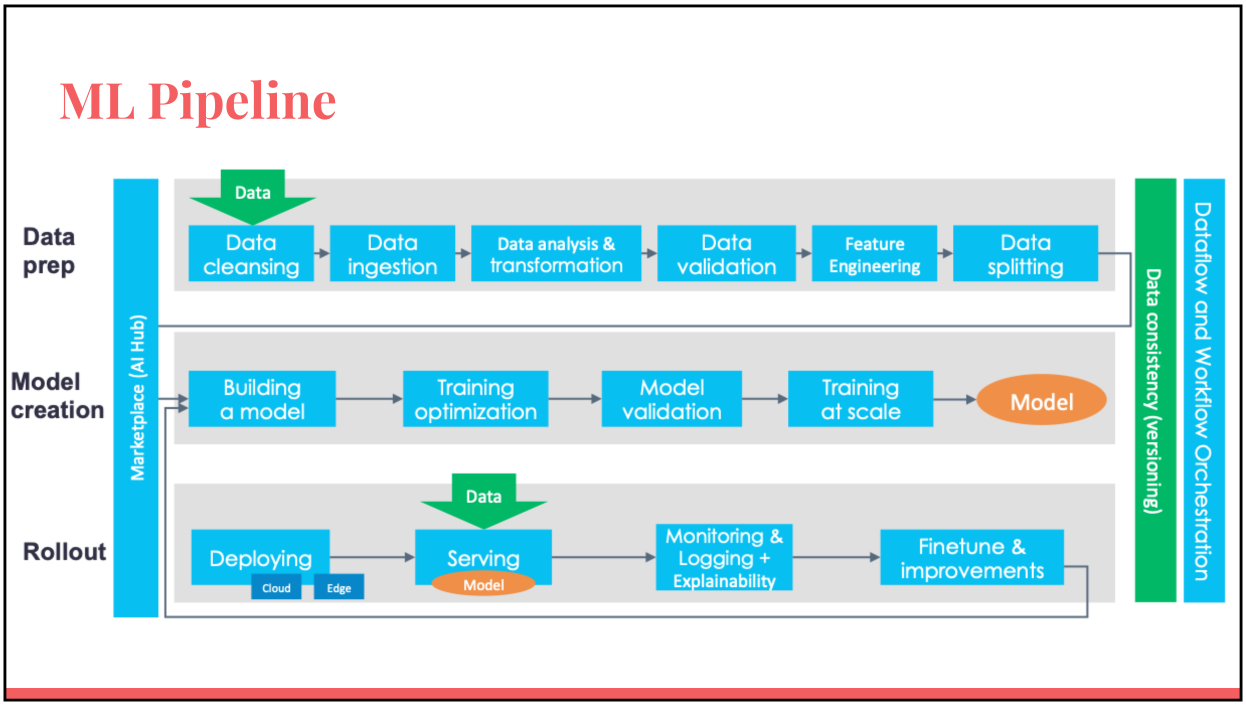
https://www.ml4devs.com/articles/scalable-efficient-big-data-analytics-machine-learning-pipeline-architecture-on-cloud/

For deploying big-data analytics, data science, and machine learning (ML) applications in the real world, analytics-tuning and model-training is only around 25% of the work. Approximately 50% of the effort goes into making data ready for analytics and ML. The remaining 25% effort goes into making insights and model inferences easily consumable at scale. The big data pipeline puts it all together. It is the railroad on which heavy and marvelous wagons of ML run. Long-term success depends on getting the data pipeline right.


Stages in a Big Data Pipeline
ml4devs.com/big-data-pipeline
Orchestration
Data Quality
Catalog
Governance
Capture
Ingest
Store
Compute
Use
© Satish Chandra Gupta, Creative Commons BY-NC-ND 4.0 International License.
scgupta
linkedin.com/in/scgupta




Data Pipeline – Open Source Stack
ml4devs.com/big-data-pipeline
Apache Airflow
Orchestration
Flyte
Apache Atlas
{ REST }
MQTT
kafka
PULSAR
Flink
beam
Memcached
apache Ignite
Unqush
Gotify
jupyter
pinot
Airbyte
GOBBLIN
ICEBERG
hadoop HDFS
MINIO
Spark
APACHE DRILL
trino
presto
Metabase
Superset
ML Services
Capture
Ingest
Store
Compute
Use
© Satish Chandra Gupta, Creative Commons BY-NC-ND 4.0 International License.
scgupta
linkedin.com/in/scgupta


ML Pipeline
Data prep
Model creation
Rollout
Marketplace (AI Hub)
Data
Data cleansing
Data ingestion
Data analysis & transformation
Data validation
Feature Engineering
Data splitting
Building a model
Training optimization
Model validation
Training at scale
Model
Deploying
Cloud
Edge
Data
Serving
Model
Monitoring & Logging + Explainability
Finetune & improvements
Data consistency (versioning)
Dataflow and Workflow Orchestration

Data Lake

A **data lake** is a centralized repository that ingests and stores large volumes of data in its original form. The data can then be processed and used as a basis for a variety of analytic needs. Due to its open, scalable architecture, a data lake can accommodate all types of data from any source, from structured (database tables, Excel sheets) to semi-structured (XML files, webpages) to unstructured (images, audio files, tweets), all without sacrificing fidelity.

