1&1 AI Workshop

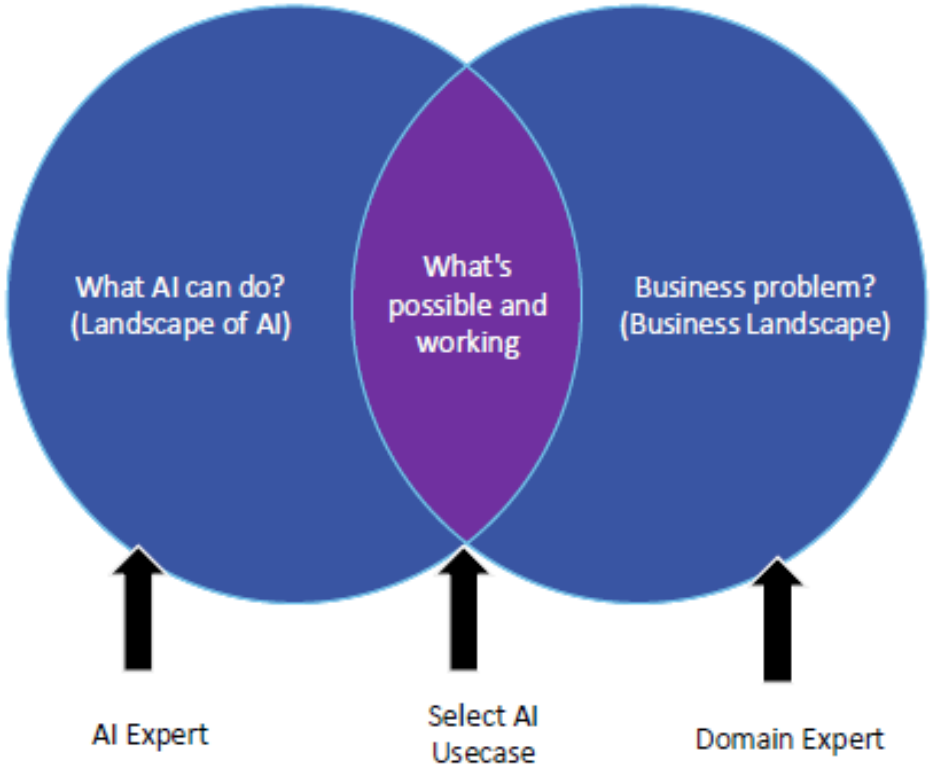
Tag 2

CGI

Themen

1. 09:30 – 10:45 Business cases
2. 11:00 – 12:00 & 13:30 – 14:45 Kafka & ML
3. 15:00 – 16:30 Python & NLP (JIRA)
4. 16:30 – 16:45 Advanced models
5. 16:45 – 17:00 Feedback

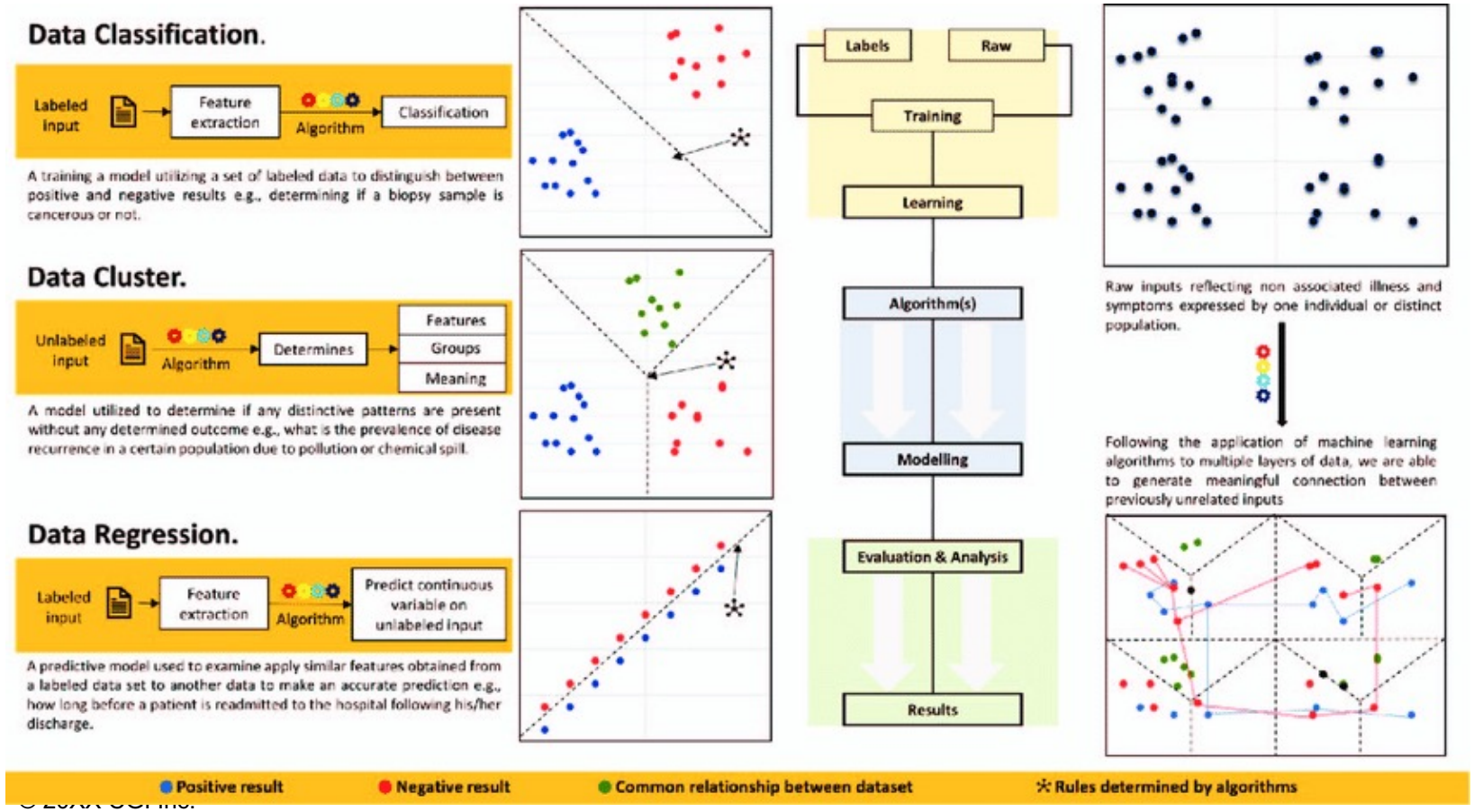
Business cases



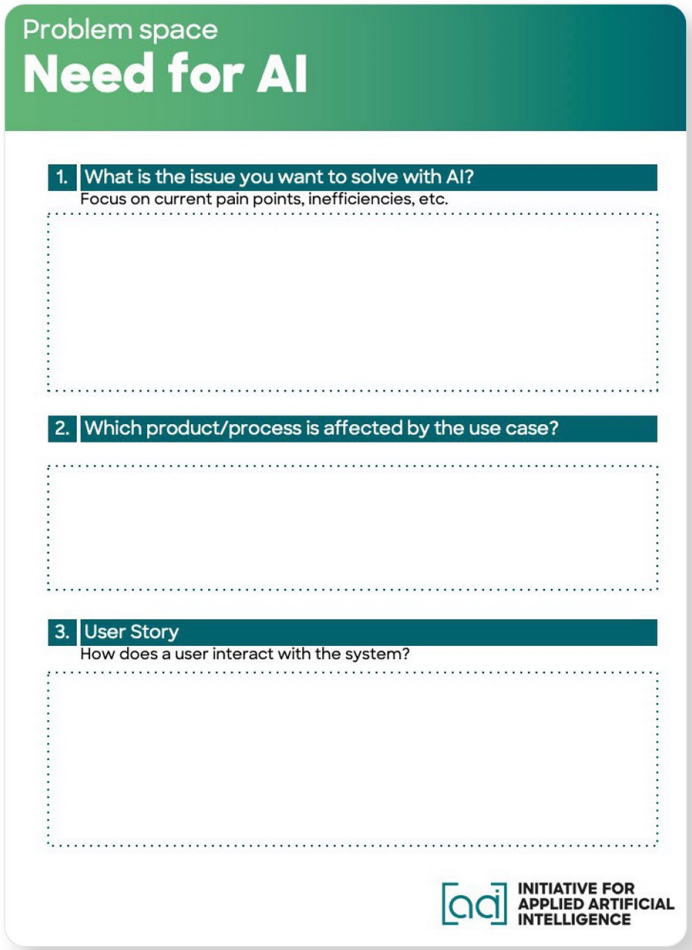
- Strategische Ausrichtung
- Auswirkungen und Erfolgskriterien
- Datenverfügbarkeit
- Ethische oder regulatorische Fragen
- Verfügbarkeit von Talenten
- Infrastruktur
- Technologie
- Technische Komplexität
- Herausforderungen bei der Integration
- Herausforderungen bei der Implementierung

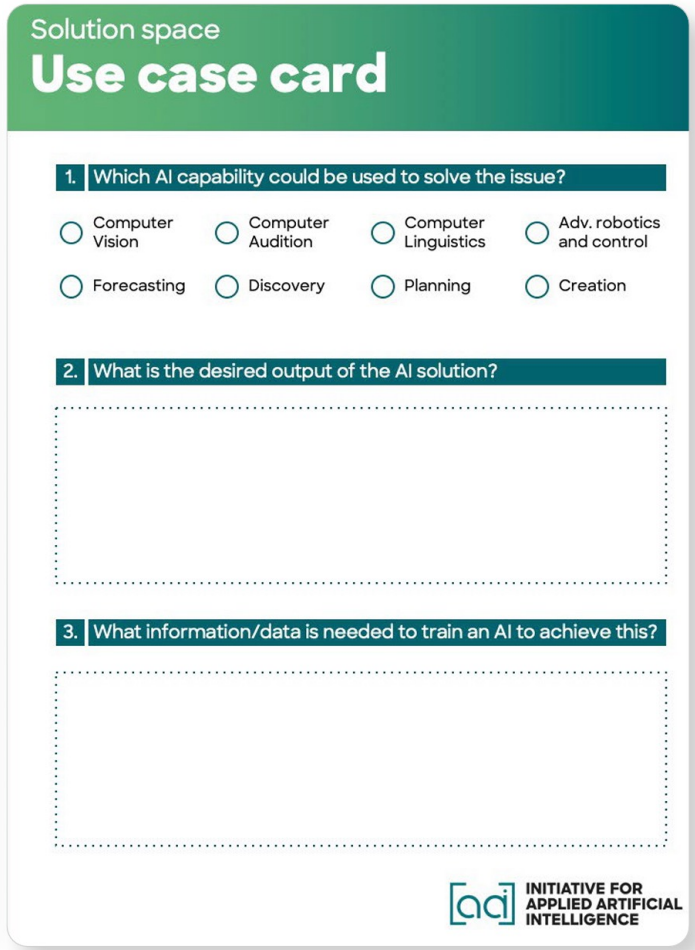
Business cases

1. https://research.aimultiple.com/ai-usecases/

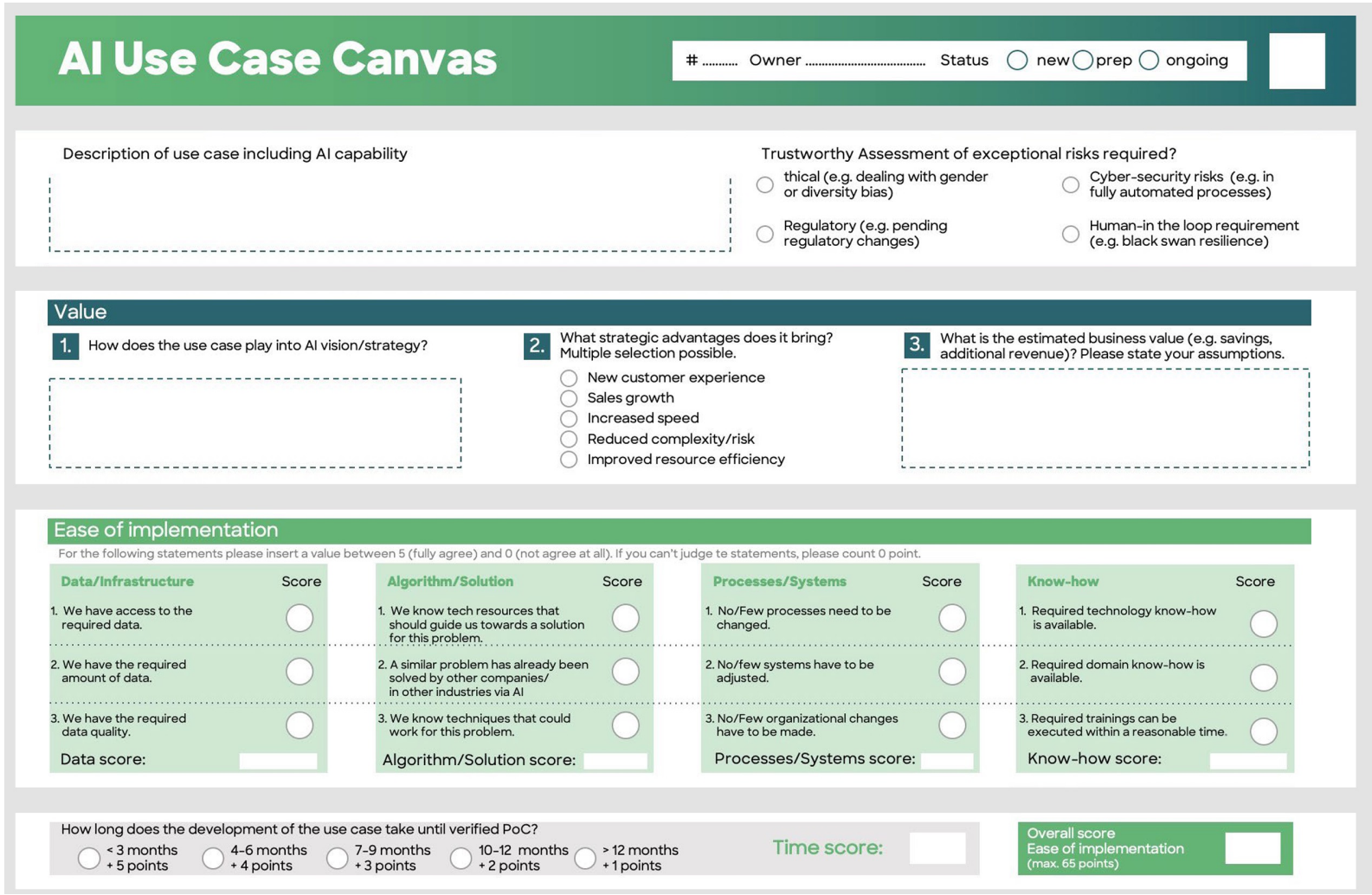


Business cases





Business cases



AI Use Case Canvas

..... Owner ..... Status ◯ new ◯ prep ◯ ongoing

Description of use case including AI capability

Trustworthy Assessment of exceptional risks required?

- ◯ thical (e.g. dealing with gender or diversity bias)
- ◯ Cyber-security risks (e.g. in fully automated processes)
- ◯ Regulatory (e.g. pending regulatory changes)
- ◯ Human-in the loop requirement (e.g. black swan resilience)

Value

1. How does the use case play into AI vision/strategy?

2. What strategic advantages does it bring? Multiple selection possible.
   - ◯ New customer experience
   - ◯ Sales growth
   - ◯ Increased speed
   - ◯ Reduced complexity/risk
   - ◯ Improved resource efficiency

3. What is the estimated business value (e.g. savings, additional revenue)? Please state your assumptions.

Ease of implementation

For the following statements please insert a value between 5 (fully agree) and 0 (not agree at all). If you can't judge te statements, please count 0 point.

Data/Infrastructure	Score	Algorithm/Solution	Score	Processes/Systems	Score	Know-how	Score
1. We have access to the required data.	◯	1. We know tech resources that should guide us towards a solution for this problem.	◯	1. No/Few processes need to be changed.	◯	1. Required technology know-how is available.	◯
2. We have the required amount of data.	◯	2. A similar problem has already been solved by other companies/ in other industries via AI	◯	2. No/few systems have to be adjusted.	◯	2. Required domain know-how is available.	◯
3. We have the required data quality.	◯	3. We know techniques that could work for this problem.	◯	3. No/Few organizational changes have to be made.	◯	3. Required trainings can be executed within a reasonable time.	◯
Data score:		Algorithm/Solution score:		Processes/Systems score:		Know-how score:	

How long does the development of the use case take until verified PoC?

◯ < 3 months + 5 points ◯ 4–6 months + 4 points ◯ 7–9 months + 3 points ◯ 10–12 months + 2 points ◯ > 12 months + 1 points

Time score:

Overall score Ease of implementation (max. 65 points)

Kafka (Tensorflow IO)

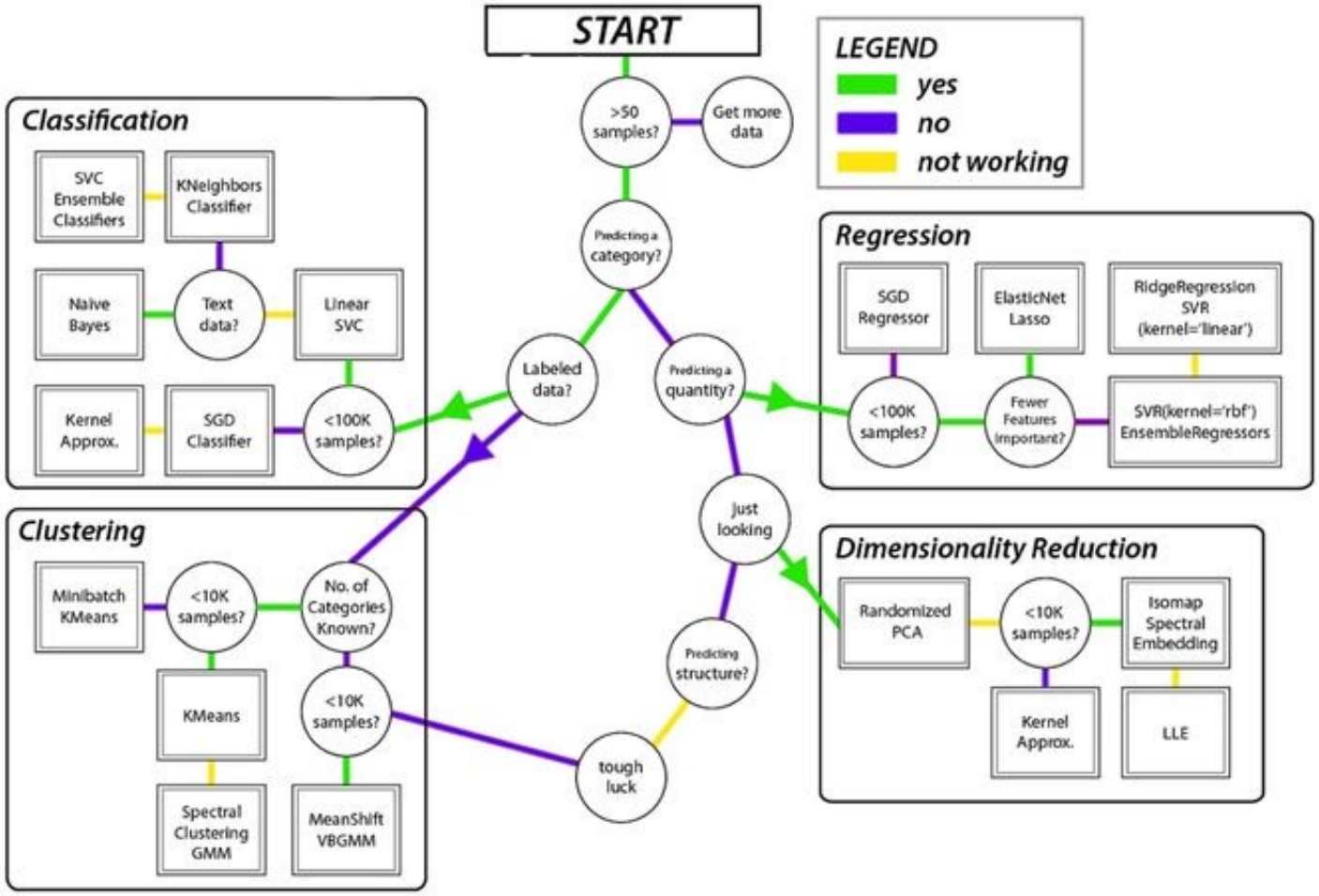
1. https://medium.com/rahasak/kafka-and-zookeeper-with-docker-65cff2c2c34f
2. https://www.tensorflow.org/io/tutorials/kafka