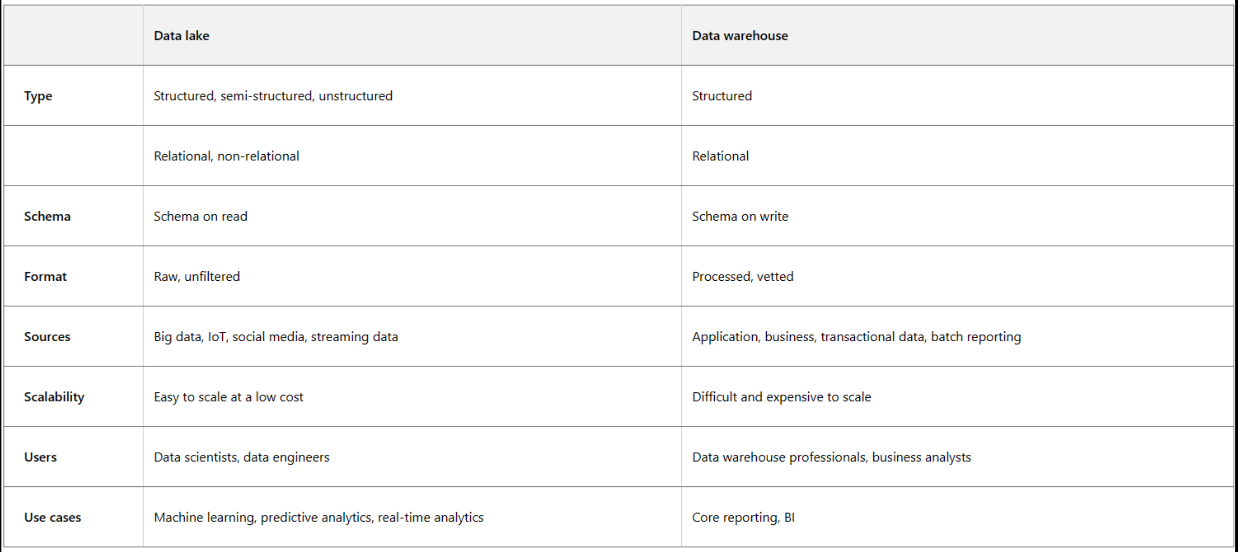
A **data lake** captures both relational and non-relational data from a variety of sources—business applications, mobile apps, IoT devices, social media, or streaming—without having to define the structure or schema of the data until it is read. Schema-on-read ensures that any type of data can be stored in its raw form. As a result, data lakes can hold a wide variety of data types, from structured to semi-structured to unstructured, at any scale. Their flexible and scalable nature make them essential for performing complex forms of data analysis using different types of compute processing tools like **Apache Spark**.

https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-data-lake

Data Warehouse

A **data warehouse** is relational in nature. The structure or schema is modeled or predefined by business and product requirements that are curated, conformed, and optimized for **SQL** query operations. While a data lake holds data of all structure types, including raw and unprocessed data, a data warehouse stores data that has been treated and transformed with a specific purpose in mind, which can then be used to source analytic or operational reporting. This makes data warehouses ideal for producing more standardized forms of BI analysis, or for serving a business use case that has already been defined.

Data Lake vs. Warehouse



	Data lake	Data warehouse
Type	Structured, semi-structured, unstructured	Structured
	Relational, non-relational	Relational
Schema	Schema on read	Schema on write
Format	Raw, unfiltered	Processed, vetted
Sources	Big data, IoT, social media, streaming data	Application, business, transactional data, batch reporting
Scalability	Easy to scale at a low cost	Difficult and expensive to scale
Users	Data scientists, data engineers	Data warehouse professionals, business analysts
Use cases	Machine learning, predictive analytics, real-time analytics	Core reporting, BI

Data Mart

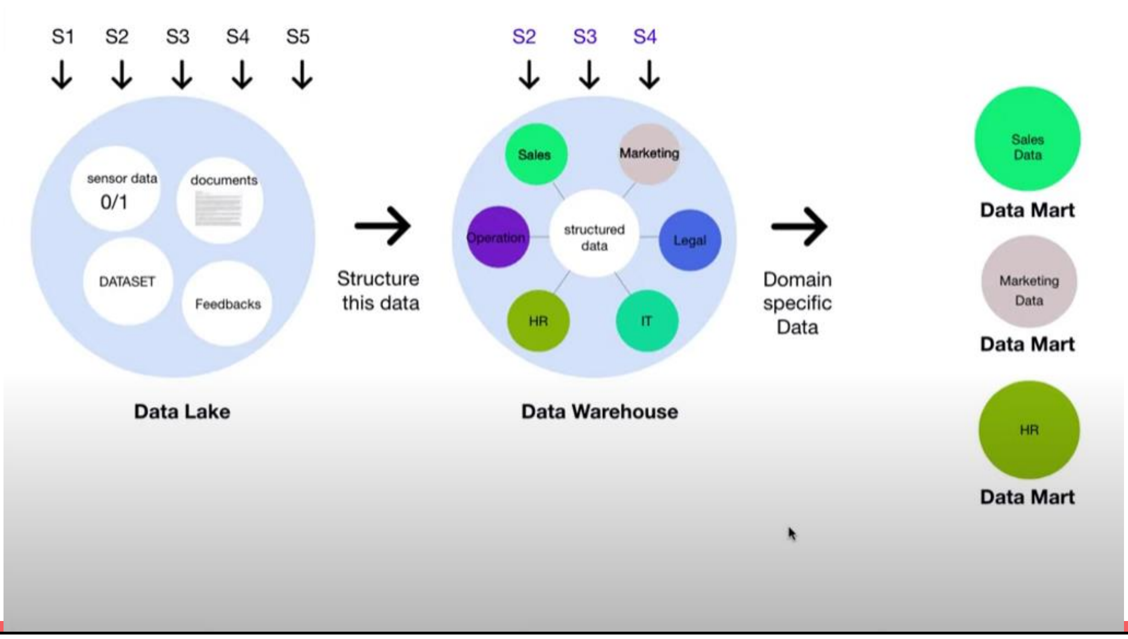
A **data mart** is a subset of a data warehouse oriented to a specific business line. Data marts contain repositories of summarized data collected for analysis on a specific section or unit within an organization, for example, the sales department.

A **data warehouse** is a large centralized repository of data that contains information from many sources within an organization. The collated data is used to guide business decisions through analysis, reporting, and data mining tools.

https://panoply.io/data-warehouse-guide/data-mart-vs-data-warehouse/

Data Mart vs. Warehouse

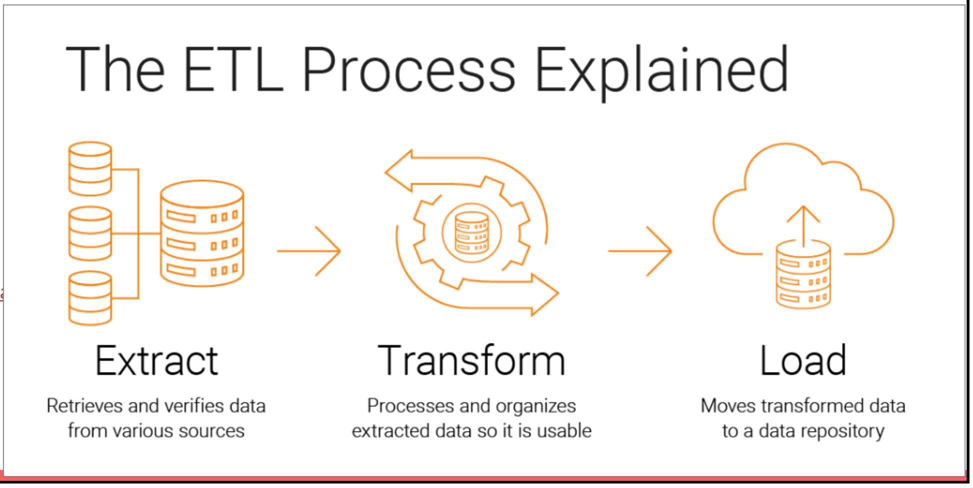
Data Mart	Data Warehouse
Focus: A single subject or functional organization area	Focus: Enterprise-wide repository of disparate data sources
Data Sources: Relatively few sources linked to one line of business	Data Sources: Many external and internal sources from different areas of an organization
Size: Less than 100 GB	Size: 100 GB minimum but often in the range of terabytes for large organizations
Normalization: No preference between a normalized and denormalized structure	Normalization: Modern warehouses are mostly denormalized for quicker data querying and read performance
Decision Types: Tactical decisions pertaining to particular business lines and ways of doing things	Decision Types: Strategic decisions that affect the entire enterprise
Cost: Typically from $10,000 upwards	Cost: Varies but often greater than $100,000
Setup Time: 3-6 months	Setup Time: At least a year for on-premise warehouses
Data Held: Typically summarized data	Data Held: Raw data, metadata, and summary data



ETL

ETL = “extract, transform and load” - It’s a three-step data integration process used by organizations to combine and synthesize raw data from multiple data sources into a data warehouse, data lake, data store, relational database or any other application.

https://www.informatica


The ETL Process Explained
Extract
Retrieves and verifies data from various sources
Transform
Processes and organizes extracted data so it is usable
Load
Moves transformed data to a data repository

ETL

Extract
Extraction is the first phase of “extract, transform, load.” Data is collected from one or more data sources. It is then held in temporary storage, where the next two steps are executed.
During extraction, validation rules are applied. This tests whether data meets the requirements of its destination. Data that fails validation is rejected and doesn’t continue to the next step.

Transform
In the transformation phase, data is processed to make its values and structure conform consistently with its intended use case. The goal of transformation is to make all data fit within a uniform schema before it moves on to the last step.
Typical transformations include aggregators, data masking, expression, joiner, filter, lookup, rank, router, union, XML, Normalizer, H2R, R2H and web service. This helps to normalize, standardize and filter data. It also makes the data fit for consumption for analytics, business functions and other downstream activities.

Load
Finally, the load phase moves the transformed data into a permanent target system. This could be a target database, data warehouse, data store, data hub or data lake — on-premises or in the cloud. Once all the data has been loaded, the process is complete.
Many organizations regularly perform this process to keep their data warehouse updated.

ETL vs. ELT

Extract transform load and extract load transform are two different data integration processes. They use the same steps in a different order for different data management functions.

Both ELT and ETL extract raw data from different data sources. Examples include an enterprise resource planning (ERP) platform, social media platform, Internet of Things (IoT) data, spreadsheet and more. With ELT, raw data is then loaded directly into the target data warehouse, data lake, relational database or data store. This allows data transformation to happen as required. It also lets you load datasets from the source. With ETL, after the data is extracted, it is then defined and transformed to improve data quality and integrity. Then, it is subsequently loaded into a data repository where it can be used.

Which should you use? Consider ETL if you're creating data repositories that are smaller, need to be retained for a longer period and don't need to be updated very often. ELT is best if you're dealing with high-volume datasets and big data management in real-time.

Batch Processing vs. Real Time Processing

Batch Processing system is an efficient way of processing large volumes of data. where a group of transactions is collected over a period of time. Data is collected, entered, processed and then the batch results are produced. Process large volumes of data in batches, typically overnight or on a schedule.