Python & JIRA

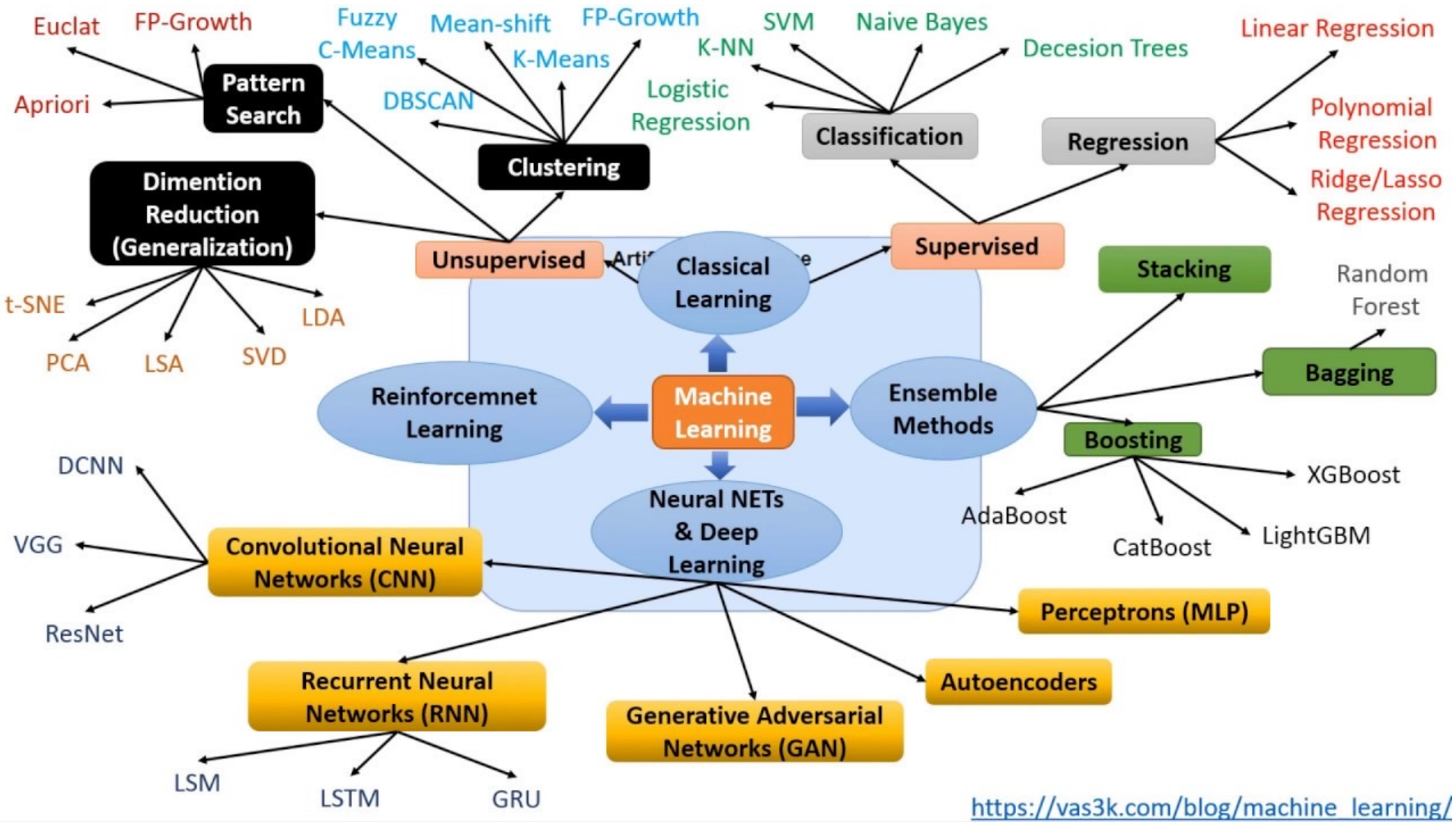
Sentiment:

- https://www.kaggle.com/datasets?search=sentiment+analysis
- https://github.com/bmcclannahan/NLP-Sentiment
- https://github.com/rtflynn/NLP-Sentiment
- https://github.com/bentrevett/pytorch-sentiment-analysis
- https://github.com/topics/sentiment-analysis?o=asc&s=stars

Advanced models



Advanced models



https://vas3k.com/blog/machine_learning/

Service Map

Global

Vision

Strategy

Metrics

CGI solution & services

CGI IP

Engineering

Data enablement
Data understanding
Data visualization
Trainings

Provide data

Science

Train, Evaluate & optimize model
Trainings

Provide model

Collaboration

Provide need

Business

Business understanding
Design thinking: Find use case
AI enablement: Trainings

Operations

Data infrastructure (DataOps)
ML infrastructure (MLOps)
Trainings

Integrate running model

Local

Infrastructure

Consulting

Team

Building
Trainings
Onboarding
Ramp up
Sprints

Data Engineering 101

Tobias Oberrauch

CGI

Tobias Oberrauch: Executive Consultant with focus on AI

Fähigkeiten und Kenntnisse

Top-Fähigkeiten

Beratung | Künstliche Intelligenz

Consulting | Zielstrebigkeit | Visionäres Denken | Machine Learning

Artificial Intelligence | Coaching | Vue.js | Maschinelles Lernen

Big Data | Natural Language Processing (NLP) | Computer Vision

Tensorflow | Konzeption | Microsoft Azure | Herausforderung

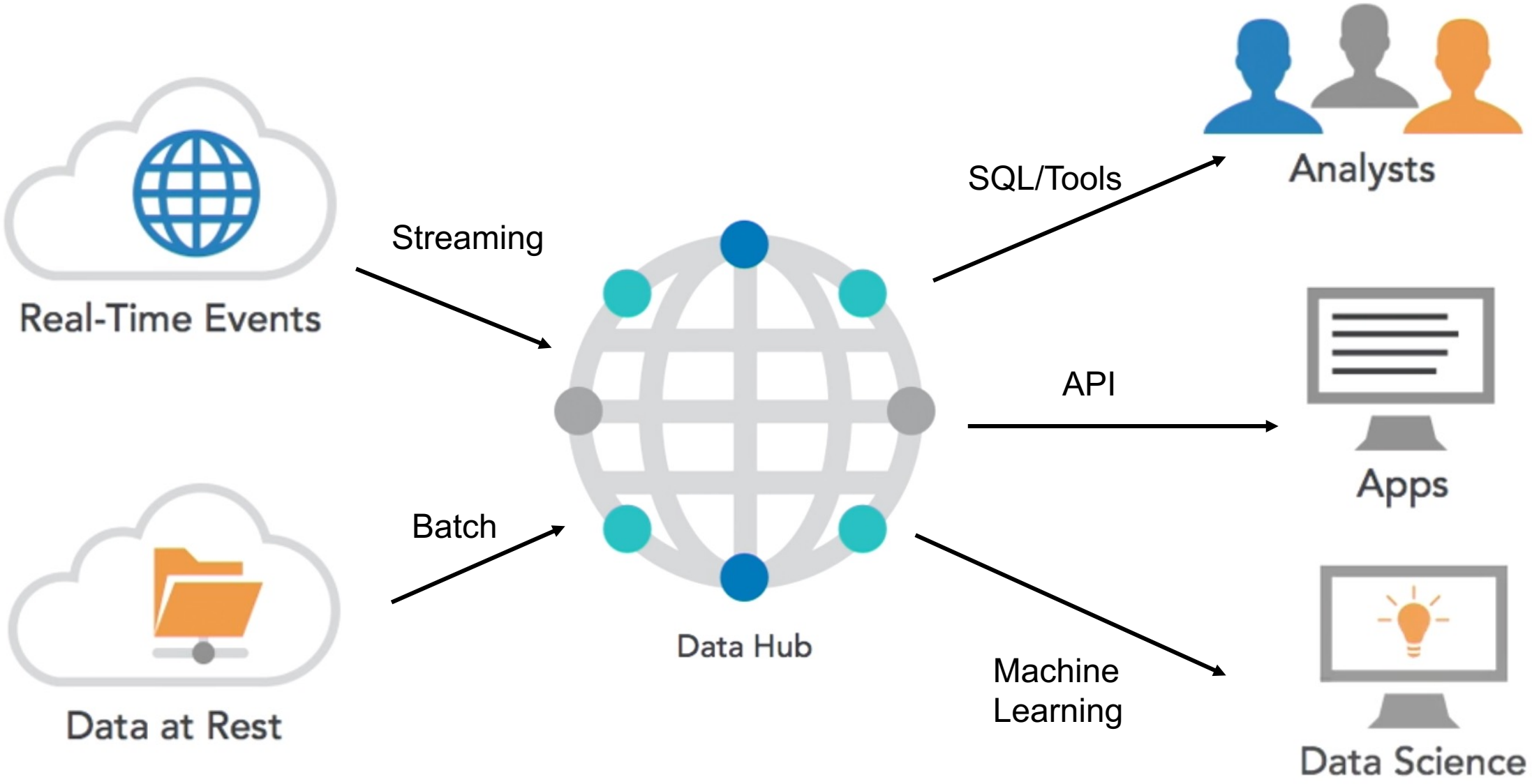
Informationstechnologie | Software | TypeScript | Projektmanagement

Agenda

- What is Data Engineering?
- Difference between Data Engineer and Data Scientist
- Different roles in Data Engineering
- **Core Data Engineering Skills and Resources to Learn**
  - Solid Knowledge of Operating Systems
  - Data Warehousing – Hadoop, MapReduce, HIVE, Apache Spark, Kafka
  - Heavy, In-Depth Database Knowledge –NoSQL and SQL
  - Basic Machine Learning Familiarity
- **Data Engineering on AWS**
  - Role of Data Engineering in cloud
  - Architecture and Data account Flow
  - some services used for data engineering

What is Data Engineering?

Designing, building and scaling system that organize data


Real-Time Events
Streaming
Data at Rest
Batch
Data Hub
SQL/Tools
Analysts
API
Apps
Machine
Learning
Data Science

Data Engineering Responsibilities

Data Ops Tasks

Infrastructure

Availablility / Performance

Data Prep

Staging

Cleansing

Conforming

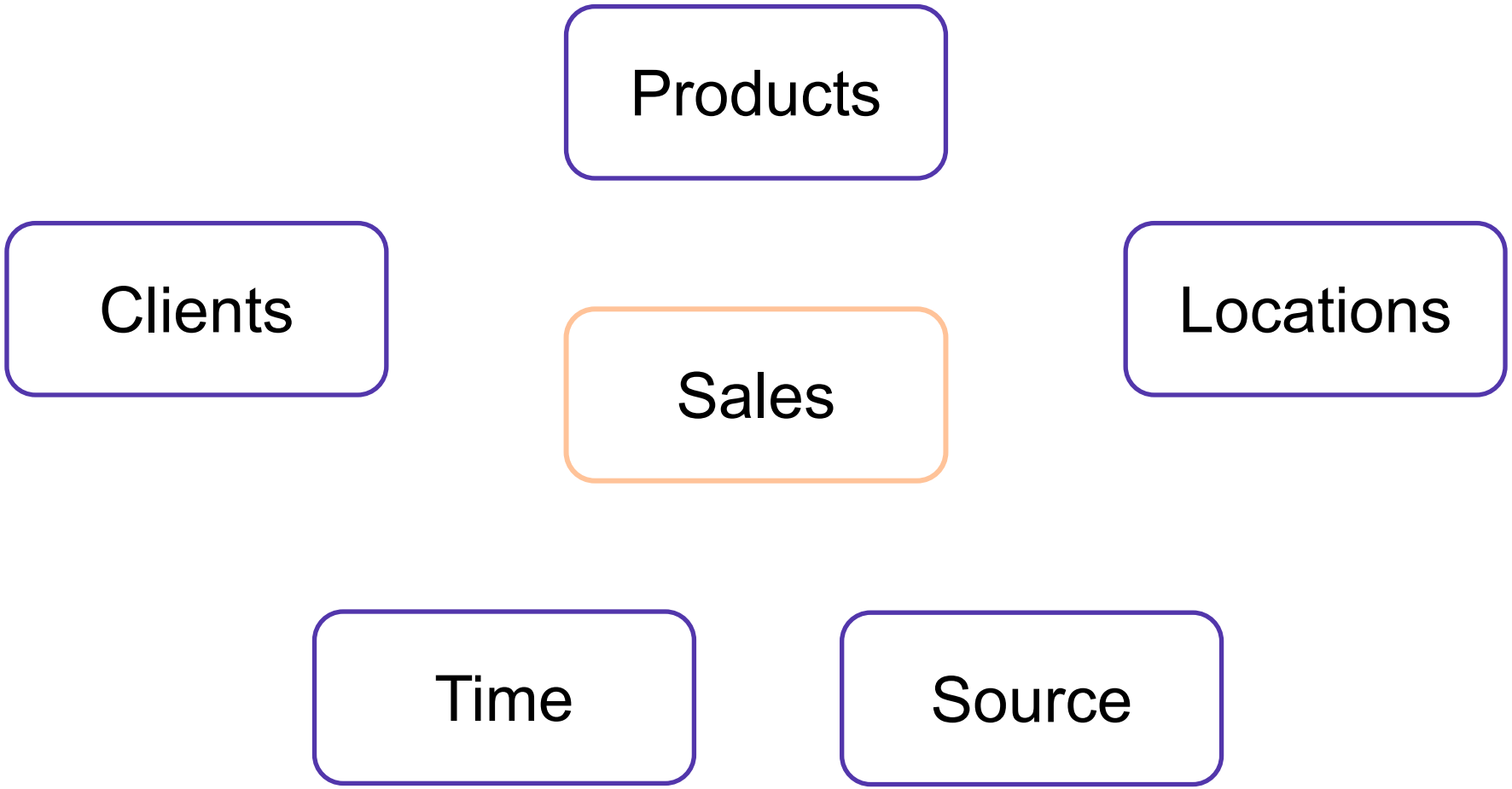
Delivering

Data Interfaces

APIs

Query Tool Compatibility

Star Schema



Star Schema

Facts

Subject of analysis
Numerical
Atomic grain
Multiple types
Additive and semi-additive

Dimensions

Context of analysis
Textual (most often)
Slowly chaning
Contain most attributes
Slice and dice

Difference between Data Engineer and Data Scientist

Data Engineer

They build and scale the platforms that enable data collection, processing and storage.

- Statistical & Analytical skills.
- Data Mining activities
- Machine learning and Deep learning principles
- In-depth programming knowledge (SAS/R/Python coding)

Data Scientist

They use linear algebra and multivariable calculus to create new insight from existing data.

- Data warehousing & ETL (Extract, Transform, Load)
- In-depth knowledge of SQL / database
- Big data storage and processing
- Data pipelines and Data architecture
- Machine learning concept knowledge

The Different Roles in Data Engineering

Data Architect: A data architect lays down the foundation for data management systems to ingest, integrate and maintain all the data sources. This role requires knowledge of tools like SQL, XML, Hive, Spark, etc.

Database Administrator: As the name suggests, a person working in this role requires extensive knowledge of databases. Responsibilities entail ensuring the databases are available to all the required users, is maintained properly and functions without any hiccups when new features are added.

Data Engineer: The master of the lot. A data engineer, as we've already seen, needs to have knowledge of database tools, languages like Python and Java, distributed systems like Hadoop, among other things. It's a combination of tasks into one single role.

Hadoop vs. Spark

	HADOOP	SPARK
Performance	Slow (operations on disk)	100 times faster (in memory)
Cost	Less expensive (disk)	High (Memory-based)
Fault tolerance	High (Replicate across nodes)	RDD (Resilient distributed datasets)
Data processing	Processes in batches	Processes in batch, real-time and graph
Ease of use	MapReduce has no interactive mode	User-friendly APIs different languages
Language Support	In Java / can write in Python, R, C++	In Scala / can write in **Python**, R, Java
Scalability	Highly (Yahoo used 42.000 nodes)	Medium (used in 8.000 nodes)
Security	Kerberos, LDAP and ACLS	password, HDFS ACL, Yarn on Kerberos
Machine Learning	Mahout for data and building models	Built-in machine learning
Schedule	External scheduler (ZooKeeper)	Own scheduler

Spark use cases

- **Customer segmentation.** Analyzing customer behavior and identifying segments of customers that demonstrate similar behavior patterns will help businesses to understand customer preferences and create a unique customer experience.

- **Risk management**. Forecasting different possible scenarios can help managers to make right decisions by choosing non-risky options.

- **Real-time fraud detection.** After the system is trained on historical data with the help of machine-learning algorithms, it can use these findings to identify or predict an anomaly in real time that may signal of a possible fraud.

- **Industrial big data analysis.** It's also about detecting and predicting anomalies, but in this case, these anomalies are related to machinery breakdowns. A properly configured system collects the data from sensors to detect pre-failure conditions.

What is Spark?

Spark is a newer project, initially developed in 2012, at the AMPLab at UC Berkeley. It's also a top-level Apache project focused on processing data in parallel across a cluster, but the biggest difference is that it works in-memory.

Whereas Hadoop reads and writes files to HDFS, Spark processes data in RAM using a concept known as an RDD, Resilient Distributed Dataset. Spark can run either in stand-alone mode, with a Hadoop cluster serving as the data source, or in conjunction with Mesos. In the latter scenario, the Mesos master replaces the Spark master or YARN for scheduling purposes.

Spark is structured around Spark Core, the engine that drives the scheduling, optimizations, and RDD abstraction, as well as connects Spark to the correct filesystem (HDFS, S3, RDBMs, or Elasticsearch). There are several libraries that operate on top of Spark Core, including Spark SQL, which allows you to run SQL-like commands on distributed data sets, MLLib for machine learning, GraphX for graph problems, and streaming which allows for the input of continually streaming log data.

Spark has several APIs. The original interface was written in Scala, and based on heavy usage by data scientists, Python and R endpoints were also added. Java is another option for writing Spark jobs.

Use Spark

- **Fast data processing.** In-memory processing makes Spark faster than Hadoop MapReduce – up to 100 times for data in RAM and up to 10 times for data in storage.
- **Iterative processing.** If the task is to process data again and again – Spark defeats Hadoop MapReduce. Spark's Resilient Distributed Datasets (RDDs) enable multiple map operations in memory, while Hadoop MapReduce has to write interim results to a disk.
- **Near real-time processing.** If a business needs immediate insights, then they should opt for Spark and its in-memory processing.
- **Graph processing.** Spark's computational model is good for iterative computations that are typical in graph processing. And Apache Spark has GraphX – an API for graph computation.
- **Machine learning**. Spark has MLlib – a built-in machine learning library, while Hadoop needs a third-party to provide it. MLlib has out-of-the-box algorithms that also run in memory. But if required, our Spark specialists will tune and adjust them to tailor to your needs.
- **Joining datasets.** Due to its speed, Spark can create all combinations faster, though Hadoop may be better if joining of very large data sets that requires a lot of shuffling and sorting is needed.

The Spark ecosystem

1. **Spark Core:** Underlying execution engine that schedules and dispatches tasks and coordinates input and output (I/O) operations.

2. **Spark SQL:** Gathers information about structured data to enable users to optimize structured data processing.

3. **Spark Streaming and Structured Streaming:** Both add stream processing capabilities. Spark Streaming takes data from different streaming sources and divides it into micro-batches for a continuous stream. Structured Streaming, built on Spark SQL, reduces latency and simplifies programming.

4. **Machine Learning Library (MLlib):** A set of machine learning algorithms for scalability plus tools for feature selection and building ML pipelines. The primary API for MLlib is DataFrames, which provides uniformity across different programming languages like Java, Scala and Python.

What is Hadoop?

Hadoop got its start as a Yahoo project in 2006, becoming a top-level Apache open-source project later on. It's a general-purpose form of distributed processing that has several components: the Hadoop Distributed File System (HDFS), which stores files in a Hadoop-native format and parallelizes them across a cluster; YARN, a schedule that coordinates application runtimes; and MapReduce, the algorithm that actually processes the data in parallel. Hadoop is built in Java, and accessible through many programming languages, for writing MapReduce code, including Python, through a Thrift client.

In addition to these basic components, Hadoop also includes Sqoop, which moves relational data into HDFS; Hive, a SQL-like interface allowing users to run queries on HDFS; and Mahout, for machine learning. In addition to using HDFS for file storage, Hadoop can also now be configured to use S3 buckets or Azure blobs as input.

It's available either open-source through the Apache distribution, or through vendors such as Cloudera (the largest Hadoop vendor by size and scope), MapR, or HortonWorks.

Hadoop use cases

- **Linear processing of huge data sets.** Hadoop MapReduce allows parallel processing of huge amounts of data. It breaks a large chunk into smaller ones to be processed separately on different data nodes and automatically gathers the results across the multiple nodes to return a single result. In case the resulting dataset is larger than available RAM, Hadoop MapReduce may outperform Spark.

- **Economical solution, if no immediate results are expected.** Our Hadoop team considers MapReduce a good solution if the speed of processing is not critical. For instance, if data processing can be done during night hours, it makes sense to consider using Hadoop MapReduce.

The Hadoop ecosystem

1. **Hadoop Distributed File System (HDFS):** Primary data storage system that manages large data sets running on commodity hardware. It also provides high-throughput data access and high fault tolerance.

2. **Yet Another Resource Negotiator (YARN):** Cluster resource manager that schedules tasks and allocates resources (e.g., CPU and memory) to applications.

3. **Hadoop MapReduce:** Splits big data processing tasks into smaller ones, distributes the small tasks across different nodes, then runs each task.

4. **Hadoop Common (Hadoop Core):** Set of common libraries and utilities that the other three modules depend on.

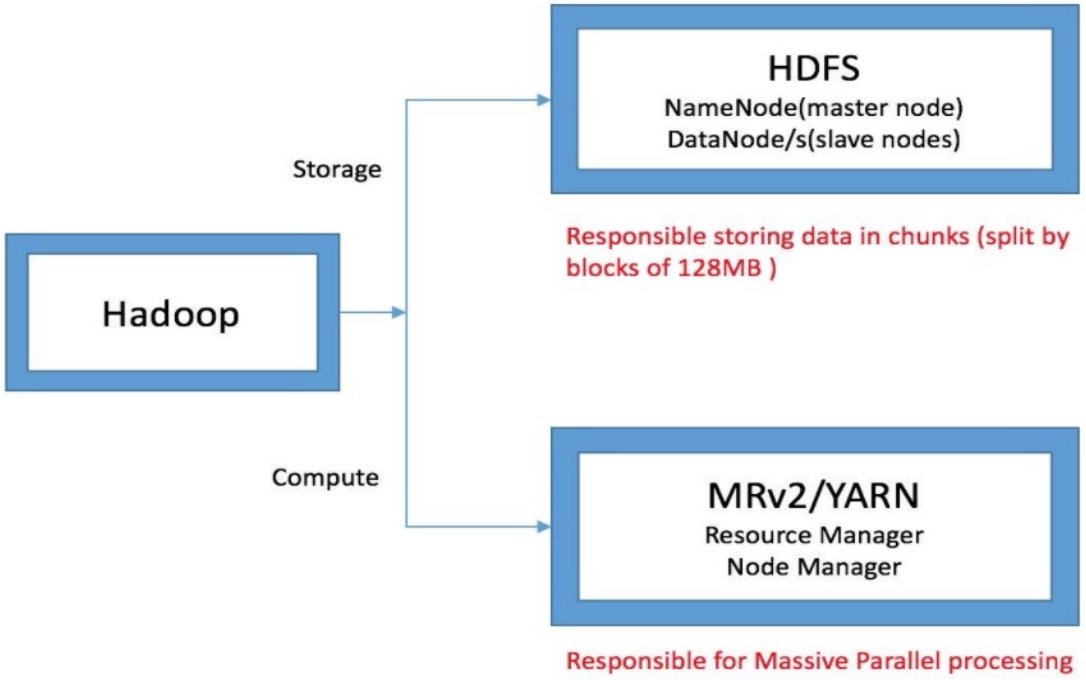
Hadoop-based analytics

Below are the main components used in Bigdata

- Hadoop
  - HDFS
  - MapReduce
  - Yarn
- Apache SPARK
- Apache Hive
- Apache HBASE
- Apache KAFKA
- Apache Sqoop
- Apache Oozie
- Apache Flume

Hadoop-based analytics: What is Hadoop?

Hadoop is a framework that allows us to store and process large datasets in parallel and distributed fashion.



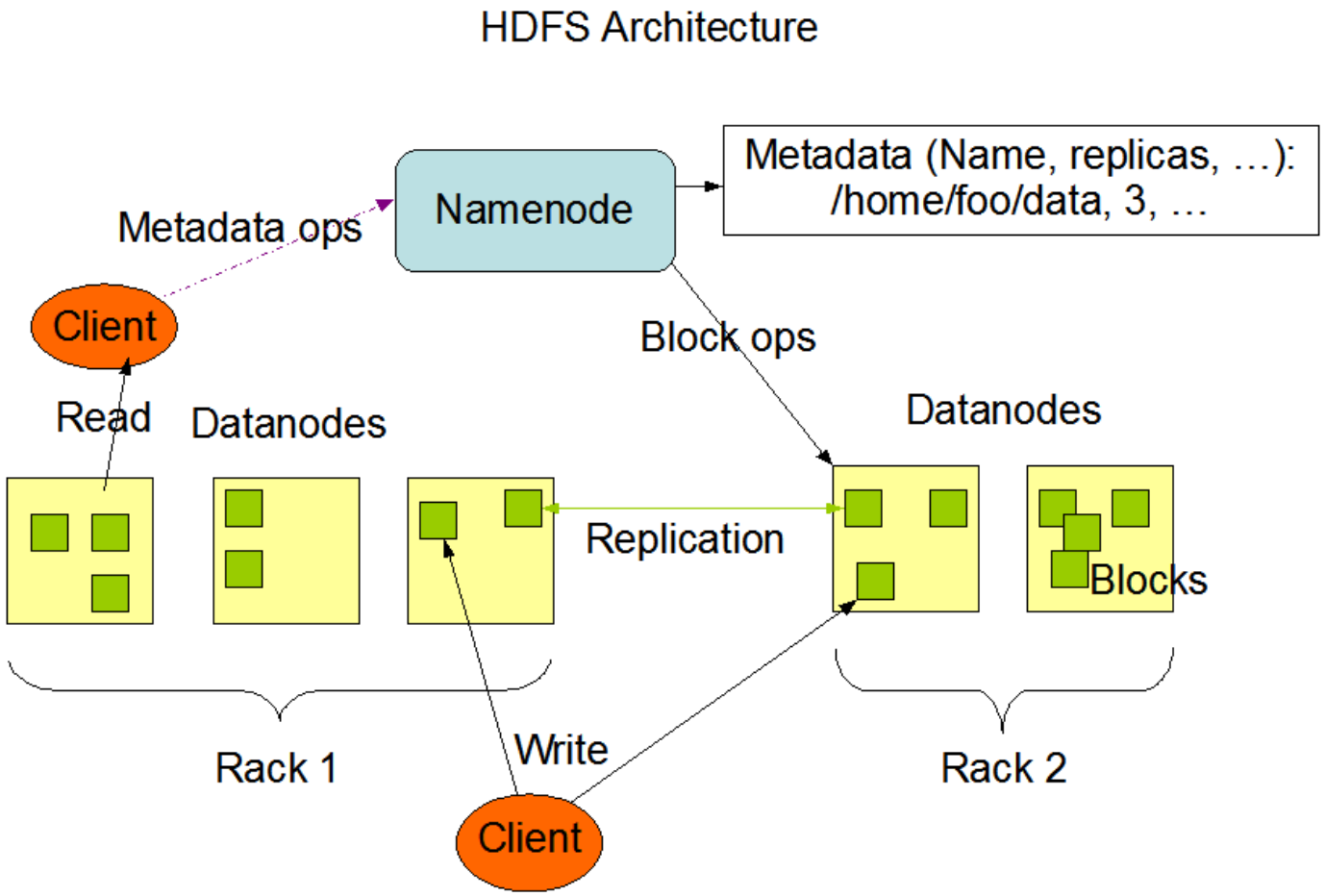
Hadoop-based analytics: HDFS(Hadoop Distibuted File System)

Allows you to dump all kind of data accross the cluster.

- Storage unit of Hadoop
- Distibuted File sytsem
- Divide files into chunks and store it across the cluster
- Stores any kind of data (structured,semi structured, unstructured)
- No schema validation is done while dumping the data into HDFS
- As per the requirement it will be scaling vertically.

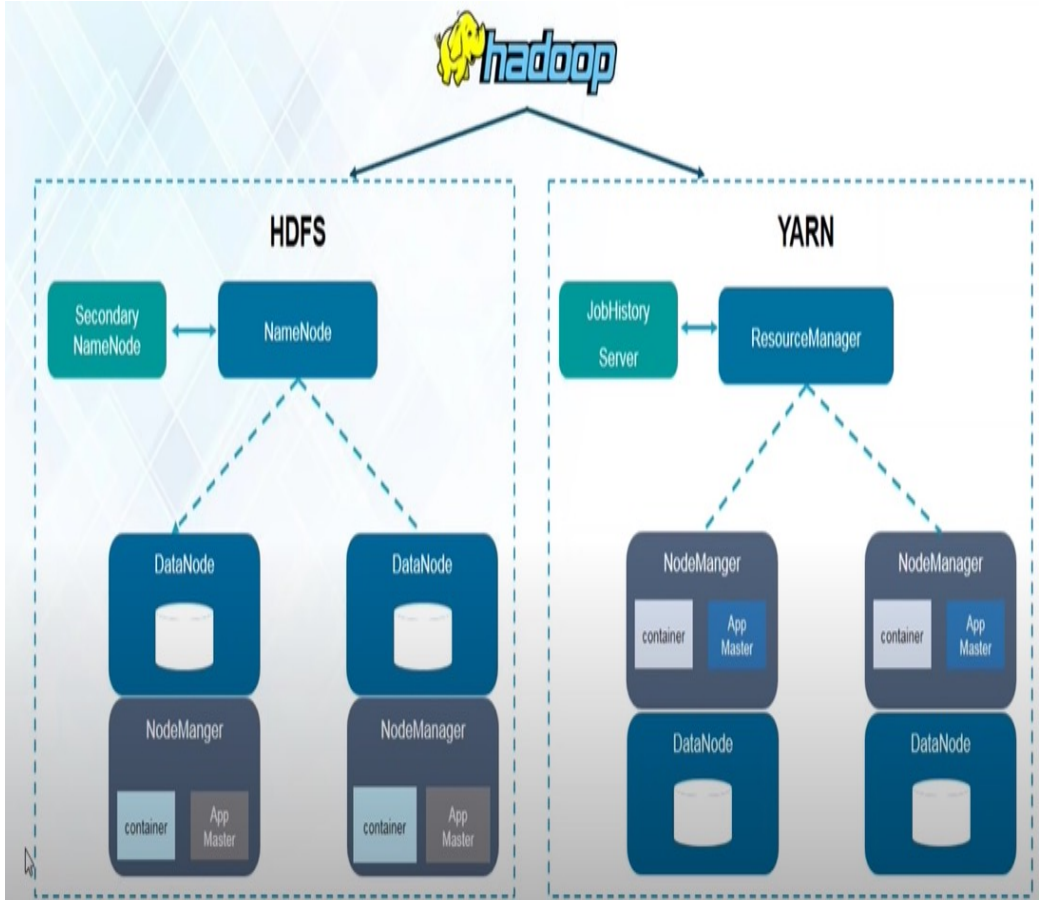
HDFS has a master/slave architecture. An HDFS cluster consists of a single NameNode, In addition, there are a number of DataNodes, usually one per node in the cluster,

Hadoop-based analytics: HDFS Architecture



Hadoop-based analytics: YARN(Yet Another Resource Negotiator)

- YARN is the job scheduling, and resource management innovation.
- YARN will take care that the different applications running in a Hadoop cluster and scheduling tasks to be executed on various cluster nodes.
- Yarn is intended to share the responsibilities of Map Reduce and deal with the cluster administration task.
- **Yarn** does efficient utilization of the resource. There are no more fixed map-reduce slots. YARN provides central resource manager. With YARN, you can now run multiple applications in Hadoop, all sharing a common resource.

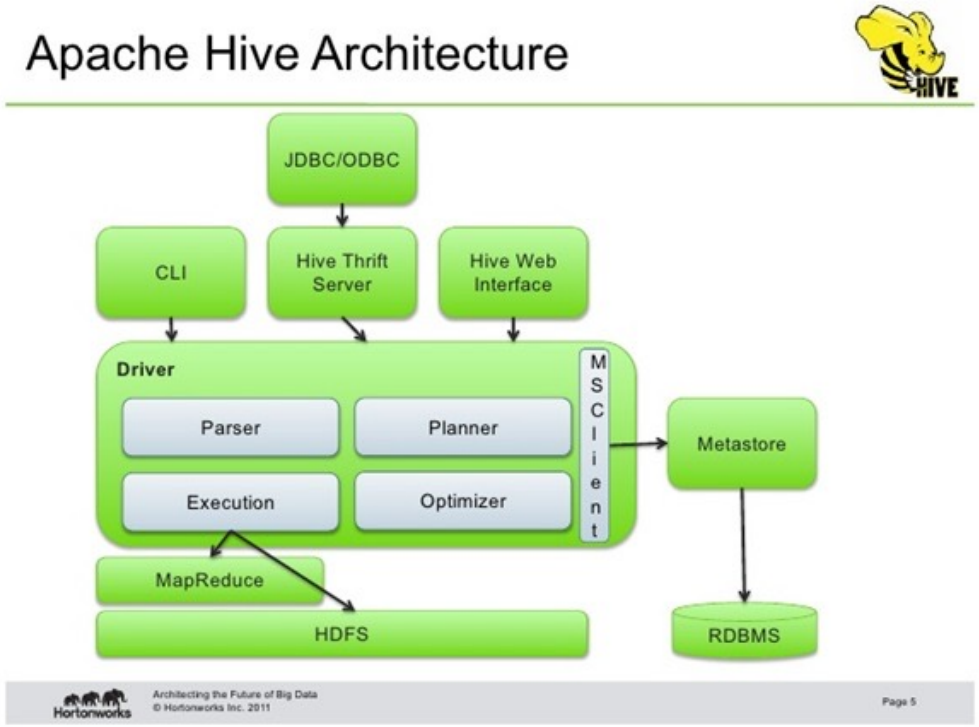


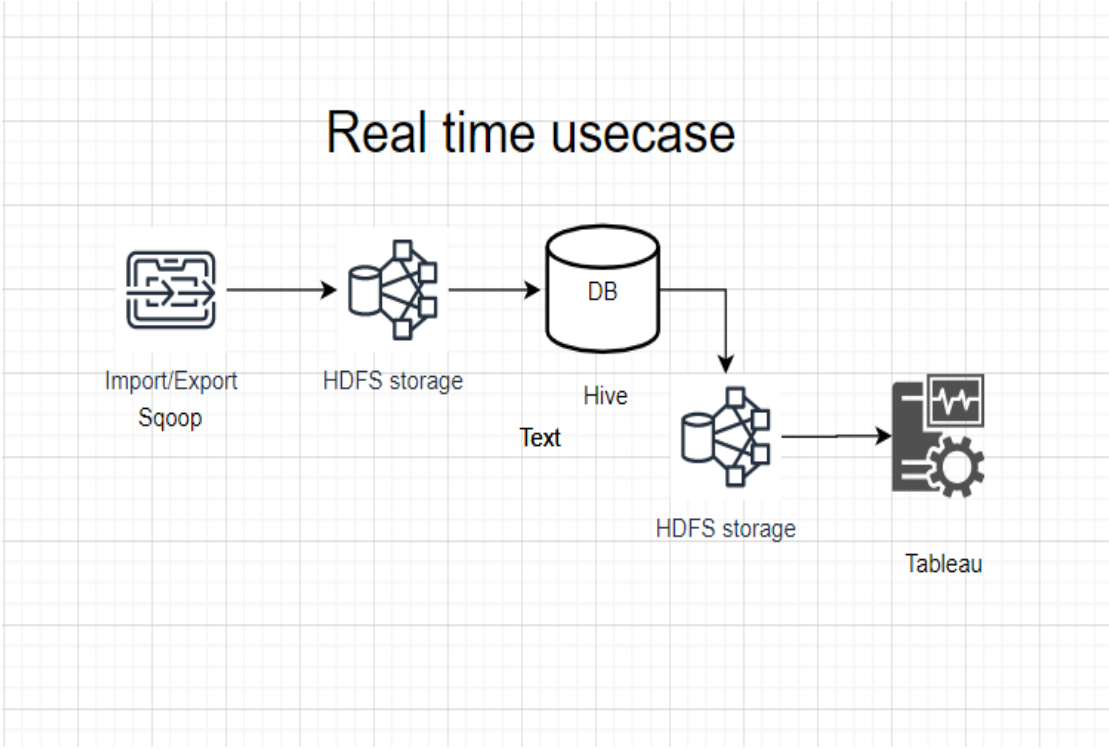
Apache Hive (SQL like Interface)

- It is a DWH software project built on top of Apache Hadoop for providing data query and analysis.
- Hive gives an SQL-like interface to query data stored in various db and file systems that integrate with Hadoop.
- **Hive** is scalable, fast, and uses familiar concepts. Schema gets stored in a database, while processed data goes into a Hadoop Distributed File System (HDFS).
- Hive is best suited for batch jobs.
- It cannot work for online transaction processing (OLTP) systems since it does not provide real-time querying for row-level updates.

Criteria	Hive
Query language	HiveQL (SQL - like)
Used for	Creating reports
Area of deployment	Server side
Support Data Type	Structured
Integration	JDBC and BI tools

Architecture of Apache Hive





Hbase (No Sql type database)

- HBase is a distributed column-oriented database built on top of the Hadoop file system. It is an open-source project and is horizontally scalable.
- It is a NoSQL database written in Java which performs faster querying.
- It is well suited for Sparse datasets.
- It is built on top most of the Hadoop file system.
- HBase is open source implementation devised on Google's Bigtable.

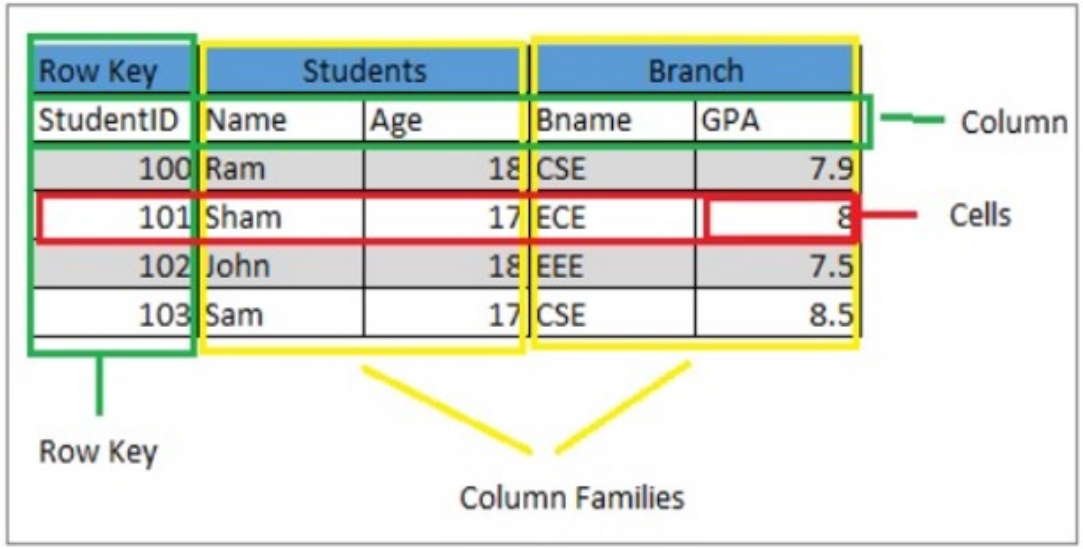
Where to use?

- Whenever there is a need to write heavy applications.
- Need to provide fast random access to available data.
- Schema less and de-normalized data.

Schema of HBase table:

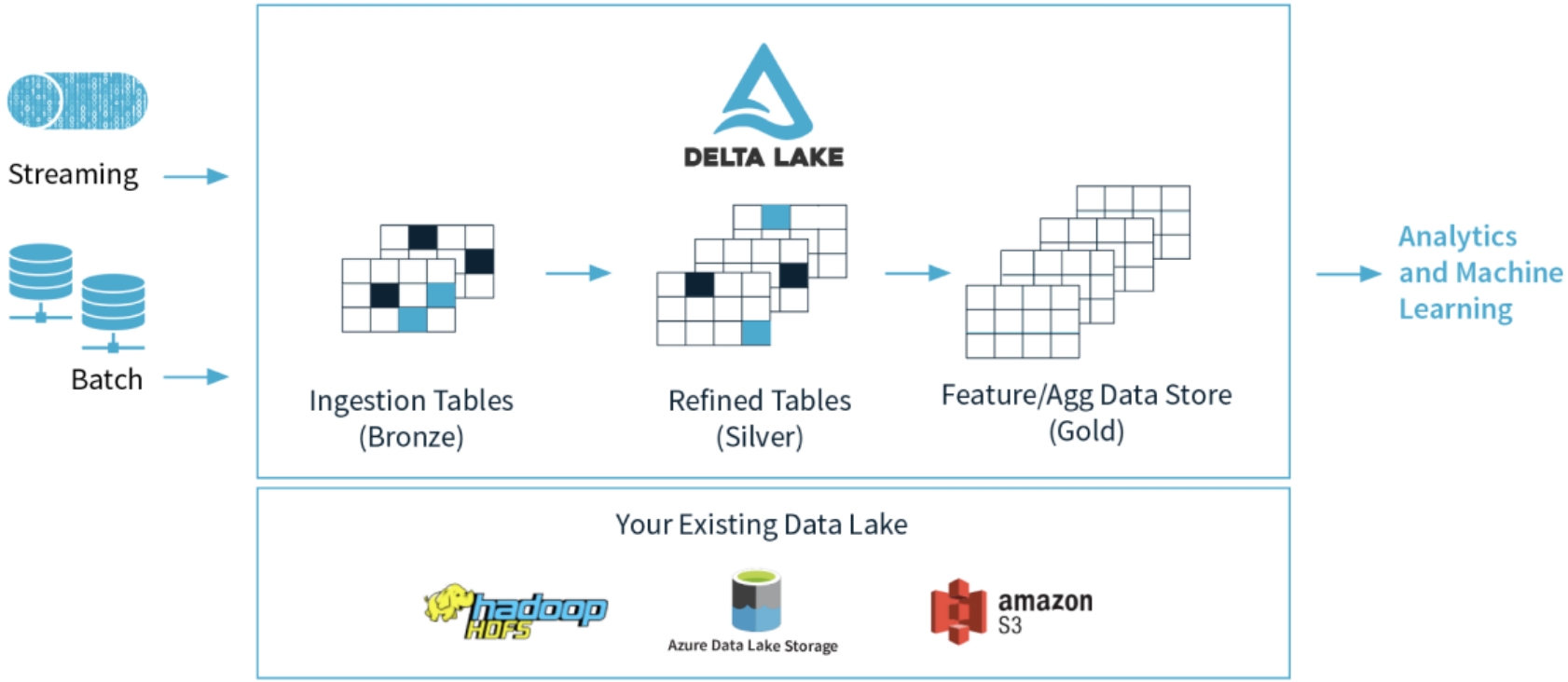
Column Oriented database, Schema less. Defines only column families. Add columns on the fly.