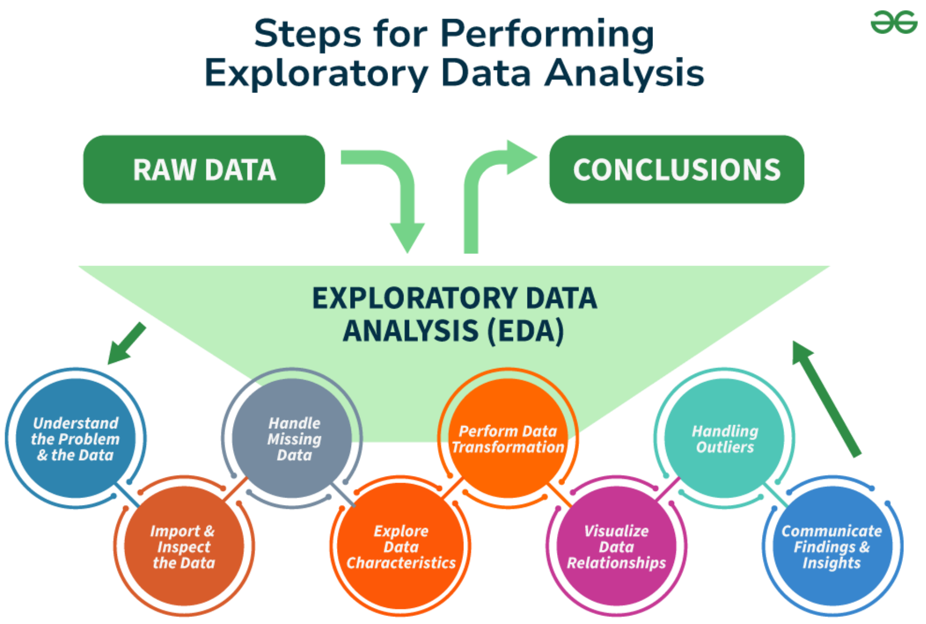
Real Time Processing systems are very fast and quick respondent systems. These systems are used in an environment where a large number of events must be accepted and processed in a short time. Real time processing requires quick transaction and characterized by supplying immediate response. For example, defense application systems like as RADAR, etc. Process data as it arrives, in real-time or near-real-time.

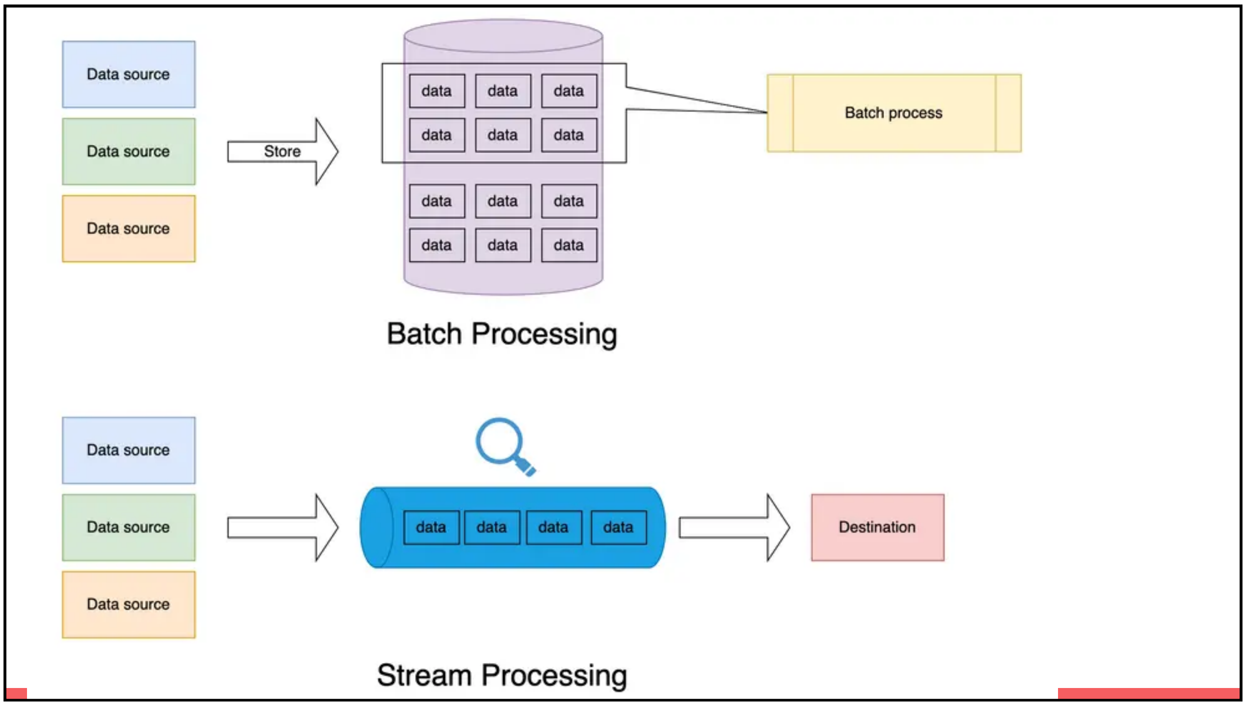
https://www.geeksforgeeks.org/difference-between-batch-processing-and-real-time-processing-system/

EDA = Exploratory data analysis

- **Distribution of Data**: Examining the distribution of data points to understand their range, central tendencies (mean, median), and dispersion (variance, standard deviation).
- **Graphical Representations**: Utilizing charts such as histograms, box plots, scatter plots, and bar charts to visualize relationships within the data and distributions of variables.
- **Outlier Detection**: Identifying unusual values that deviate from other data points. Outliers can influence statistical analyses and might indicate data entry errors or unique cases.
- **Correlation Analysis**: Checking the relationships between variables to understand how they might affect each other. This includes computing correlation coefficients and creating correlation matrices.
- **Handling Missing Values**: Detecting and deciding how to address missing data points, whether by imputation or removal, depending on their impact and the amount of missing data.
- **Summary Statistics**: Calculating key statistics that provide insight into data trends and nuances.
- **Testing Assumptions**: Many statistical tests and models assume the data meet certain conditions (like normality or homoscedasticity). EDA helps verify these assumptions.