Row id	Column Family			Column Family			Column Family			Column Family		
	col 1	col2	col 3	col1	col2	col3	col1	col2	col3	col1	col2	col3
1												
2												
3												

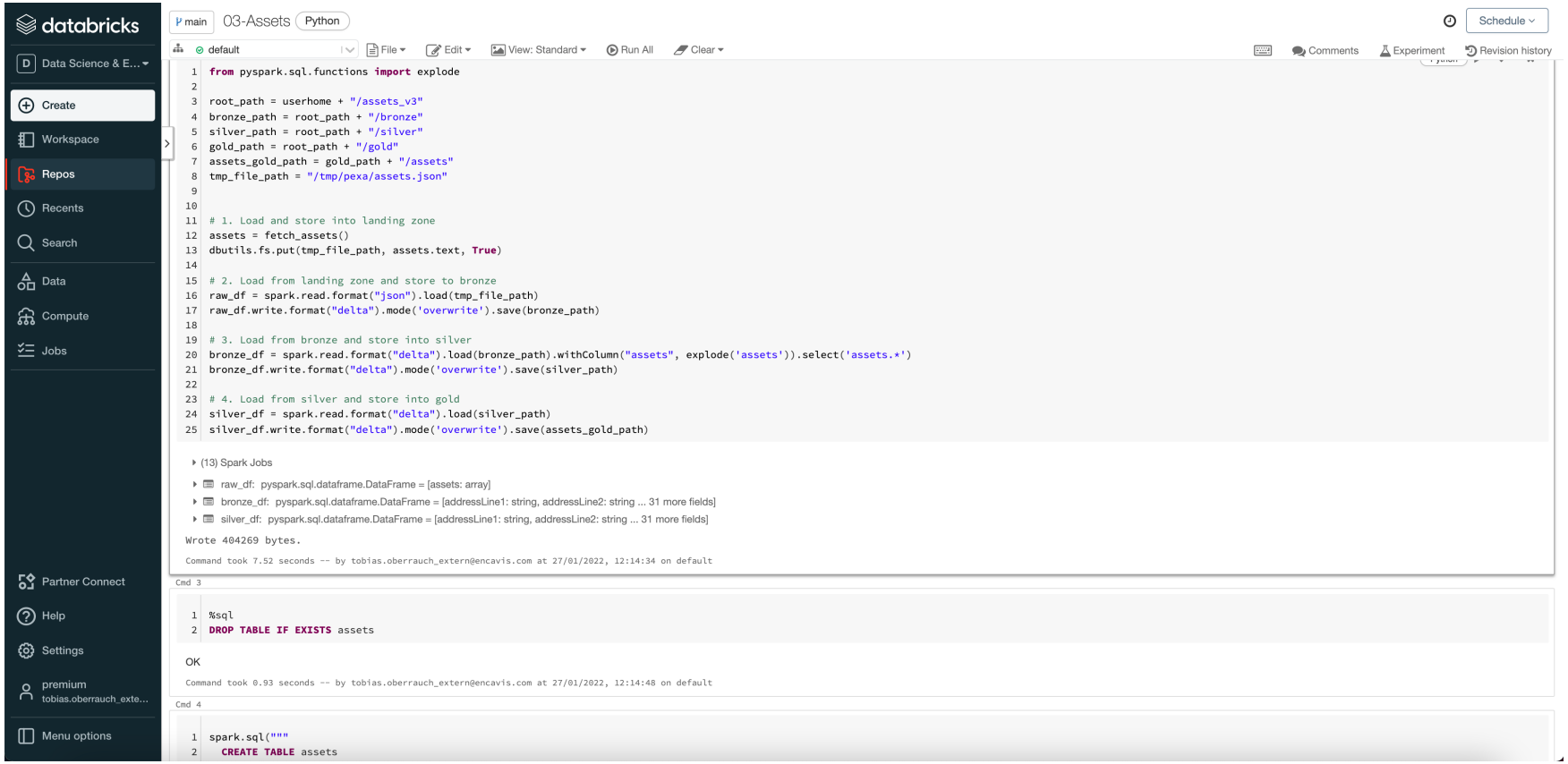


Data Engineering on Cloud

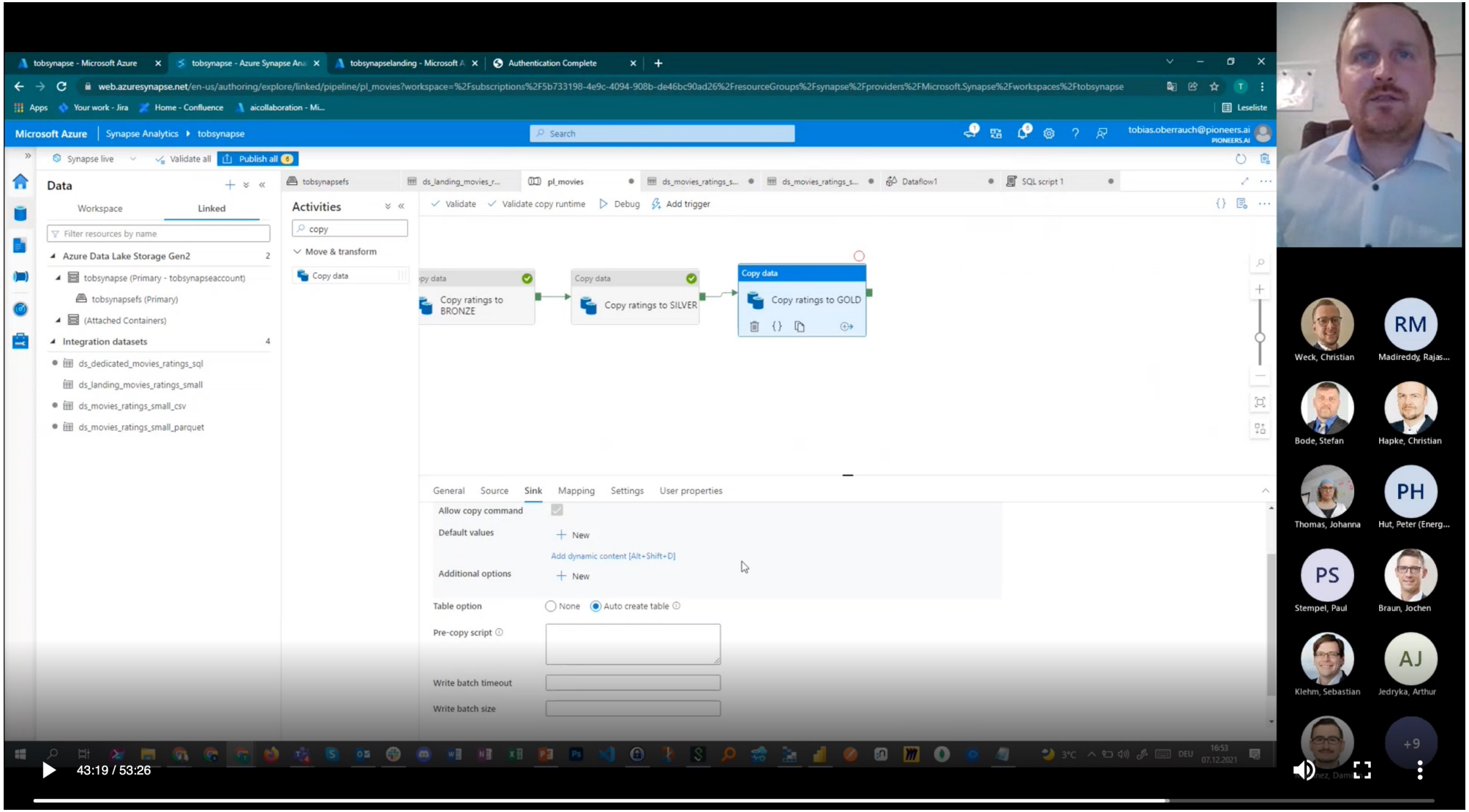
Delta Lake



Databricks



Data Engineering on Azure


tobsynapse - Microsoft Azure
tobsynapse - Azure Synapse Ana
tobsynapselanding - Microsoft A
Authentication Complete
web.azuresynapse.net/en-us/authoring/explore/linked/pipeline/pl_movies?workspace=%2Fsubscriptions%2F5b733198-4e9c-4094-908b-de46bc90ad26%2FresourceGroups%2Fsynapse%2Fproviders%2FMicrosoft.Synapse%2Fworkspaces%2Ftobsynapse
Apps
Your work - Jira
Home - Confluence
aicollaboration - Mi...
Leseliste
Microsoft Azure
Synapse Analytics
tobsynapse
Search
tobias.oberrauch@pioneers.ai
PIONEERS.AI
Synapse live
Validate all
Publish all 6
Data
Workspace
Linked
Filter resources by name
Azure Data Lake Storage Gen2
tobsynapse (Primary - tobsynapseaccount)
tobsynapsefs (Primary)
(Attached Containers)
Integration datasets
ds_dedicated_movies_ratings_sql
ds_landing_movies_ratings_small
ds_movies_ratings_small_csv
ds_movies_ratings_small_parquet
tobsynapsefs
ds_landing_movies_r...
pl_movies
ds_movies_ratings_s...
ds_movies_ratings_s...
Dataflow1
SQL script 1
Activities
copy
Move & transform
Copy data
Validate
Validate copy runtime
Debug
Add trigger
Copy ratings to BRONZE
Copy ratings to SILVER
Copy data
Copy ratings to GOLD
General
Source
Sink
Mapping
Settings
User properties
Allow copy command
Default values
New
Add dynamic content [Alt+Shift+D]
Additional options
New
Table option
None
Auto create table
Pre-copy script
Write batch timeout
Write batch size
43:19 / 53:26
Weck, Christian
Madireddy, Rajas...
Bode, Stefan
Hapke, Christian
Thomas, Johanna
Hut, Peter (Energ...
Stempel, Paul
Braun, Jochen
Klehm, Sebastian
Jedryka, Arthur
+9

Data Engineering on AWS

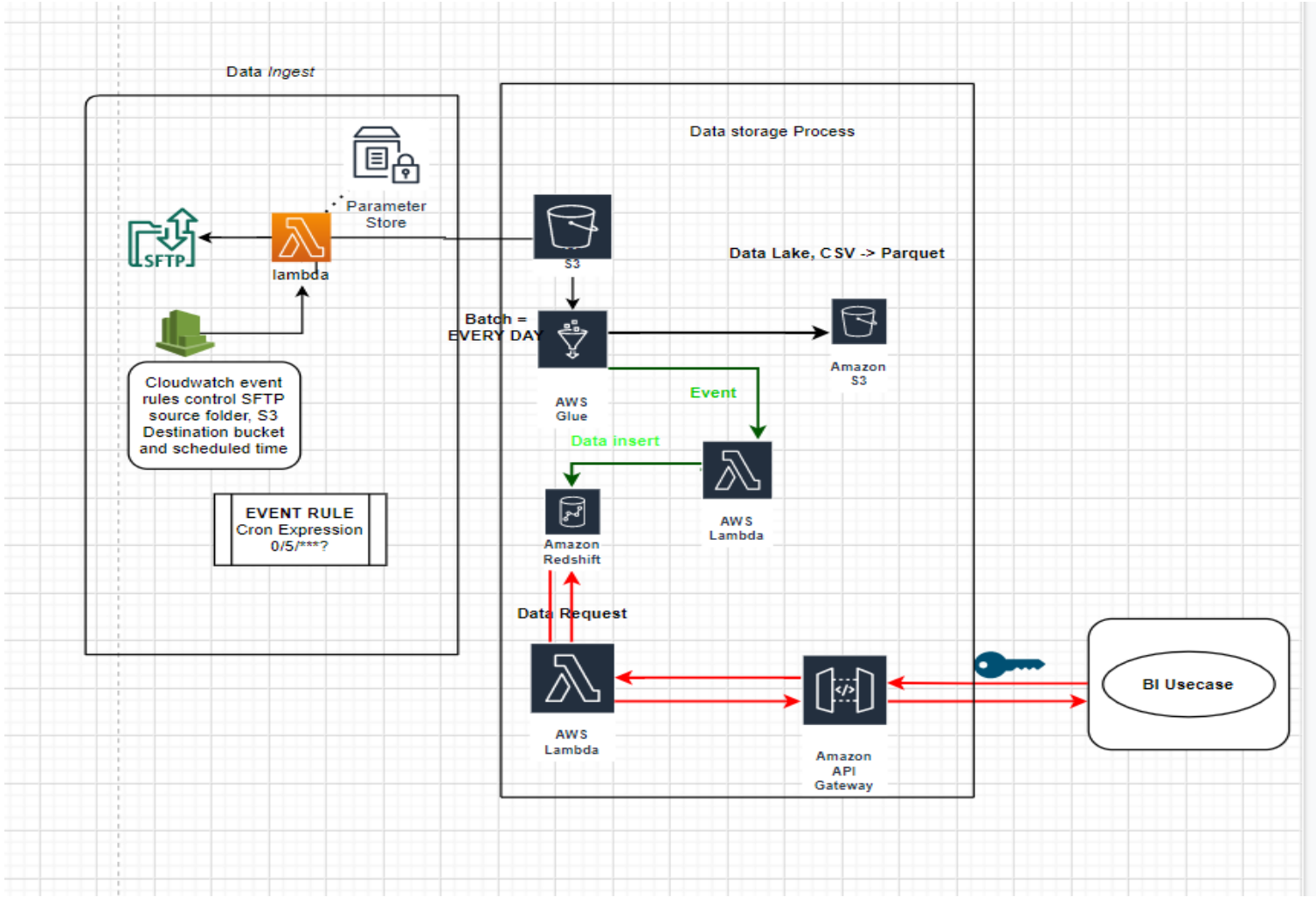
Data Engineering in AWS

Design, build and operationalize large scale enterprise data solutions and applications using AWS and ist services.

Use cases:

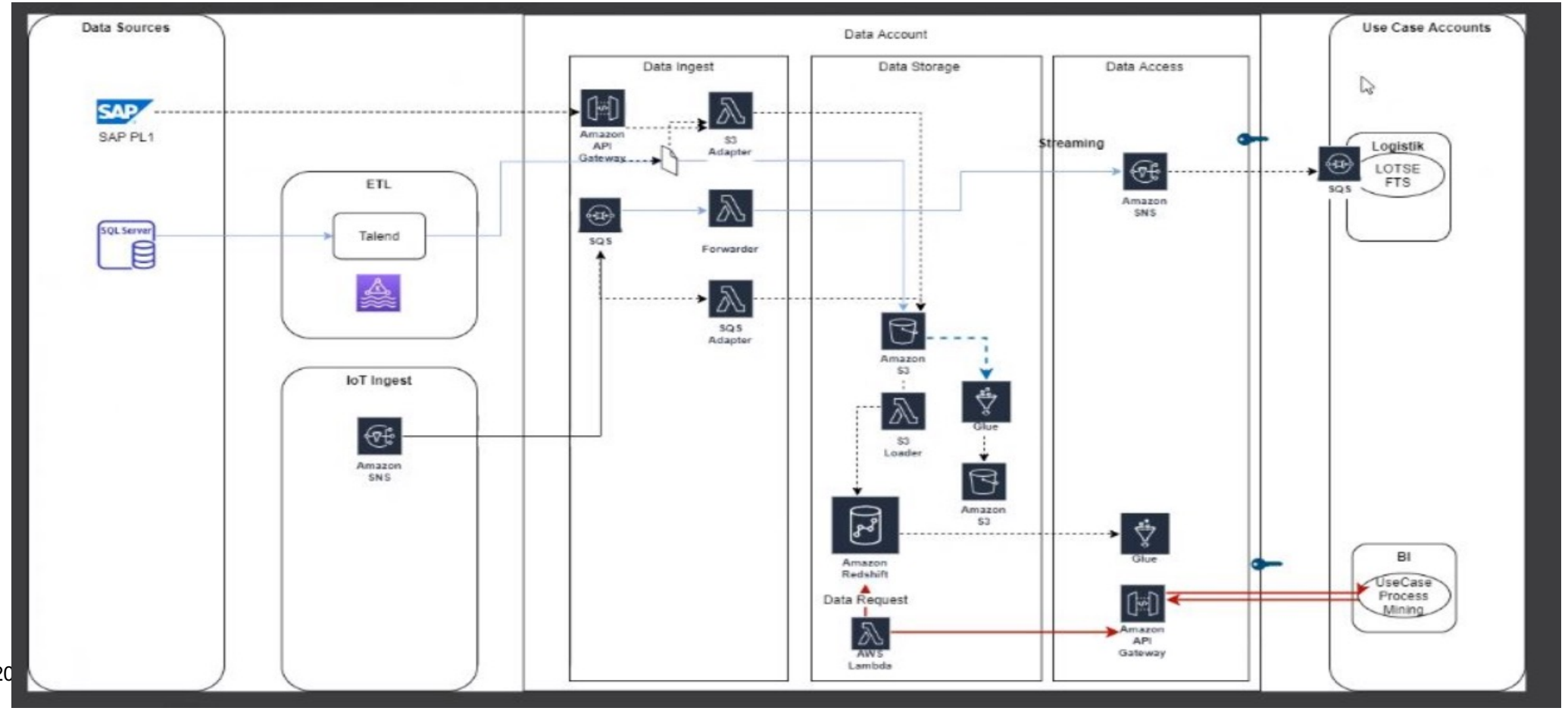
- Data warehousing (AWS Redshift)
- Bigdata Processing (AWS Glue, EMR)
- Real time analytics (SNS,SQS or MSK)
- Operational analytics (Elastic Search)

Architecture- Data Flow



Data Ingestion:

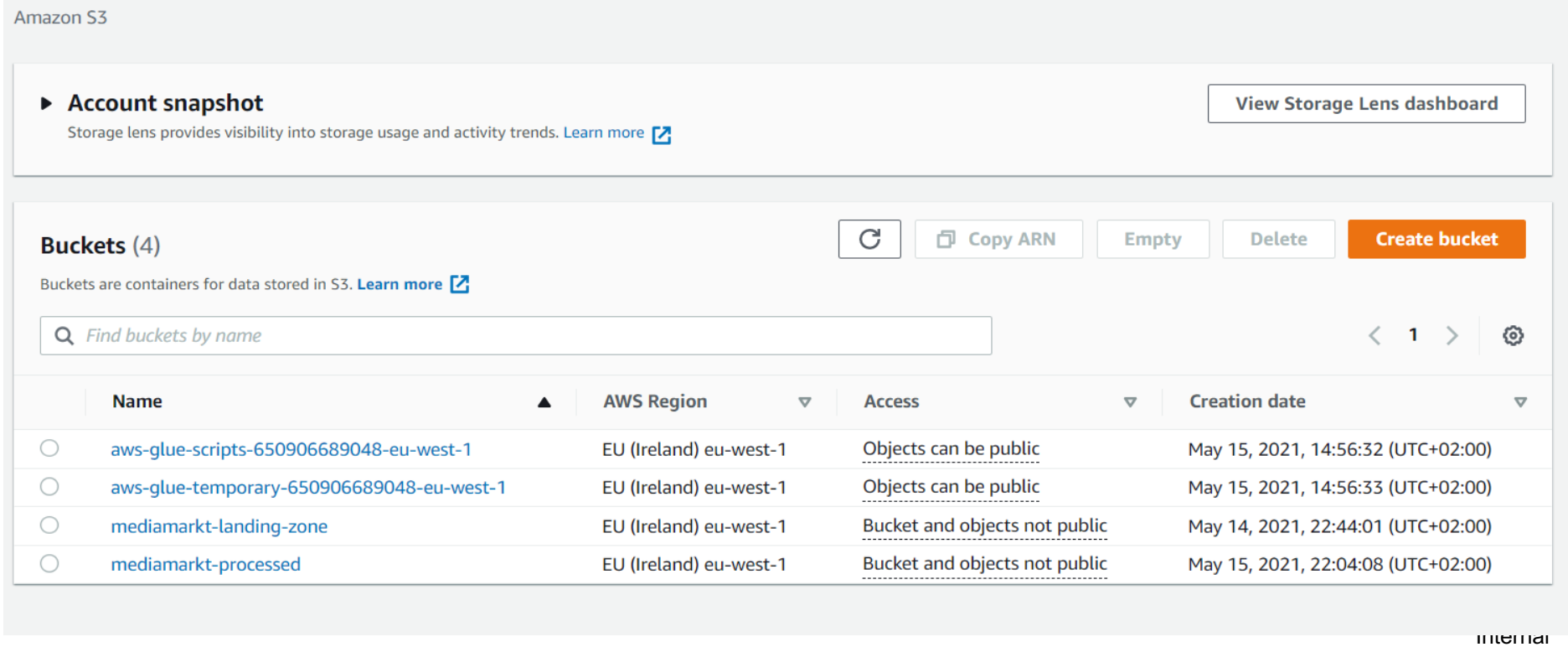
- Transportation of **data** from assorted sources to a storage medium where it can be accessed, used, and analyzed by an organization.



Data Storage and Processing

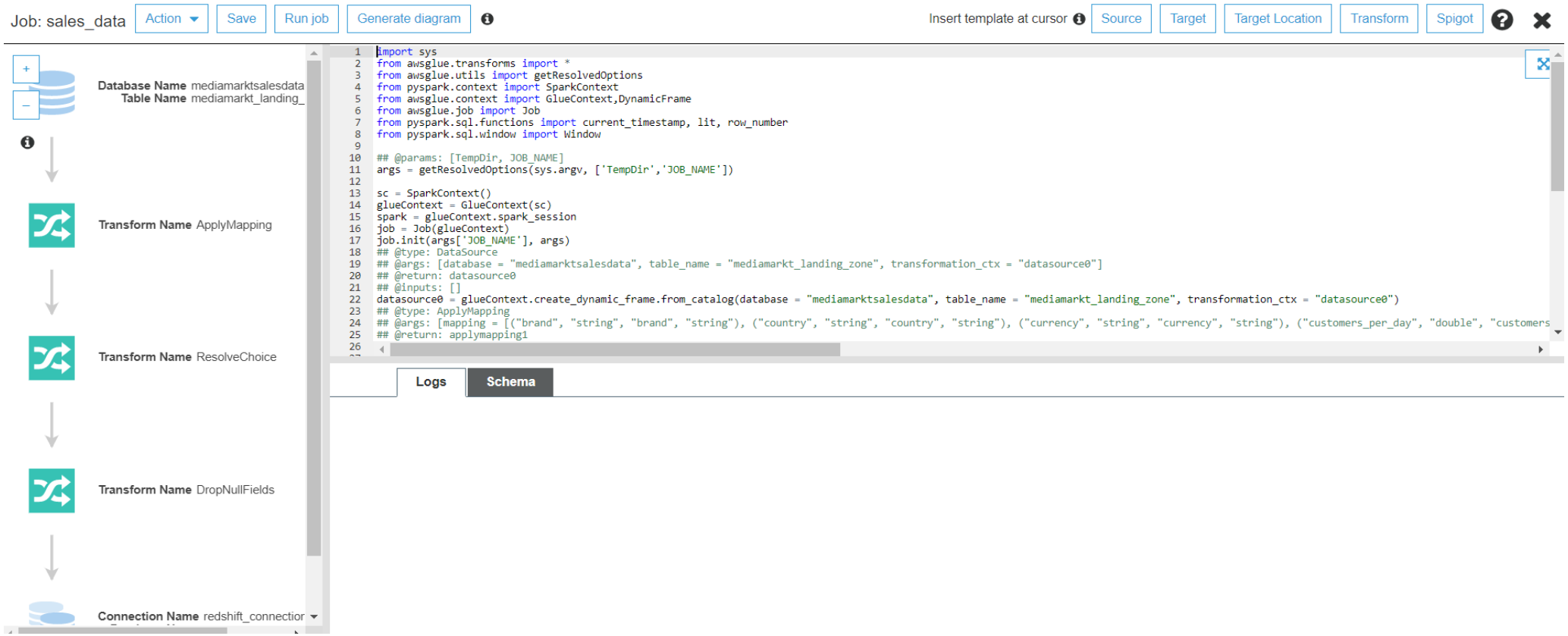
S3(Simple Storage Service):

- After the data ingestion data store in S3
- Central data storage
- Can store any kind of data(batch or stream)



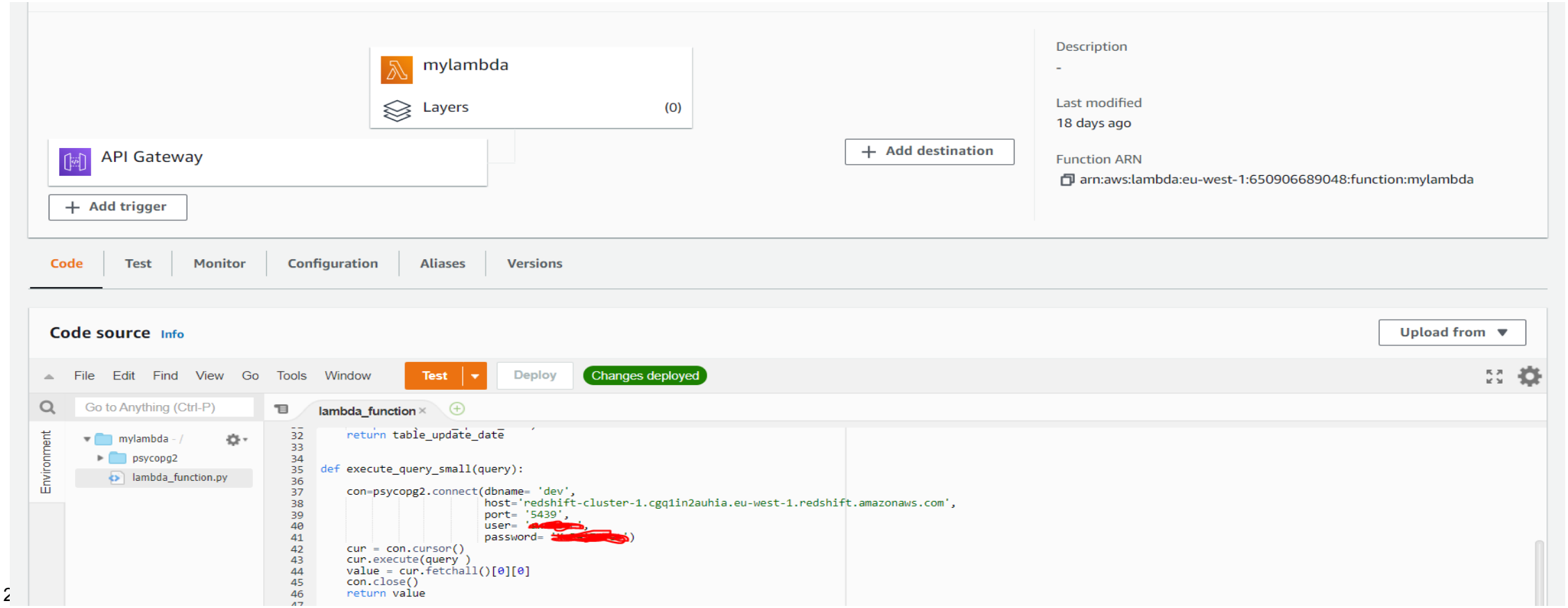
AWS Glue:

The **AWS Glue** is an ETL service that utilizes a fully managed Apache Spark environment. Glue ETL that can clean, enrich our data and load it to database engines.



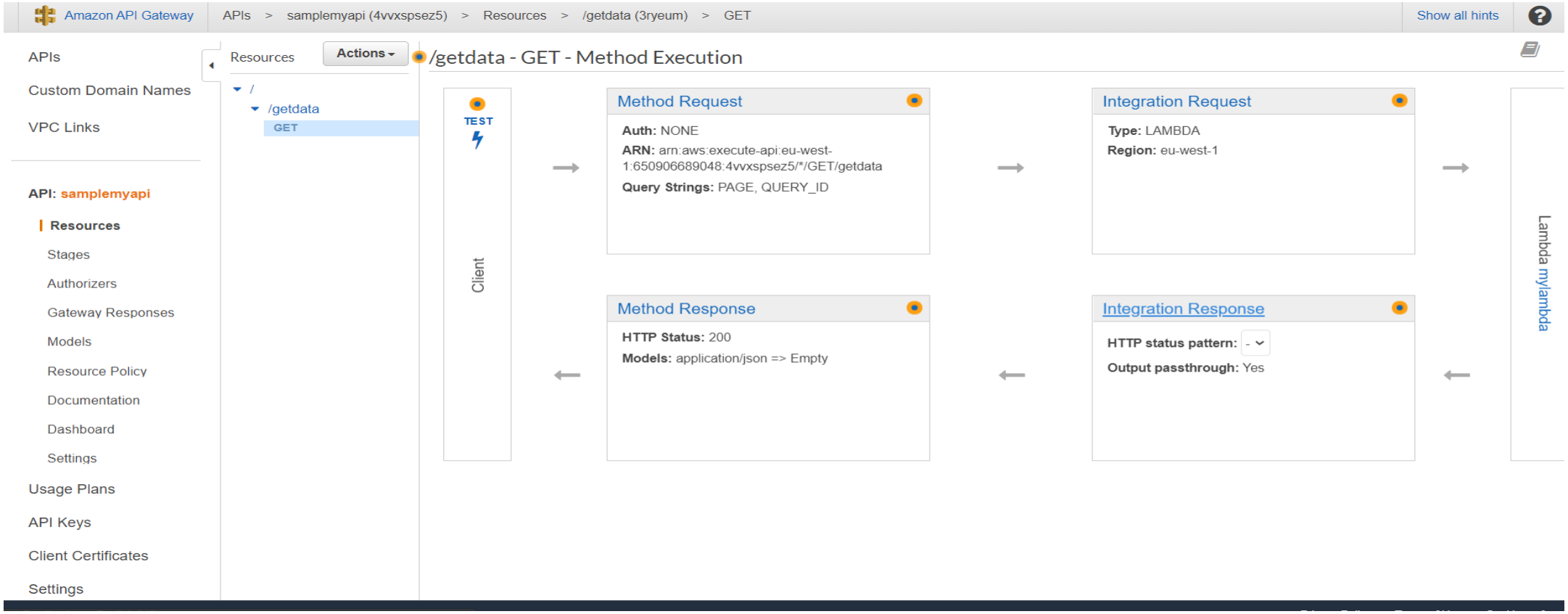
AWS Lambda:

- Lambda is a compute service that lets you run code in Serverless
- With Lambda, you can run code for virtually any type of application or backend service.
- Here in use-case to fetch the records from respective DWH whenever the front end calls the respective REST endpoint.
- The call perhaps trigger the API Gateway which in turn calls the lambda function and fetches the records.



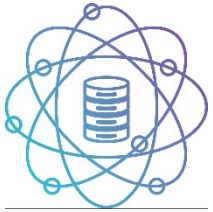
API Gateway:

- **AWS** service for creating, publishing, maintaining, monitoring, and securing REST, HTTP, and WebSocket **APIs** at any scale.
- API developers can create APIs that access AWS or other web services, as well as data stored in the AWS Cloud.
- As an API Gateway API developer, you can create APIs for use in your own client applications. Or you can make your APIs available to third-party app developers.



Data Science 101

Data Science 101 - Agenda

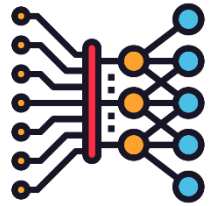


Overview of Data Science

Statistics

Exploratory Data Analysis (EDA)

Feature Engineering



Modeling

A Billion-Dollar Question - Why Datascience?

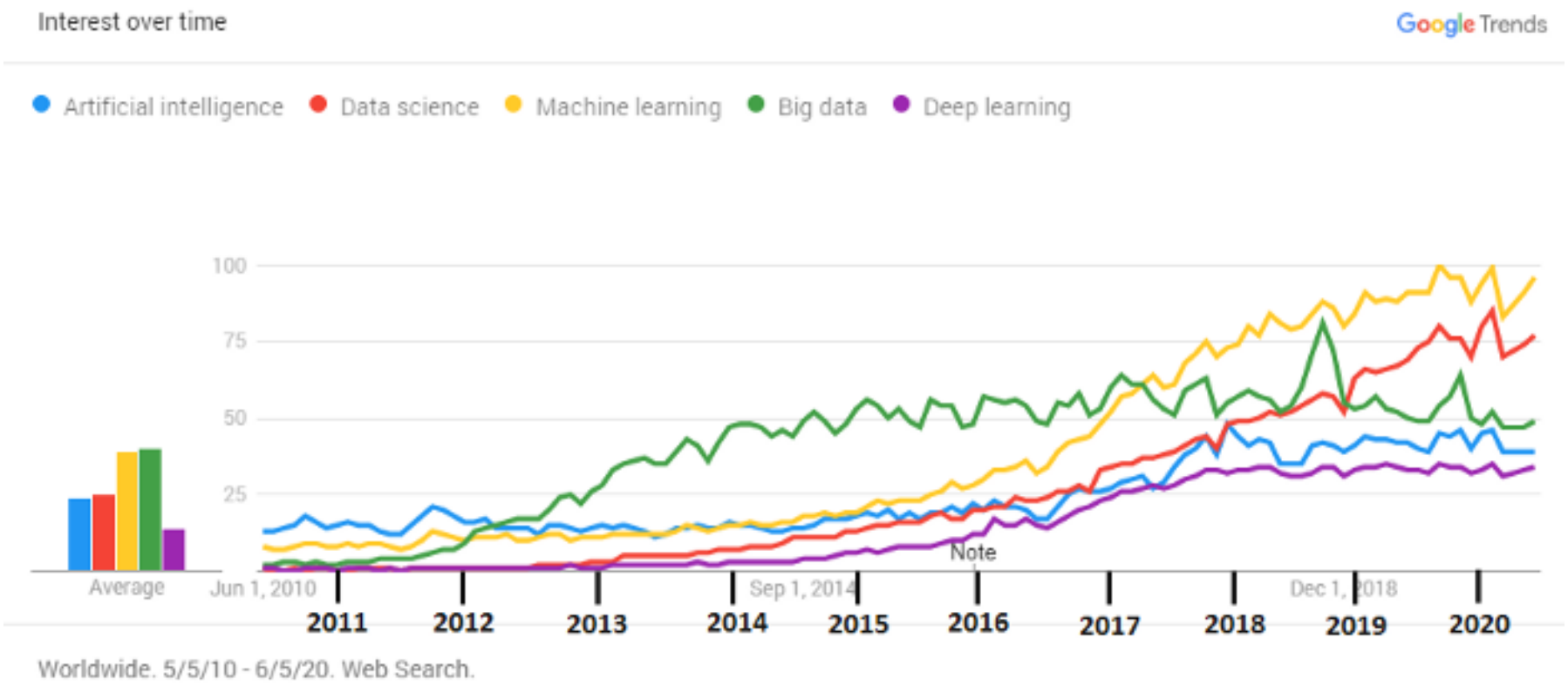
Making sense of data will reduce the horrors of uncertanity for organizations

Example:

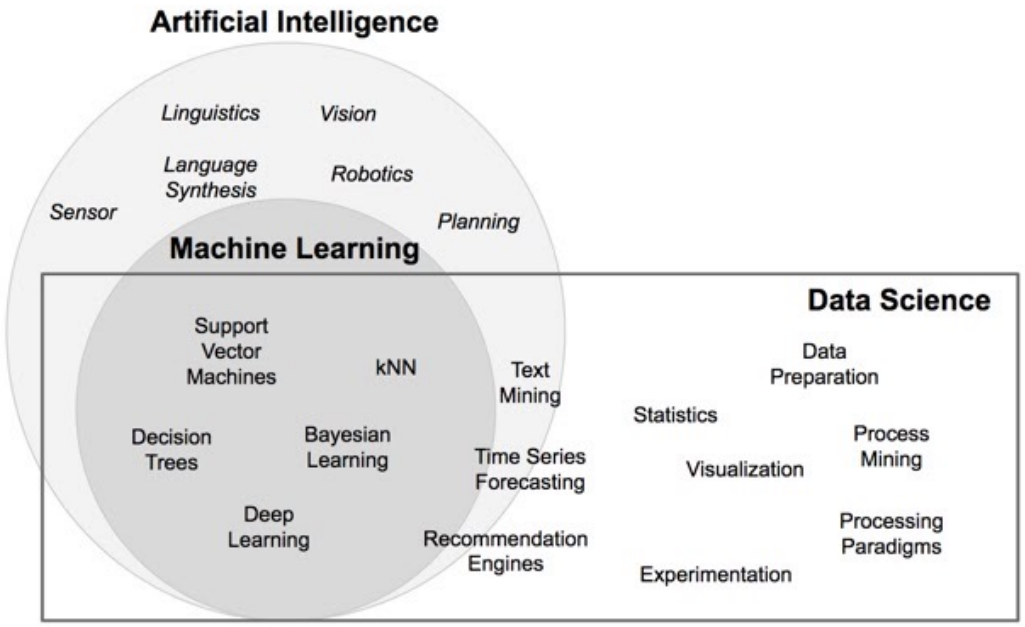
1. In 2003, iTunes took 100 months to reach 100 million users, while for Pokemon in 2016, it took days to reach the million mark.

2. Fraud detection, using simple features like geolocation, number of active session, device detail, asset type, login pattern/timing etc can be considered as a fraud login or fraud user.

Future of Data Science?

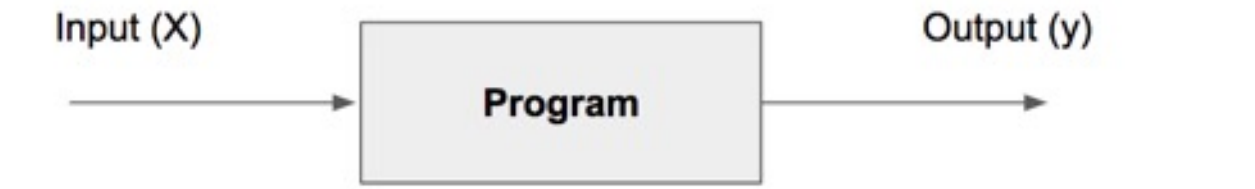


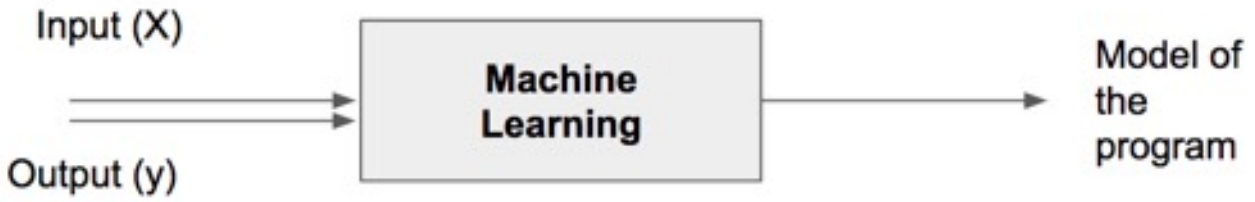
What is Data Science?



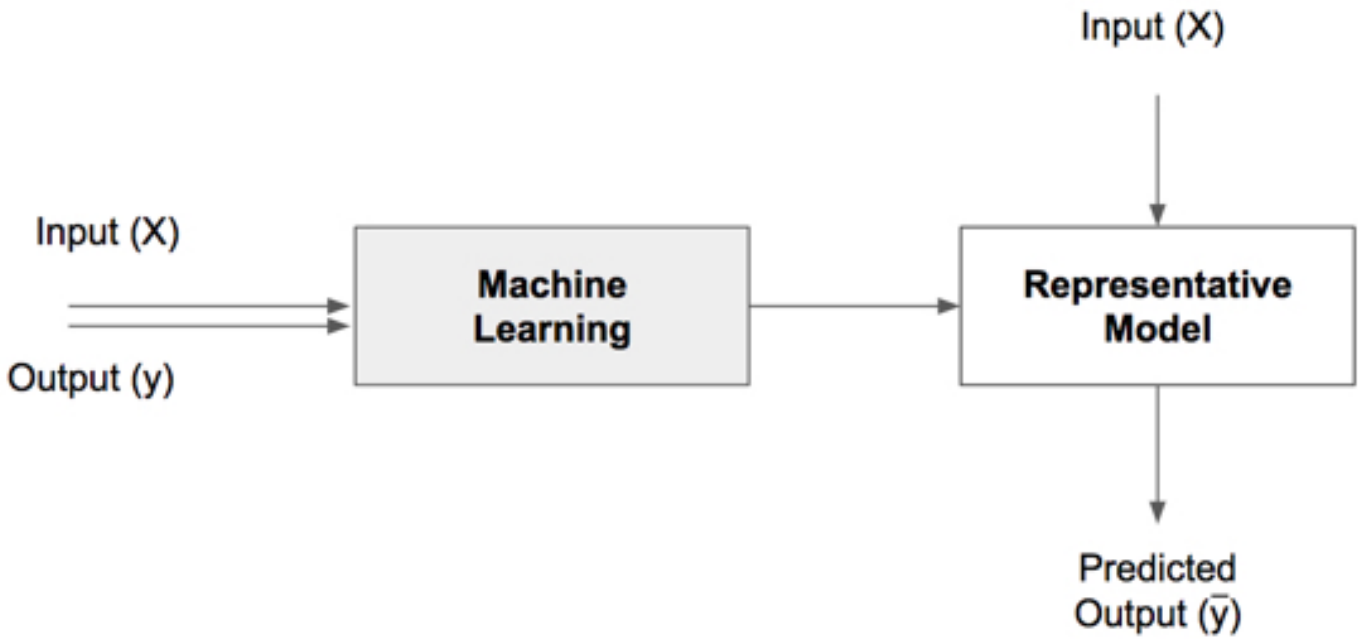
"**Data science** is an inter-disciplinary field that uses **scientific** methods, processes, algorithms and systems to **extract knowledge and insights** from **structured and unstructured data**, and apply knowledge and actionable insights from data across a broad range of application domains", *Wikipedia*

How it works?