EDA

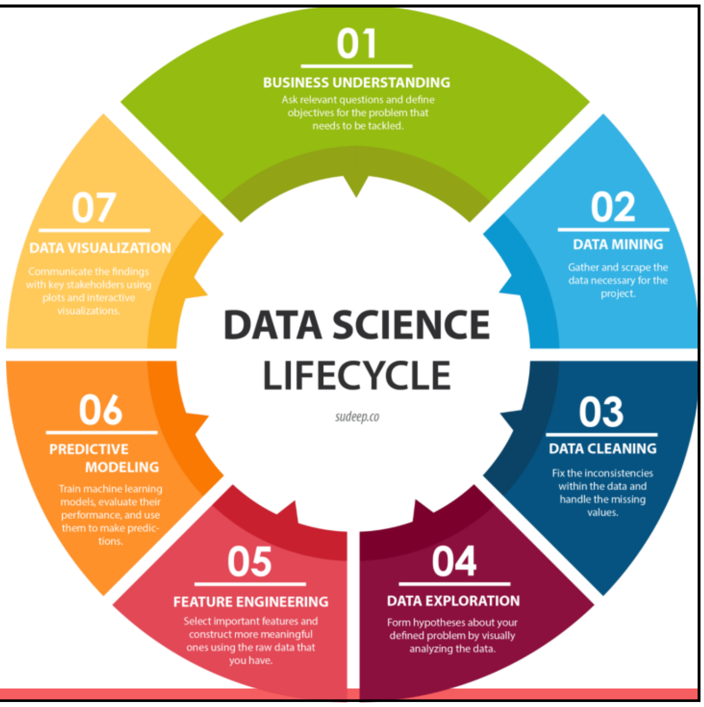

Steps for Performing Exploratory Data Analysis
RAW DATA
CONCLUSIONS
EXPLORATORY DATA ANALYSIS (EDA)
Understand the Problem & the Data
Import & Inspect the Data
Handle Missing Data
Explore Data Characteristics
Perform Data Transformation
Visualize Data Relationships
Handling Outliers
Communicate Findings & Insights



Data Life Cycle

- Define Question
- Find / Collect / Create data
- Store Data
- Extract Data
- Clean Data
- Verify Data
- Analyse Data
- Frame the Context
- Visualize / Dashboards
- Present results

Data Life Cycle


01
BUSINESS UNDERSTANDING
Ask relevant questions and define objectives for the problem that needs to be tackled.
02
DATA MINING
Gather and scrape the data necessary for the project.
03
DATA CLEANING
Fix the inconsistencies within the data and handle the missing values.
04
DATA EXPLORATION
Form hypotheses about your defined problem by visually analyzing the data.
05
FEATURE ENGINEERING
Select important features and construct more meaningful ones using the raw data that you have.
06
PREDICTIVE MODELING
Train machine learning models, evaluate their performance, and use them to make predictions.
07
DATA VISUALIZATION
Communicate the findings with key stakeholders using plots and interactive visualizations.
DATA SCIENCE LIFECYCLE
sudeep.co

Data Life Cycle

Plan & Design: Plan processes from onboarding to project closure and data resources

Collect & Create: Organization and integration of data sets and collection processes

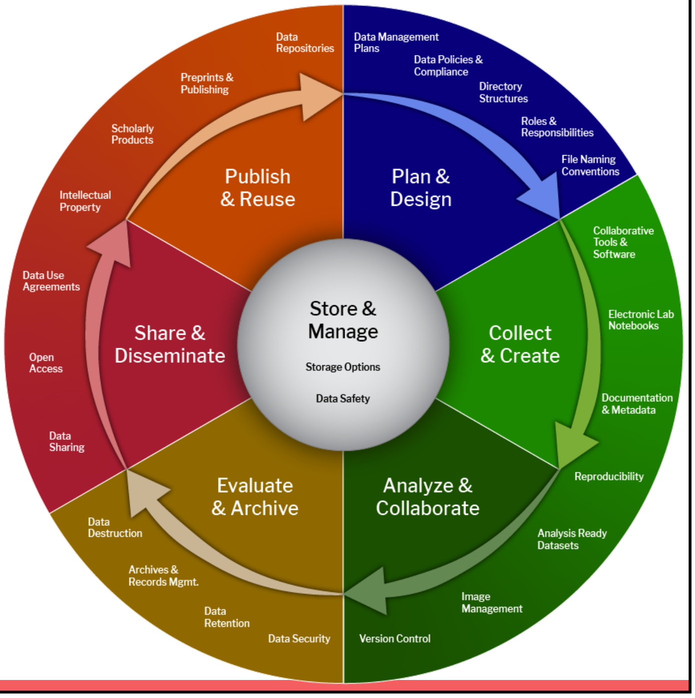
Analyze & Collaborate: Processing and analyzing data should be collaborative and documented

Store & Manage: Each stage of the Biomedical Data Lifecycle revolves around the management of data storage

Evaluate & Archive: Identify essential research records and evaluate for retention

Share & Disseminate: Establishing and supporting the reach and impact of your data

Publish & Reuse: Ensuring the broad utility of your research data efforts for other researchers



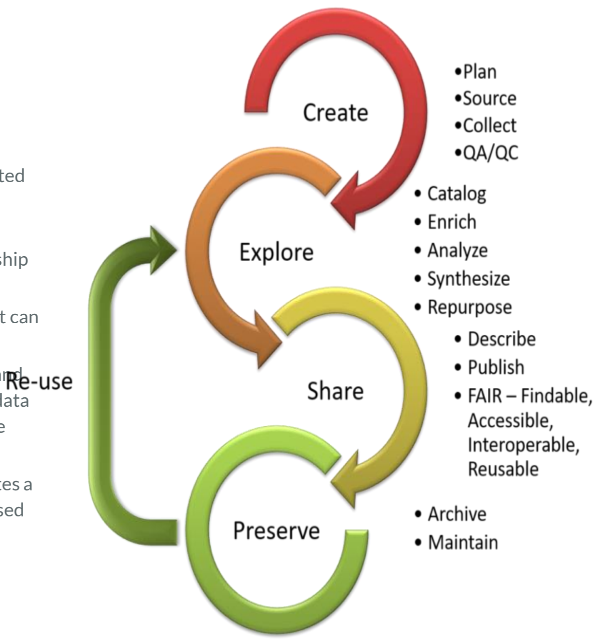
Create: The data lifecycle begins with a researcher(s) developing a concept for a study; once a study concept is developed, data is collected for that study.

Explore: In this phase, you clean, analyze and visualize your data to support your research and demonstrate evidence. This is the authorship phase of storytelling.

Share: After data is collected, it is processed for distribution so that it can be archived and used by other researchers later.

Preserve: Once data reaches the distribution stage of the lifecycle, and you have produced documentation, including a clear and accessible data usage license, it is stored in a location (i.e., repository, registry) where other researchers can discover it.

Re-Use: Data discovery leads to the repurposing of data, which creates a continual loop back to the data processing stage, where the repurposed data is archived and distributed for discovery.



Pandas

pandas provides data structures for in-memory analytics, which makes using pandas to analyze datasets that are larger than memory datasets somewhat tricky.

Scaling to large datasets:

- Load less data (less columns)
- Use efficient datatypes
- Use chunking - We can implement an out-of-core value_counts() for example. The peak memory usage of this workflow is the single largest chunk, plus a small series storing the unique value counts up to this point. As long as each individual file fits in memory, this will work for arbitrary-sized datasets. Chunking works well when the operation you're performing requires zero or minimal coordination between chunks. For more complicated workflows, you're better off using other libraries

  https://pandas.pydata.org/community/ecosystem.html#out-of-core

Big Data - Out of Core

Cylon

Cylon is a fast, scalable, distributed memory parallel runtime with a pandas like Python DataFrame API. ”Core Cylon” is implemented with C++ using Apache Arrow format to represent the data in-memory. Cylon DataFrame API implements most of the core operators of pandas such as merge, filter, join, concat, group-by, drop_duplicates, etc. These operators are designed to work across thousands of cores to scale applications. It can interoperate with pandas DataFrame by reading data from pandas or converting data to pandas so users can selectively scale parts of their pandas DataFrame applications.

Modin

The modin.pandas DataFrame is a parallel and distributed drop-in replacement for pandas. This means that you can use Modin with existing pandas code or write new code with the existing pandas API. Modin can leverage your entire machine or cluster to speed up and scale your pandas workloads, including traditionally time-consuming tasks like ingesting data (read_csv, read_excel, read_parquet, etc.).

Big Data - Out of Core

Vaex

Increasingly, packages are being built on top of pandas to address specific needs in data preparation, analysis and visualization. Vaex is a python library for Out-of-Core DataFrames (similar to Pandas), to visualize and explore big tabular datasets. It can calculate statistics such as mean, sum, count, standard deviation etc, on an N-dimensional grid up to a billion (10^9) objects/rows per second. Visualization is done using histograms, density plots and 3d volume rendering, allowing interactive exploration of big data. Vaex uses memory mapping, zero memory copy policy and lazy computations for best performance (no memory wasted).

Dask / Dask-ML

Dask is a flexible parallel computing library for analytics. Dask provides a familiar DataFrame interface for out-of-core, parallel and distributed computing.

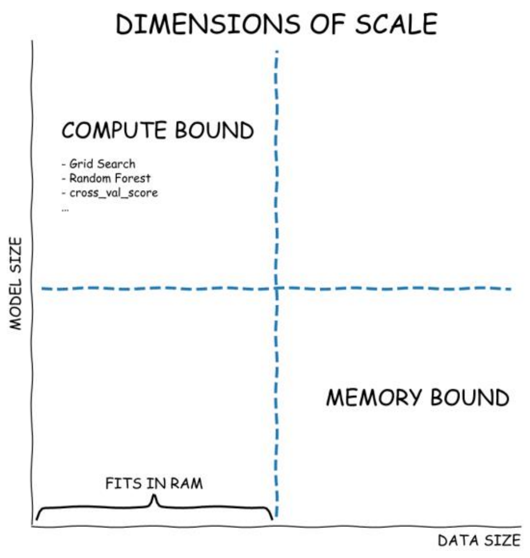
Dask-ML enables parallel and distributed machine learning using Dask alongside existing machine learning libraries like Scikit-Learn, XGBoost, and TensorFlow.