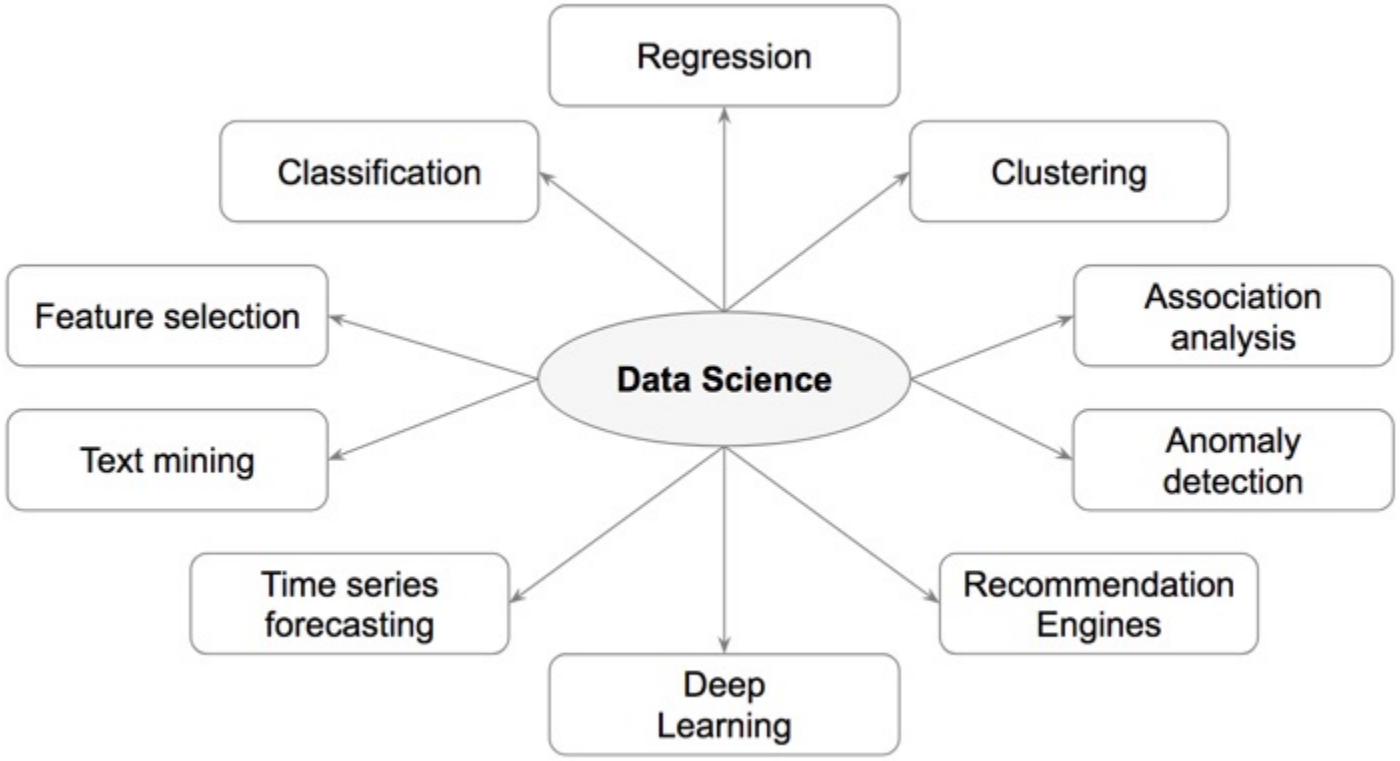




How it works?



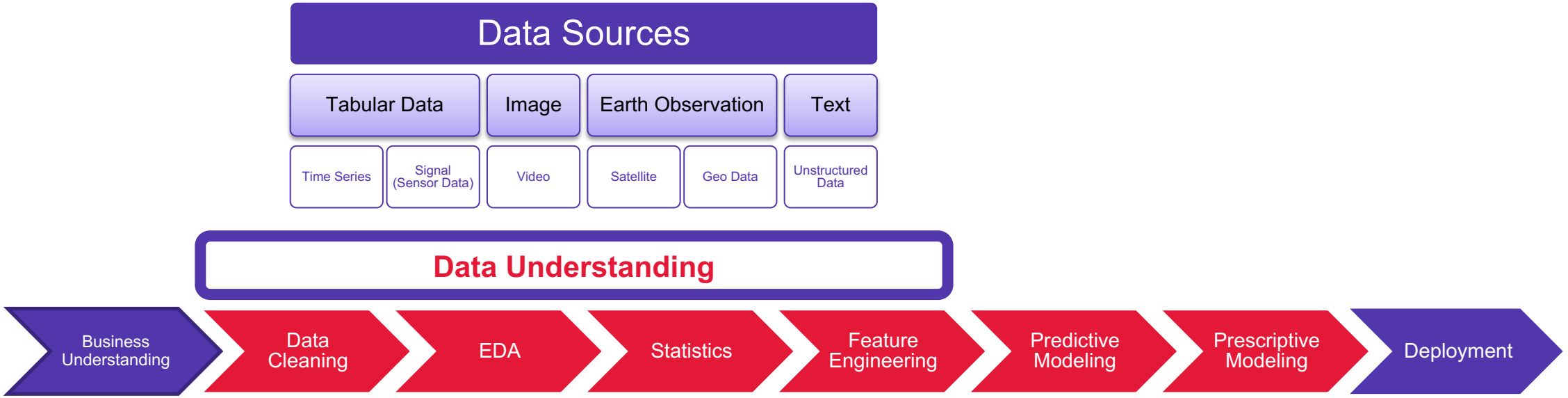
How Data Science can be used?



Overview of Data Science use cases

Tasks	Description	Algorithms	Examples
Classification	Predict if a data point belongs to one of predefined classes. The prediction will be based on learning from known data set.	Decision Trees, Neural networks, Bayesian models, Induction rules, K nearest neighbors	Assigning voters into known buckets by political parties eg: soccer moms. Bucketing new customers into one of known customer groups.
Regression	Predict the numeric target label of a data point. The prediction will be based on learning from known data set.	Linear regression, Logistic regression	Predicting unemployment rate for next year. Estimating insurance premium.
Anomaly detection	Predict if a data point is an outlier compared to other data points in the data set.	Distance based, Density based, LOF	Fraud transaction detection in credit cards. Network intrusion detection.
Time series	Predict if the value of the target variable for future time frame based on history values.	Exponential smoothing, ARIMA, regression	Sales forecasting, production forecasting, virtually any growth phenomenon that needs to be extrapolated
Clustering	Identify natural clusters within the data set based on inherit properties within the data set.	K means, density based clustering - DBSCAN	Finding customer segments in a company based on transaction, web and customer call data.
Association analysis	Identify relationships within an itemset based on transaction data.	FP Growth, Apriori	Find cross selling opportunities for a retailor based on transaction purchase history.

What involves in Data Science Process?



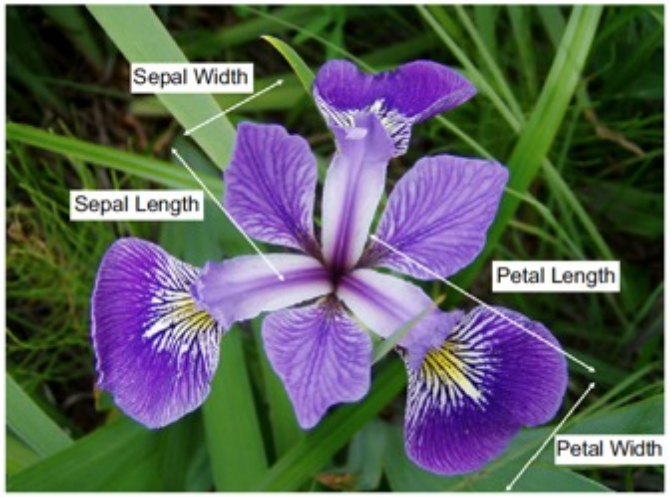
1. Business Understanding

Gaining information on:

- Objective of the problem
- Subject area of the problem
- Data

Table 2.1 Data Set

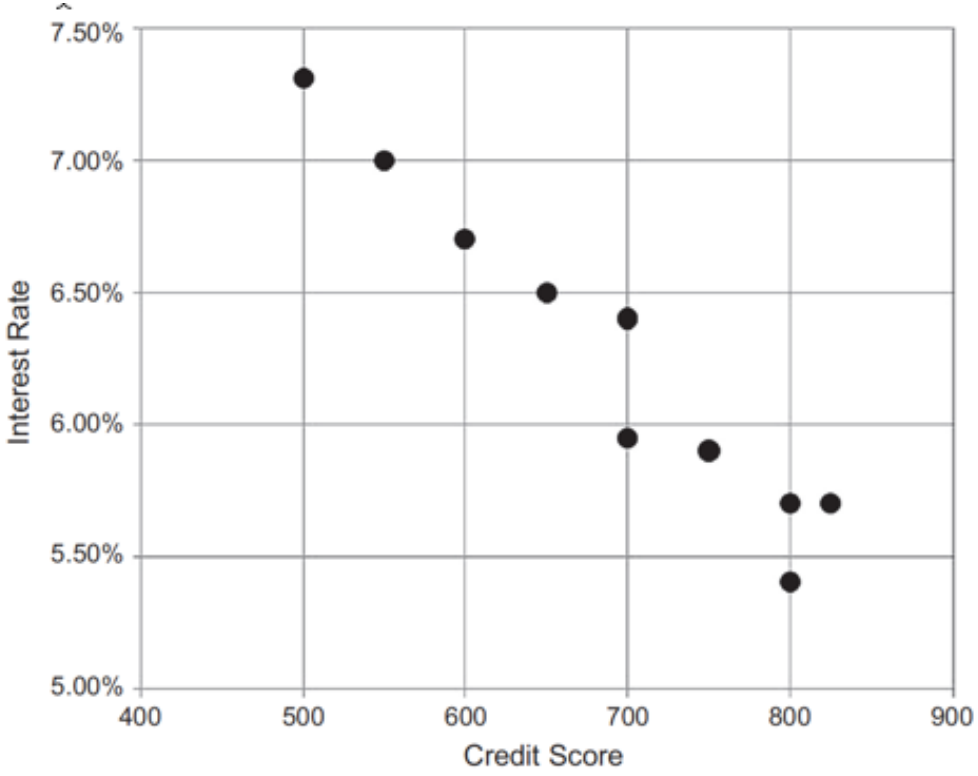
Borrower ID	Credit Score	Interest Rate
01	500	7.31%
02	600	6.70%
03	700	5.95%
04	700	6.40%
05	800	5.40%
06	800	5.70%
07	750	5.90%
08	550	7.00%
09	650	6.50%
10	825	5.70%



2. Data Understanding

Data Cleaning

1. Data quality
2. Handling mission values
3. Data type conversion
4. Transformations
5. Outliers



2. Data Understanding
Descriptive Statistics - Univariate

Characteristics of the Data Set	Measurement Technique
Center of the data set	Mean, media, mode
Spread of the data set	Range, variance, standard deviation
Shape of the distribution of the data set	Symmetry, skewness, kurtosis

Table 3.1 Iris Data Set and Descriptive Statistics (Fisher, 1936)

Observation	Sepal Length	Sepal Width	Petal Length	Petal Width
1	5.1	3.5	1.4	0.2
2	4.9	3.1	1.5	0.1
...	...	...	...	...
49	5	3.4	1.5	0.2
50	4.4	2.9	1.4	0.2
Statistics	**Sepal Length**	**Sepal Width**	**Petal Length**	**Petal Width**
Mean	5.006	3.418	1.464	0.244
Median	5.000	3.400	1.500	0.200
Mode	5.100	3.400	1.500	0.200
Range	1.500	2.100	0.900	0.500
Standard Deviation	0.352	0.381	0.174	0.107
Variance	0.124	0.145	0.030	0.011

2. Data Understanding

Descriptive Statistics - Multivariate

Covariance:

The covariance is a measure of how much two random variables vary together. It's similar to variance, The variance tells you how a single variable varies, covariance tells you how two variables vary together.

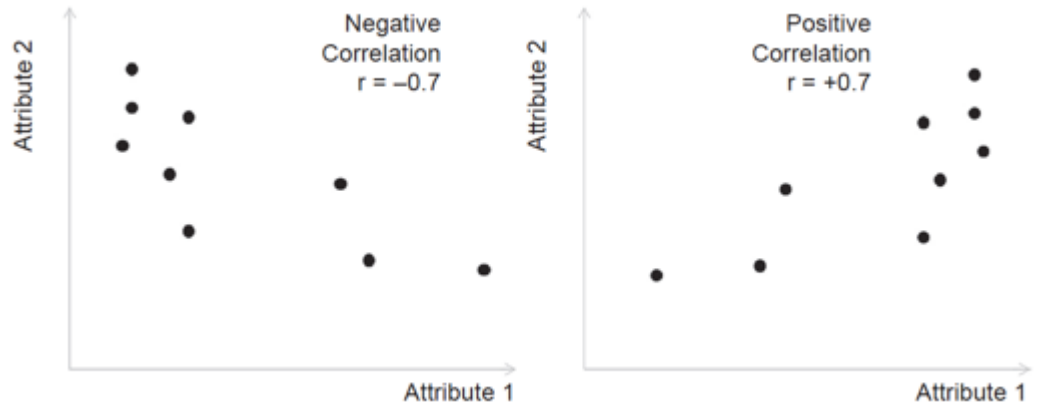
$$\text{cov(X,Y)} = \sigma_{X,Y} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

Correlation:

Degree to which two random variables move in coordination with one another. We can get the correlation by dividing the covariance by the standard deviations to get the correlation coefficient.

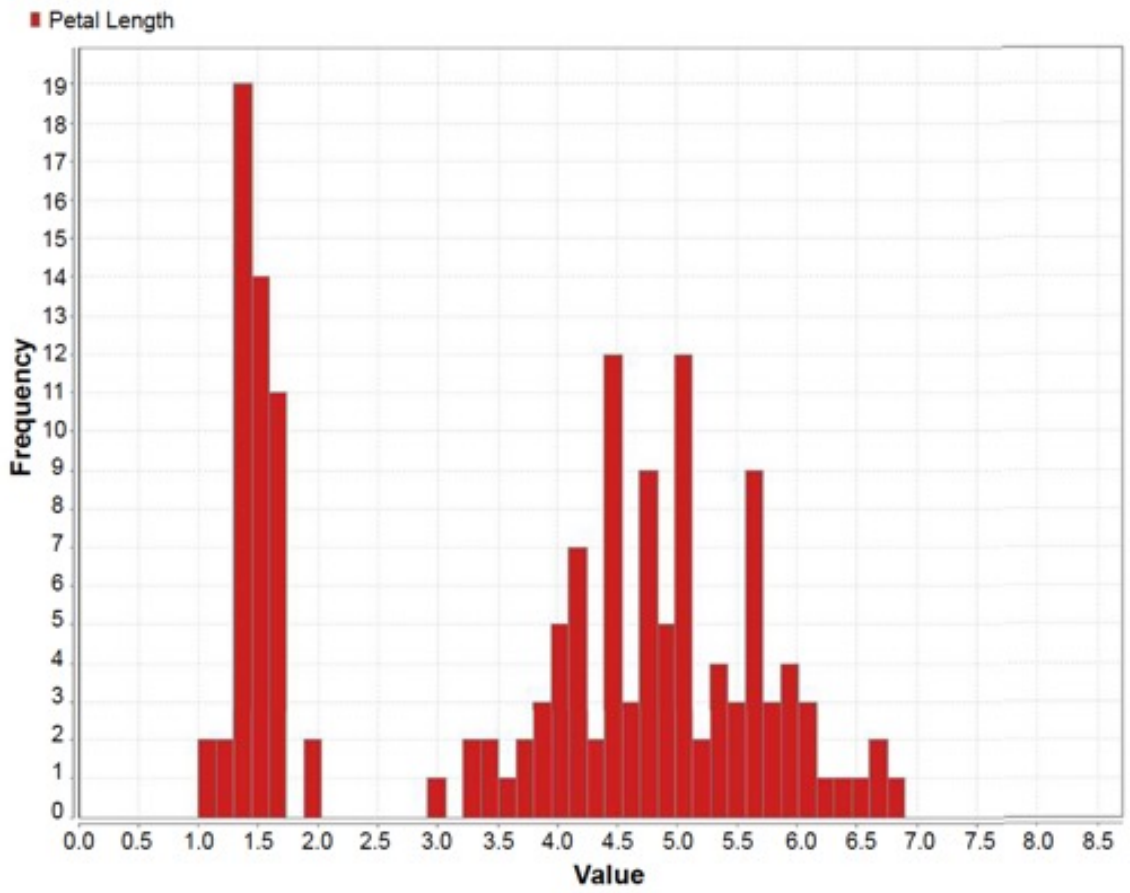
$$\rho_{X;Y} = \frac{\text{cov(X,Y)}}{\sigma_X \sigma_Y} = \frac{variability\ due\ to\ codependence}{independent\ variability}$$

observation i: {sepal length, sepal width, petal length, petal width}



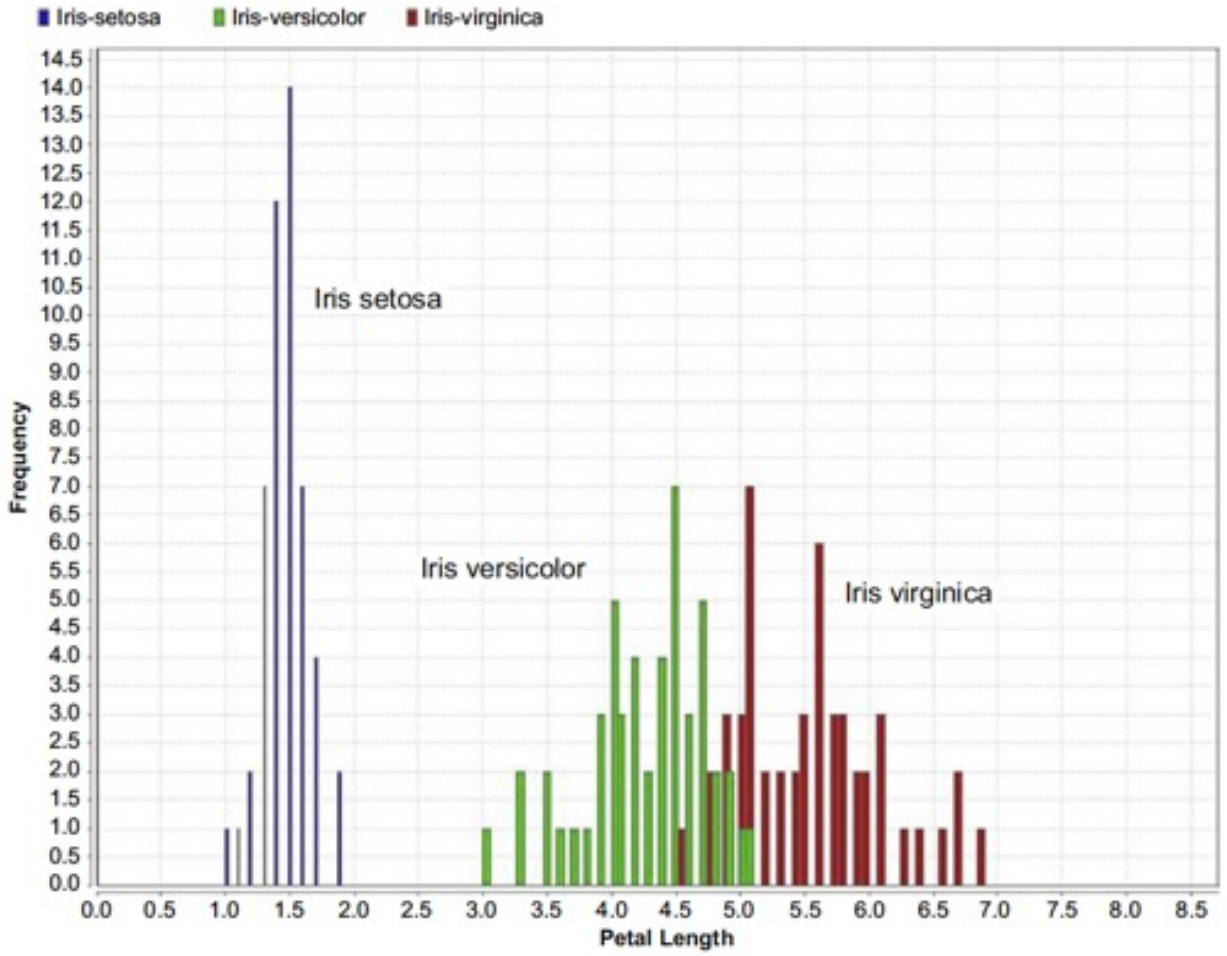
2. Data Understanding
Exploratory Data Analysis

Histogram



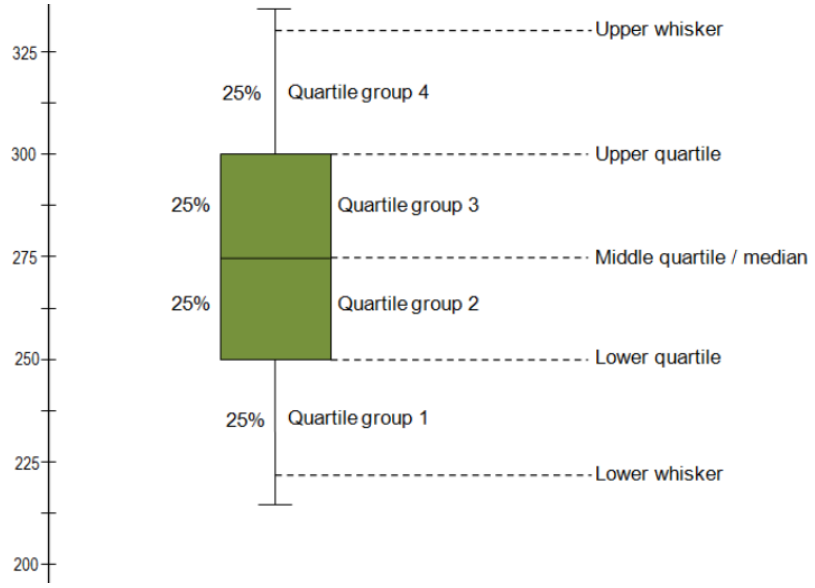
2. Data Understanding
Exploratory Data Analysis