DASK-ML Dimensions of Scale

Challenge 1: Scaling Model Size Compute

The first kind of scaling challenge comes when from your models growing so large or complex that it affects your workflow (shown along the vertical axis above). Under this scaling challenge tasks like model training, prediction, or evaluation steps will (eventually) complete, they just take too long. You've become compute bound.

To address these challenges you'd continue to use the collections you know and love (like the NumPy ndarray, pandas DataFrame, or XGBoost DMatrix) and use a Dask Cluster to parallelize the workload on many machines. The parallelization can occur through one of our integrations (like Dask's joblib backend to parallelize Scikit-Learn directly) or one of Dask-ML's estimators (like our hyper-parameter optimizers).



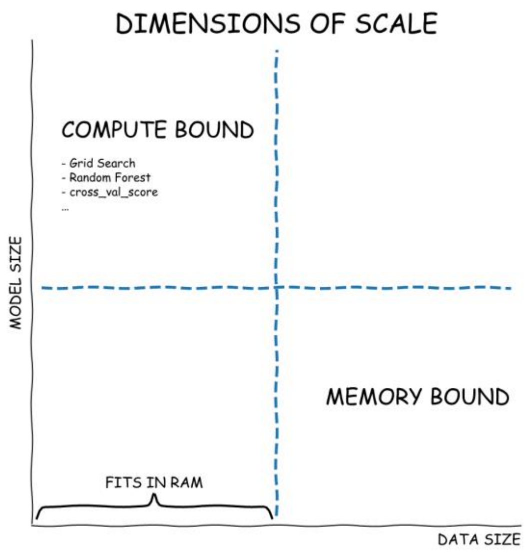
DASK-ML Dimensions of Scale

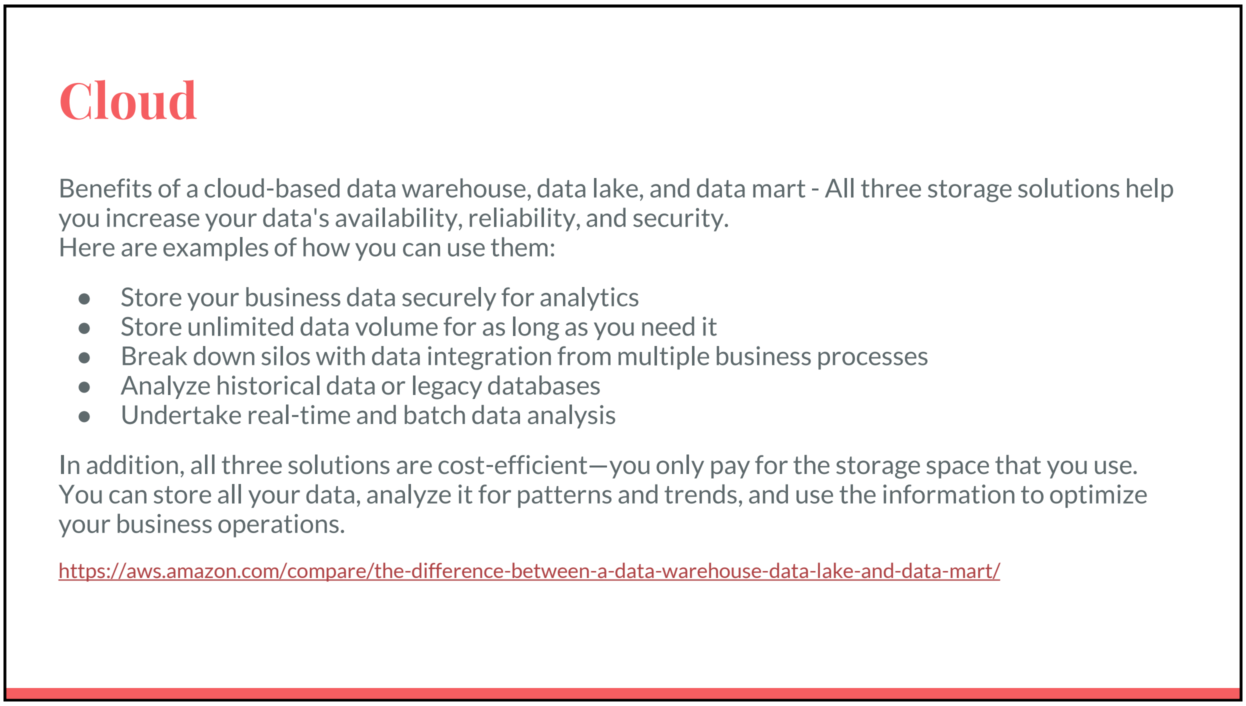
Challenge 2: Scaling Data Size Memory

The second type of scaling challenge people face is when their datasets grow larger than RAM (shown along the horizontal axis above). Under this scaling challenge, even loading the data into NumPy or pandas becomes impossible.

To address these challenges, you'd use Dask's one of Dask's high-level collections like (Dask Array, Dask DataFrame or Dask Bag) combined with one of Dask-ML's estimators that are designed to work with Dask collections. For example you might use Dask Array and one of our preprocessing estimators in dask_ml.preprocessing, or one of our ensemble methods in dask_ml.ensemble.

NOTE: Not Everyone needs Scalable Machine Learning. It's worth emphasizing that not everyone needs scalable machine learning. Tools like sampling can be effective.





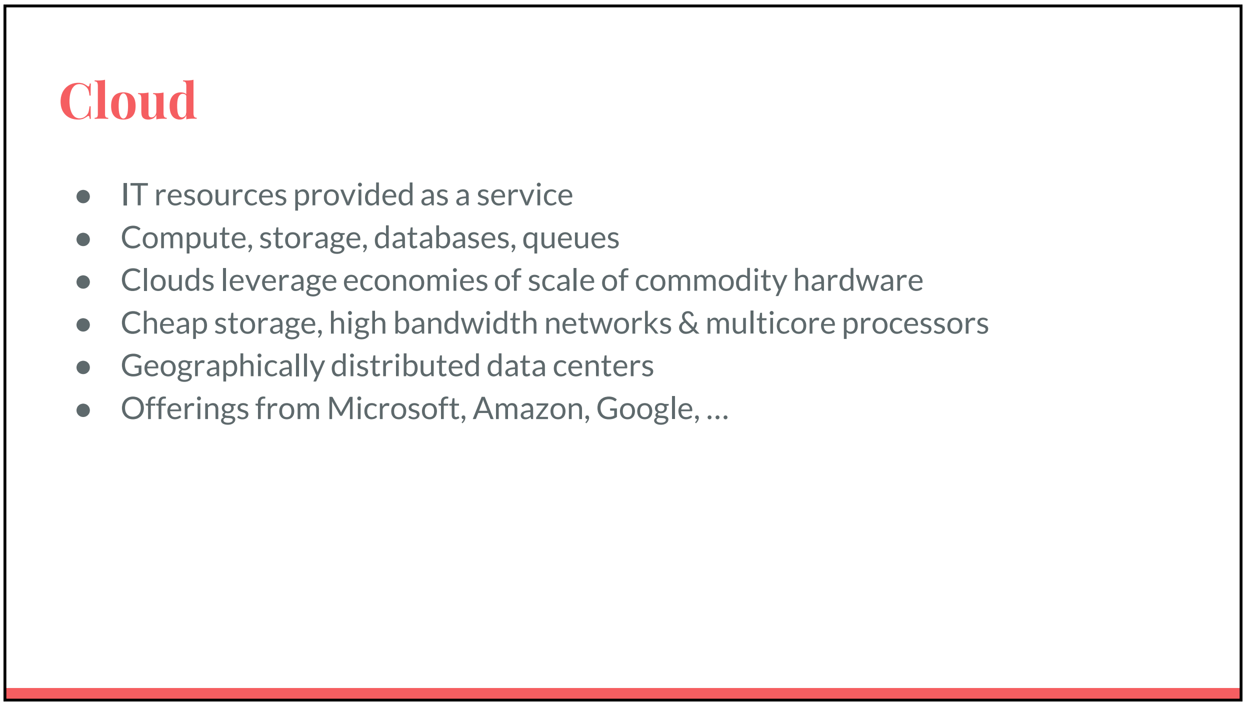
Cloud

Benefits of a cloud-based data warehouse, data lake, and data mart - All three storage solutions help you increase your data's availability, reliability, and security.
Here are examples of how you can use them:

- Store your business data securely for analytics
- Store unlimited data volume for as long as you need it
- Break down silos with data integration from multiple business processes
- Analyze historical data or legacy databases
- Undertake real-time and batch data analysis

In addition, all three solutions are cost-efficient—you only pay for the storage space that you use. You can store all your data, analyze it for patterns and trends, and use the information to optimize your business operations.

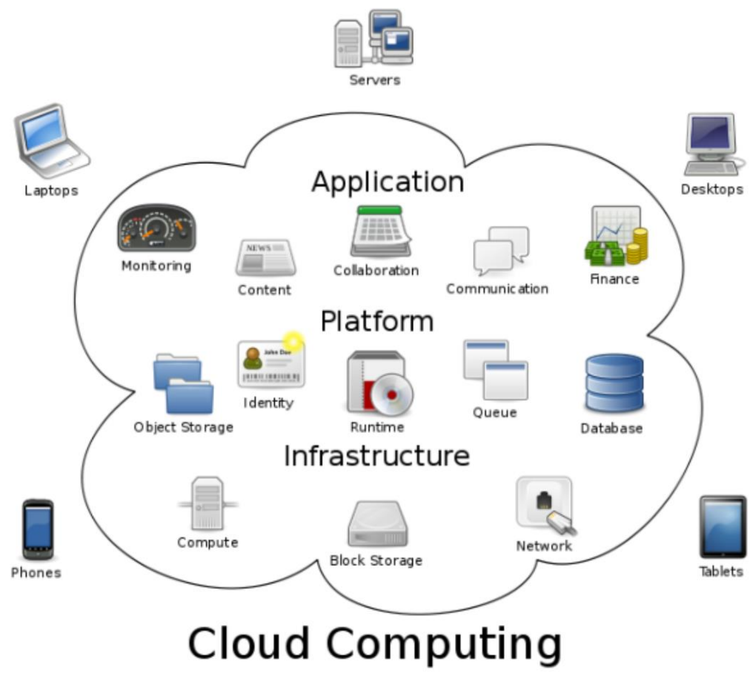
https://aws.amazon.com/compare/the-difference-between-a-data-warehouse-data-lake-and-data-mart/



Cloud

- IT resources provided as a service
- Compute, storage, databases, queues
- Clouds leverage economies of scale of commodity hardware
- Cheap storage, high bandwidth networks & multicore processors
- Geographically distributed data centers
- Offerings from Microsoft, Amazon, Google, …

Cloud



Cloud

Benefits

Virtualization

Iaas paas