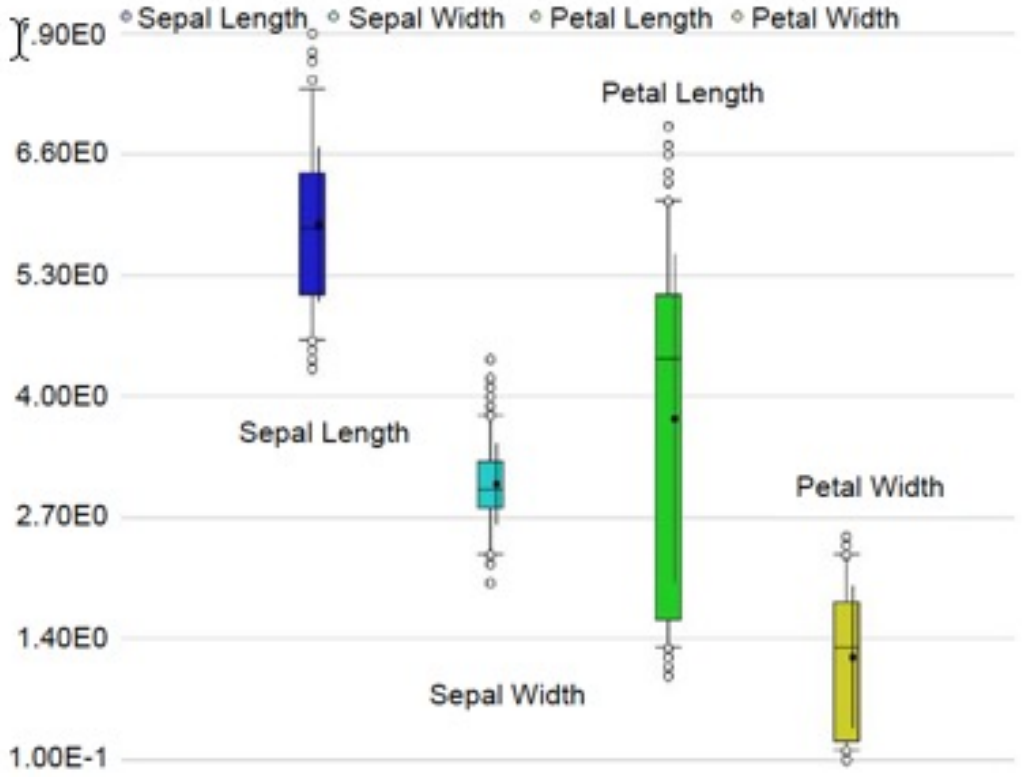
Class Stratified Histogram



2. Data Understanding
Exploratory Data Analysis

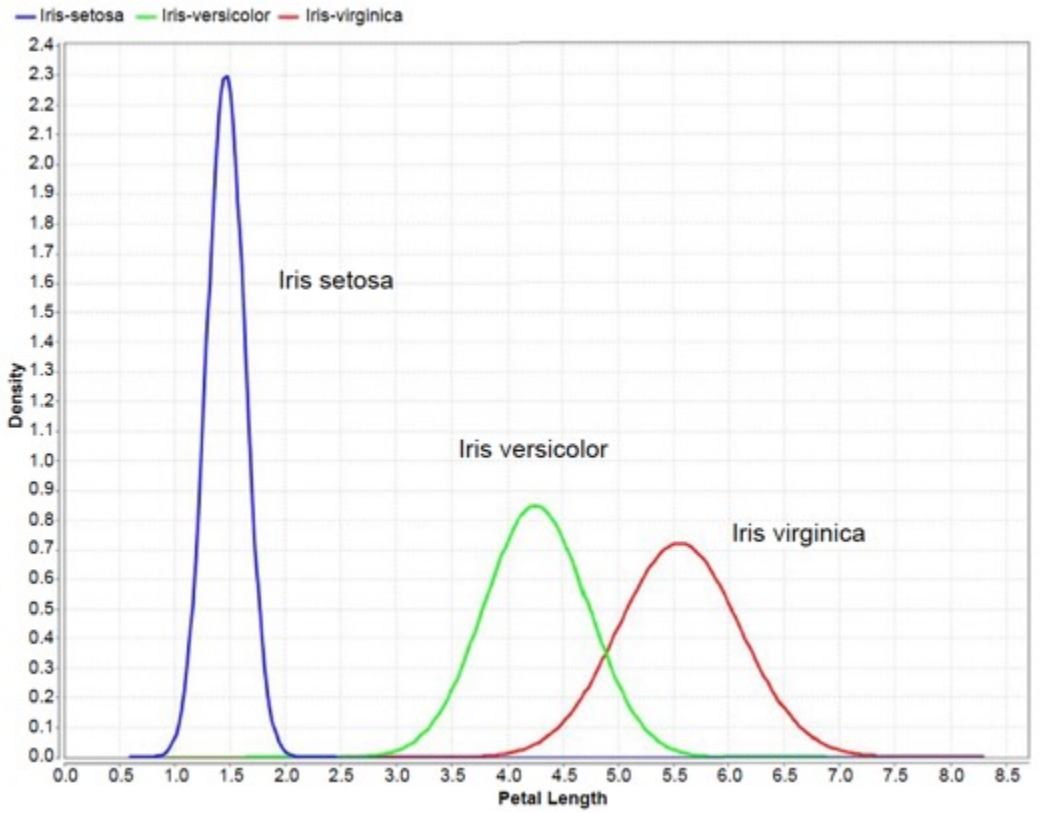
Quantile plot





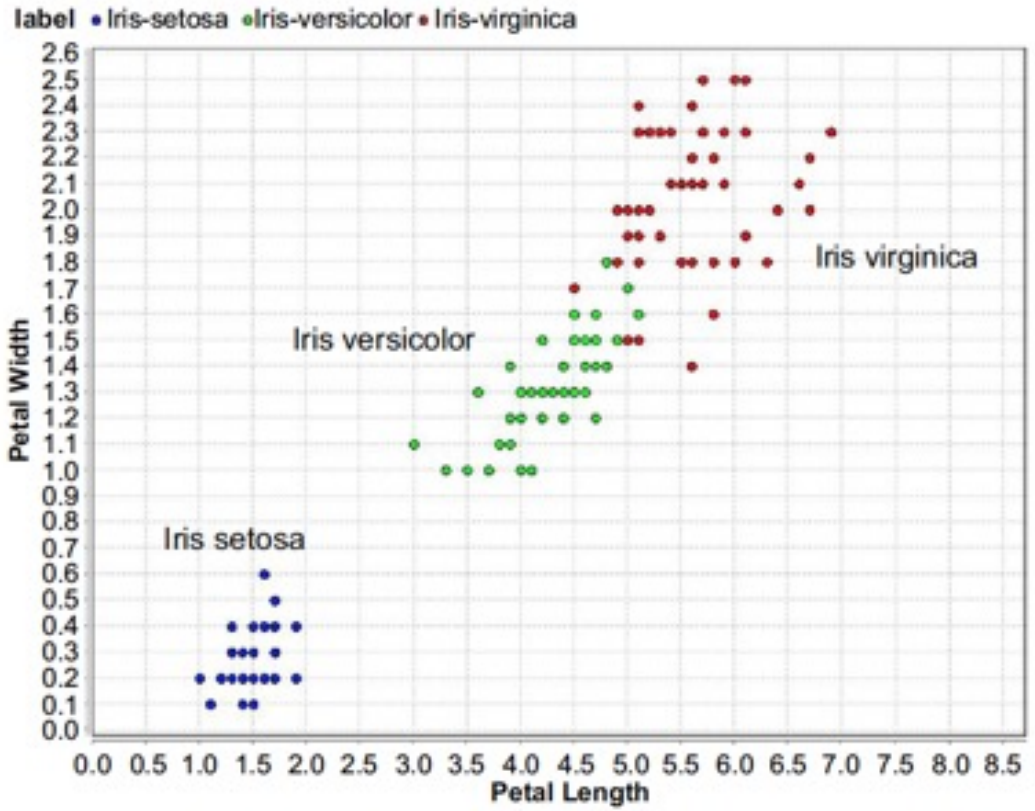
2. Data Understanding
Exploratory Data Analysis

Distribution plot



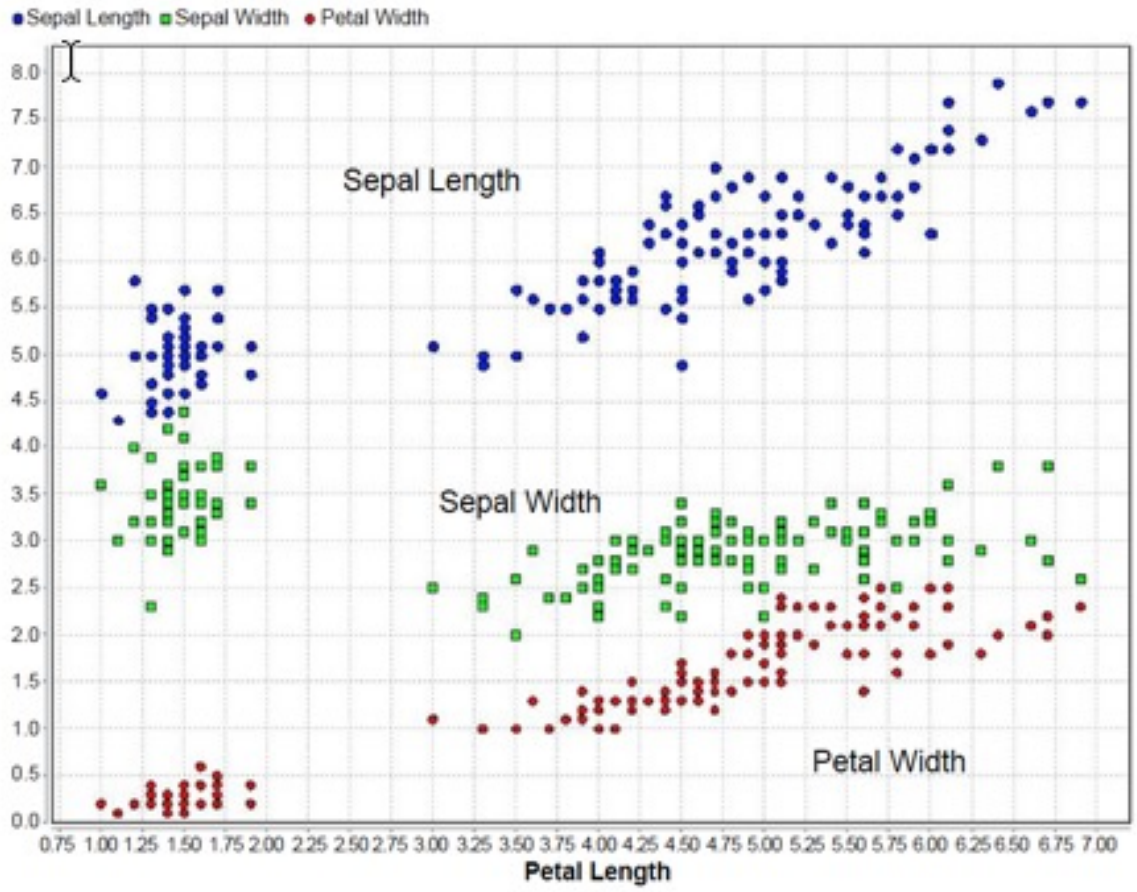
2. Data Understanding
Exploratory Data Analysis

Scatter plot



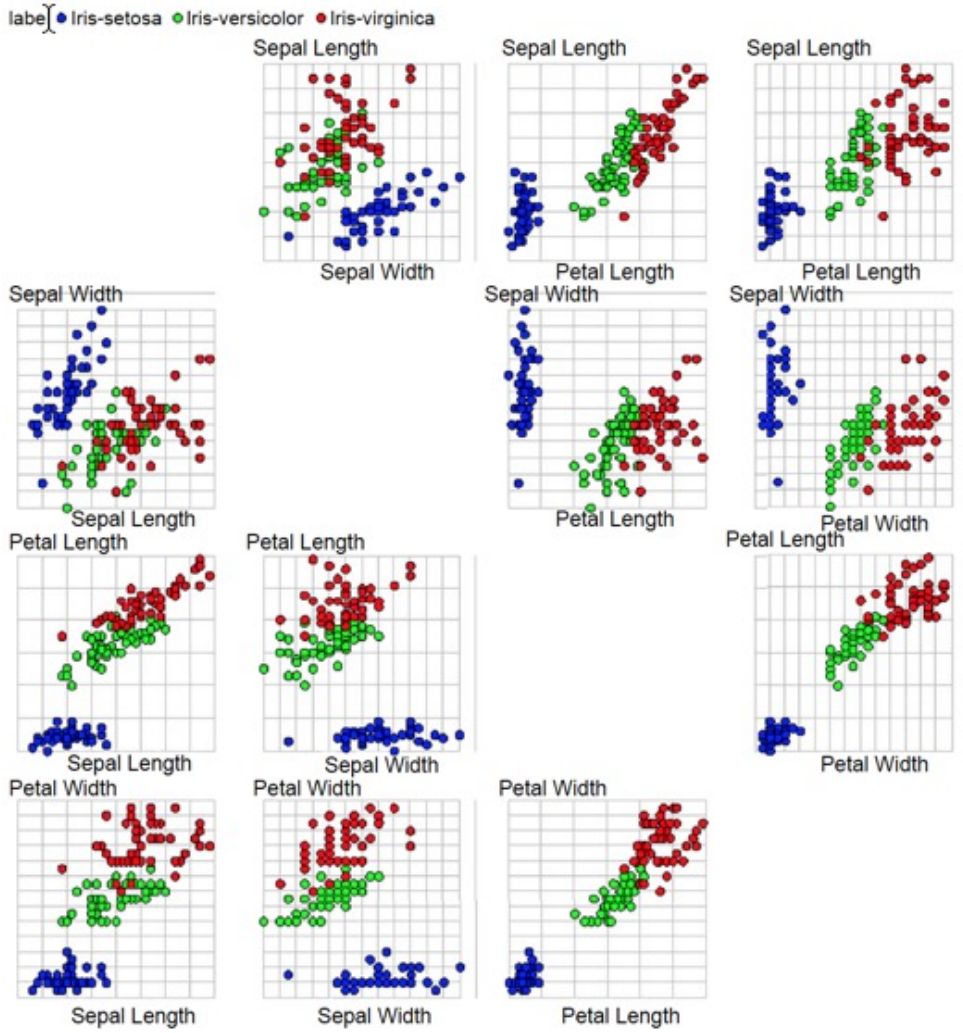
2. Data Understanding
Exploratory Data Analysis

Scatter multiple



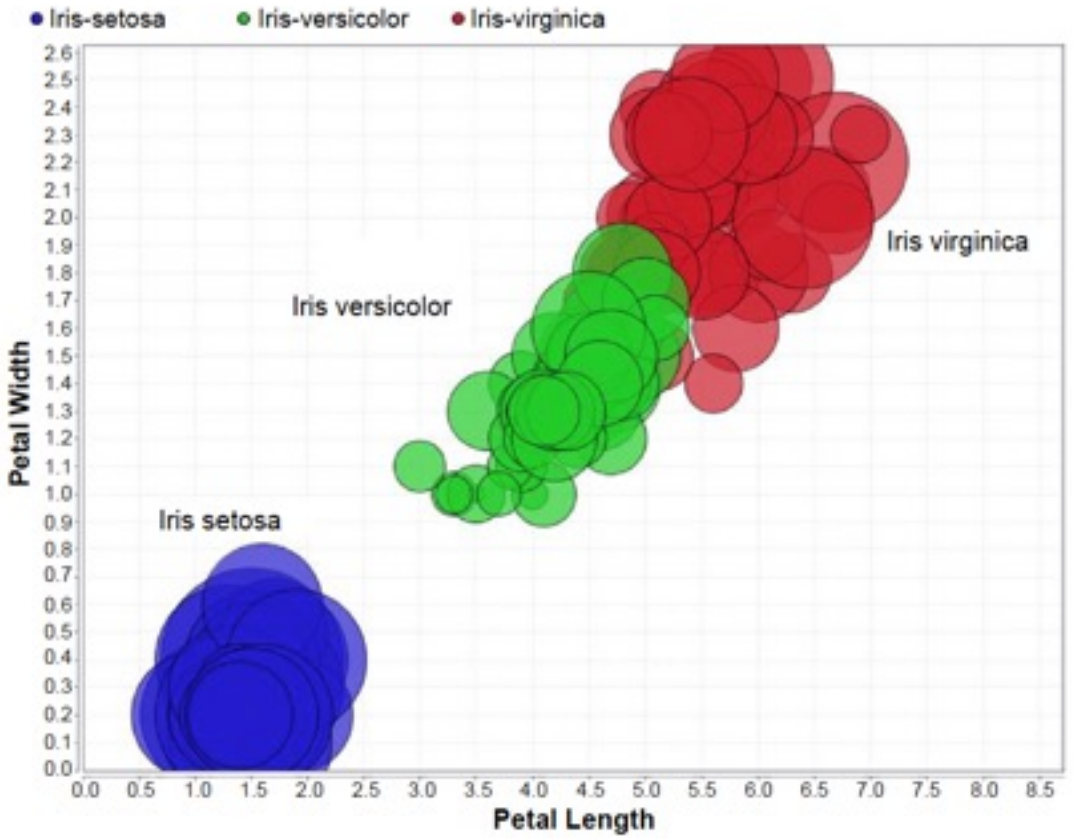
2. Data Understanding
Exploratory Data Analysis

Multiple scatter matrix



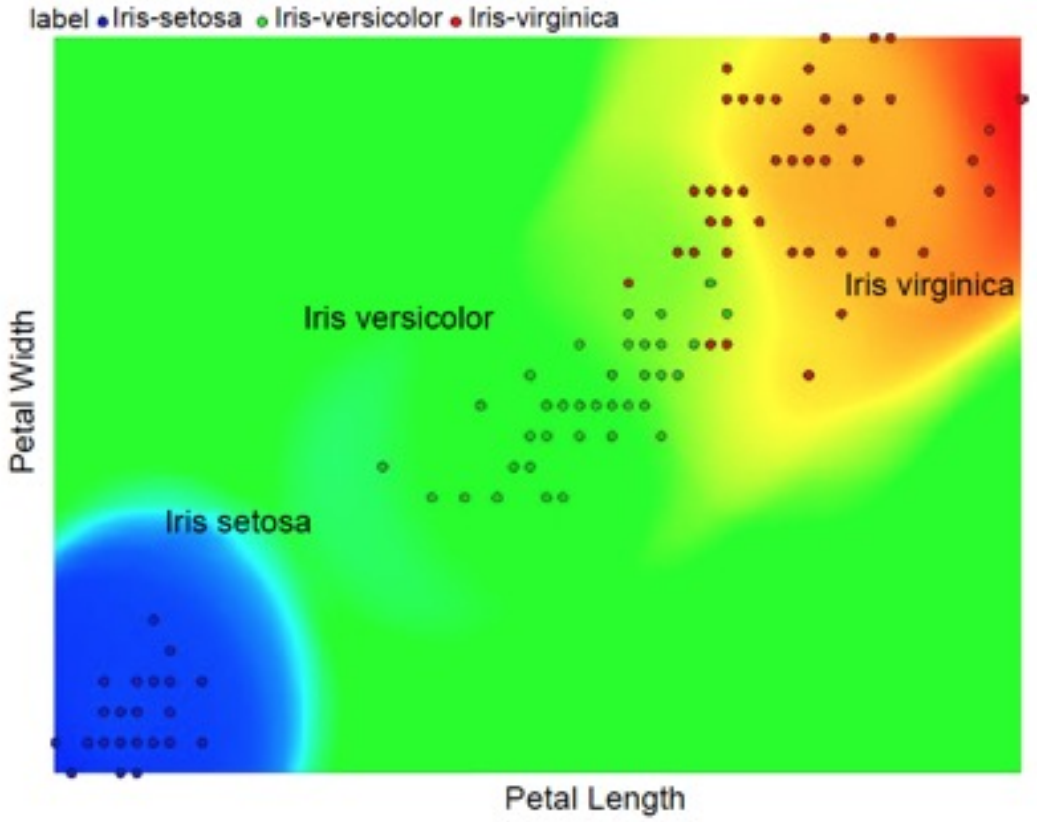
2. Data Understanding
Exploratory Data Analysis

Bubble Plot



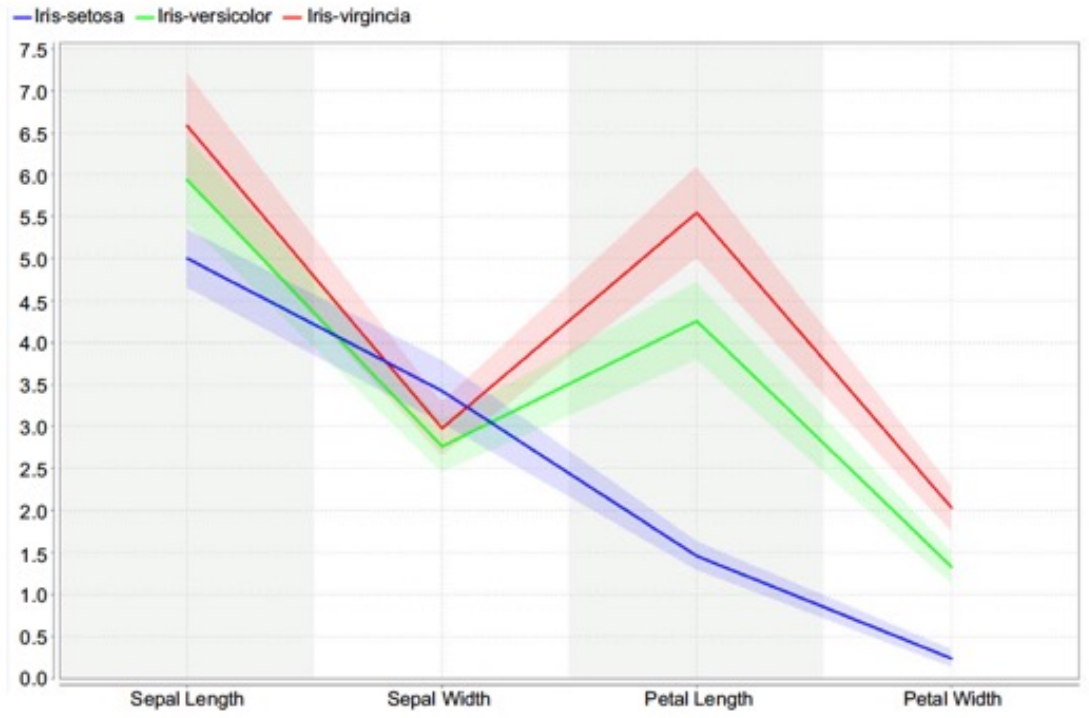
2. Data Understanding
Exploratory Data Analysis

Density Chart

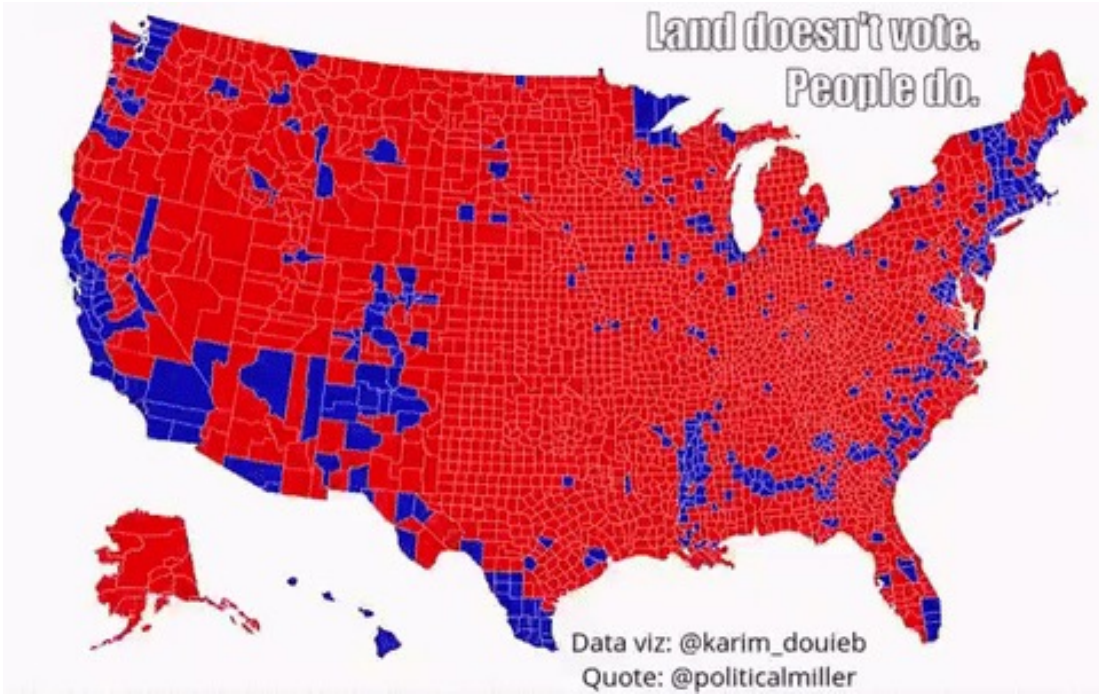


2. Data Understanding
Exploratory Data Analysis

Deviation Chart



Visualizations Can be Misleading



Roadmap for data exploration

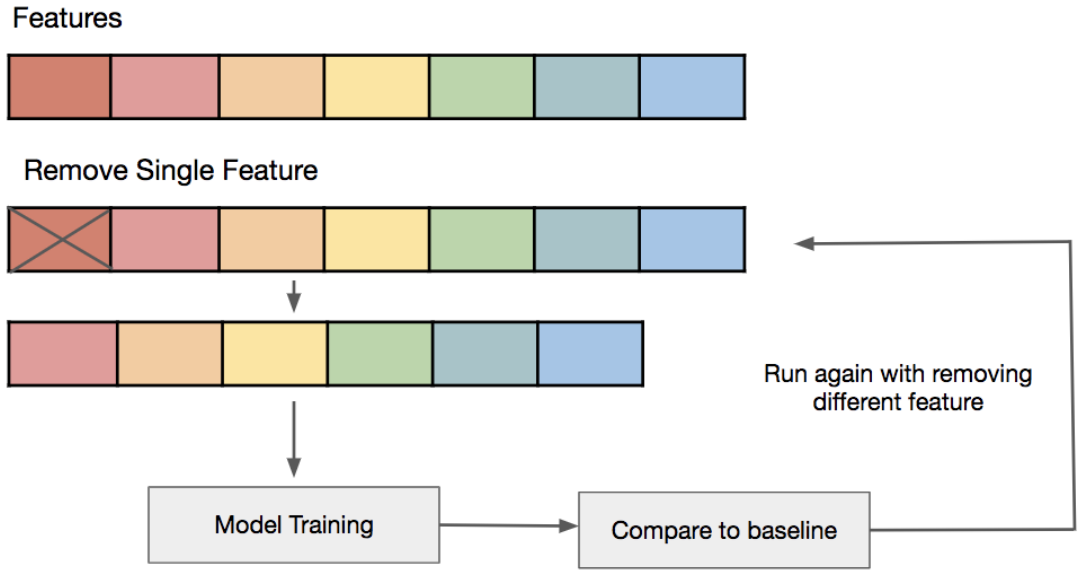
1. Organize the data set
2. Find the central point for each attribute
3. Understand the spread of the attributes
4. Visualize the distribution of each attributes
5. Pivot the data
6. Watch out for outliers
7. Understanding the relationship between attributes
8. Visualize the relationship between attributes
9. Visualization high dimensional data sets

Kotu, V., & Deshpande, B. (2014). *Predictive analytics and data mining: concepts and practice with rapidminer*. Morgan Kaufmann.

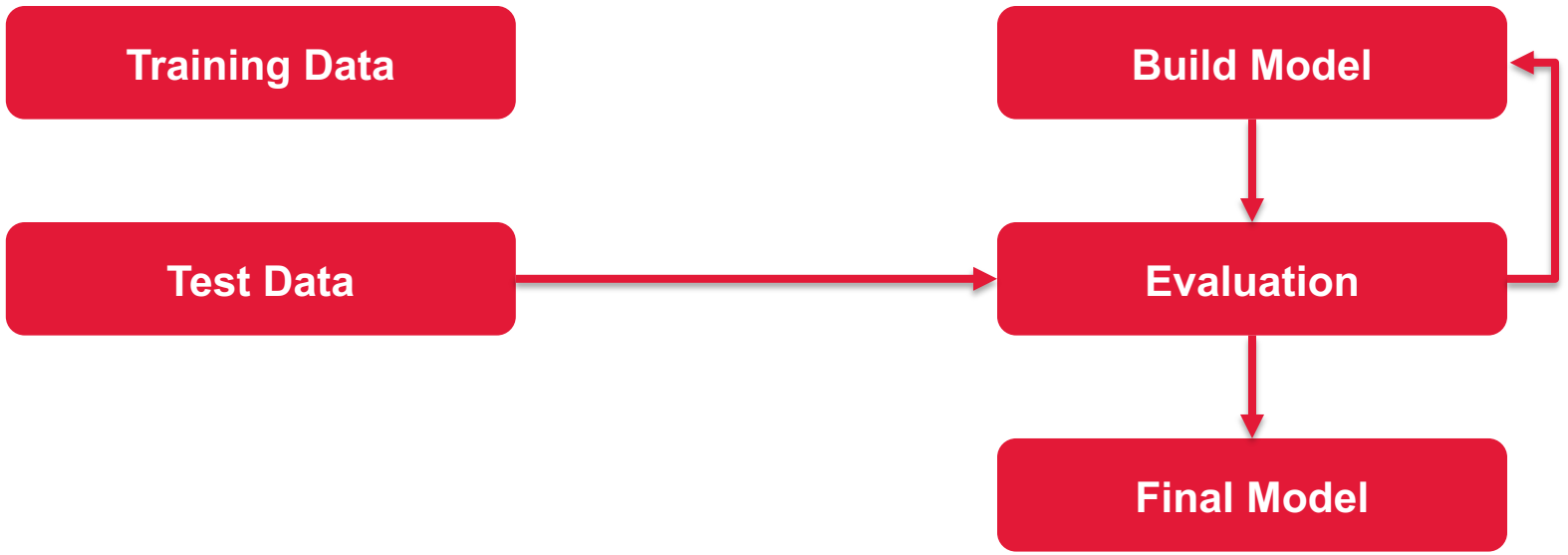
2. Data Understanding
Feature Selection

Process of reducing number of input variables when developing a predictive model.

- Correlation Statistics
  - i.e. Pearson's Correlation Coefficient
- Selection Method
  - i.e. Select the top percentile variables
- Transform Variables
  - i.e. Transform categorical variable to ordinal



3. Modeling



3. Modeling

Spliting training and test data sets

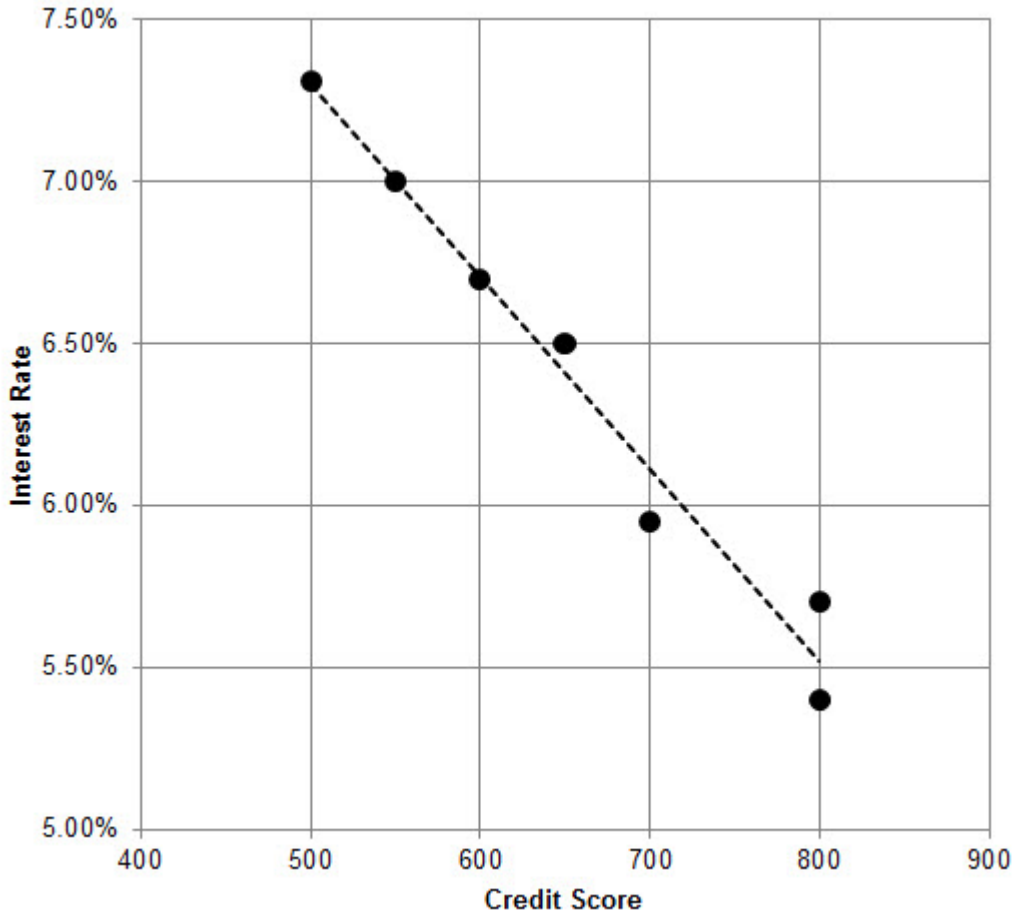
Table 2.3 Training Data Set

Borrower	Credit Score (X)	Interest Rate (Y)
01	500	7.31%
02	600	6.70%
03	700	5.95%
05	800	5.40%
06	800	5.70%
08	550	7.00%
09	650	6.50%

Table 2.4 Test Data Set

Borrower	Credit Score (X)	Interest Rate (Y)
04	700	6.40%
07	750	5.90%
10	825	5.70%

3. Modeling



$$y = 0.1 + \frac{6}{100,000}x$$

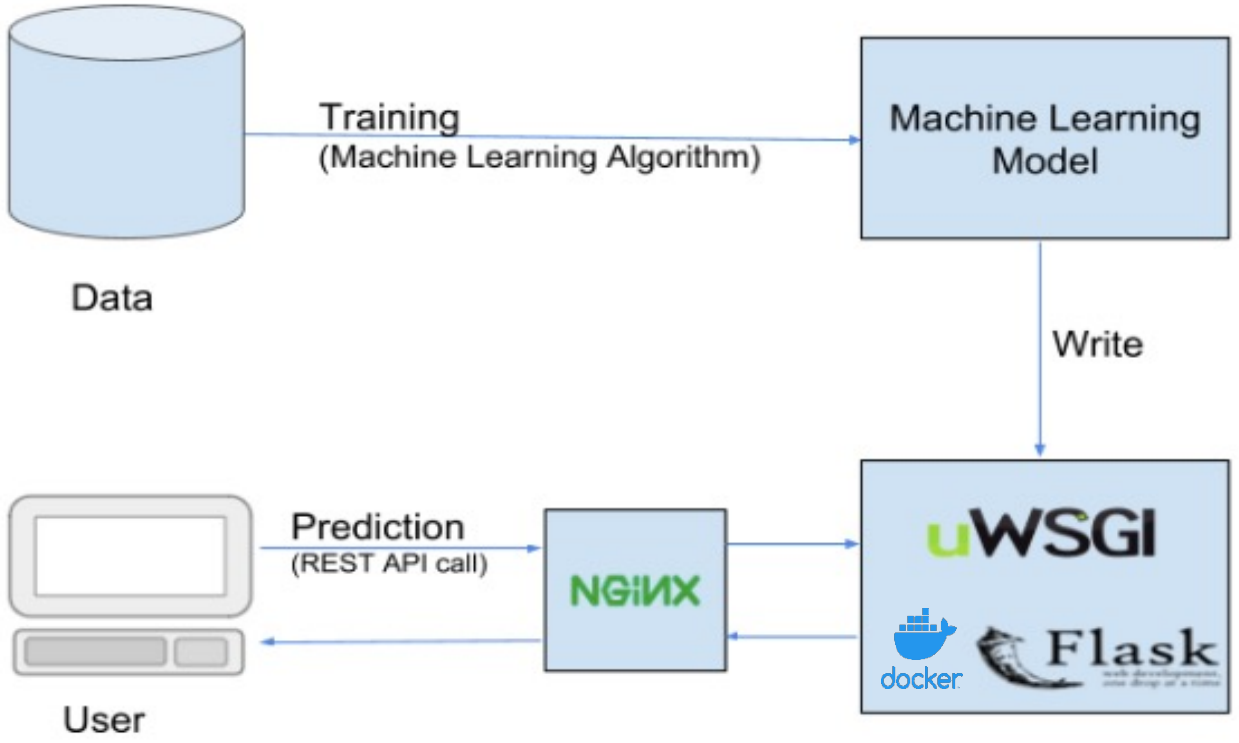
3. Modeling

Evaluation of test dataset

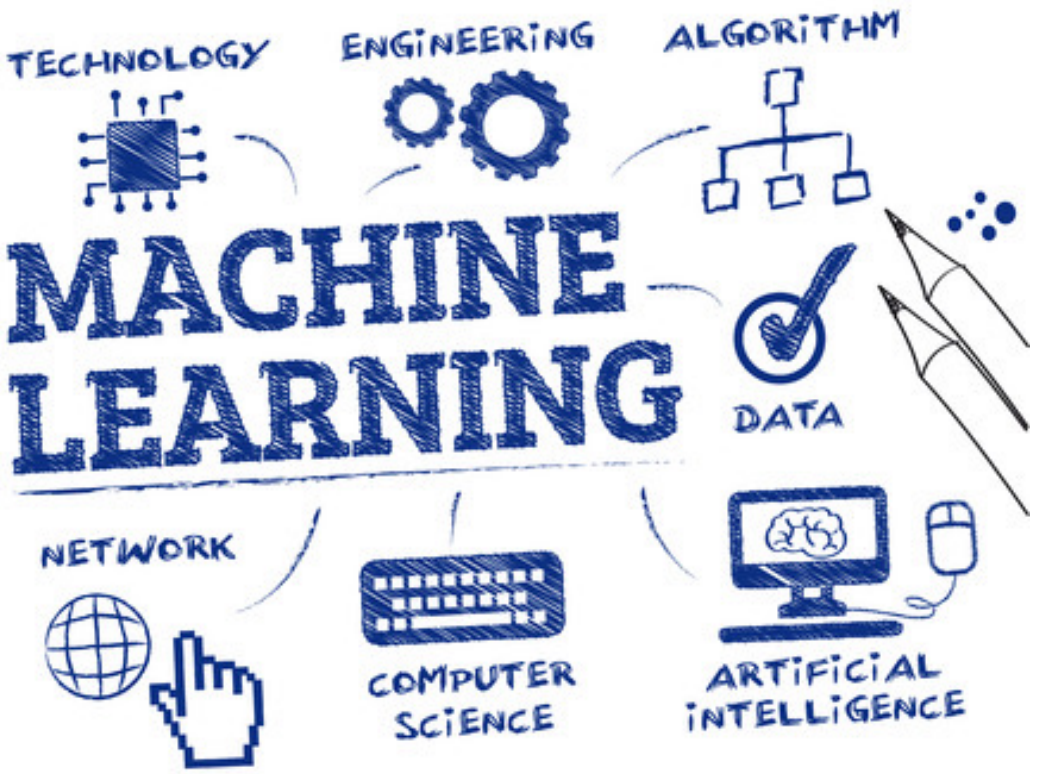
Table 2.5 Evaluation of Test Data Set

Borrower	Credit Score (X)	Interest Rate (Y)	Model Predicted (Y)	Model Error
04	700	6.40%	6.11%	-0.29%
07	750	5.90%	5.81%	-0.09%
10	825	5.70%	5.37%	-0.33%

4. Deployment

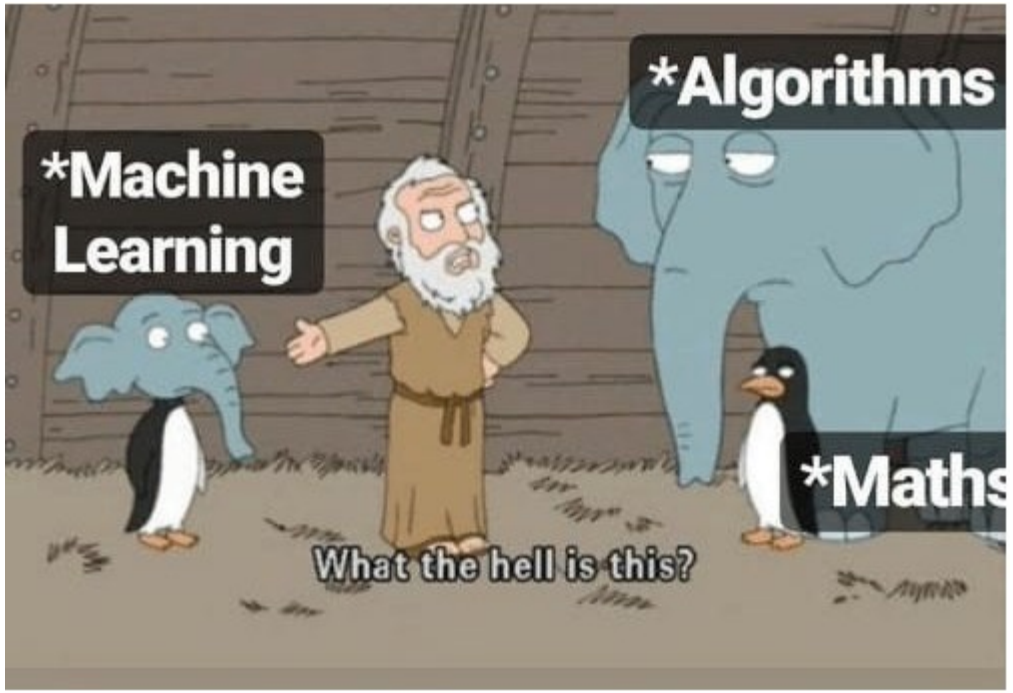


Topics in Machine Learning



What is Machine Learning?

- **“Machine learning** (**ML**) is the study of **computer algorithms** that improve automatically through experience” (Wikipedia: *Mitchell, Tom (1997). Machine Learning*)
- “Machine learning aims to teach computers how to learn and act **without being explicitly programmed**.” (deepai.org)
- “Machine learning is a branch of artificial intelligence (AI) focused on building applications that learn from data and **improve their accuracy over time** without being programmed to do so.” (ibm.com)
- **Learning with data**



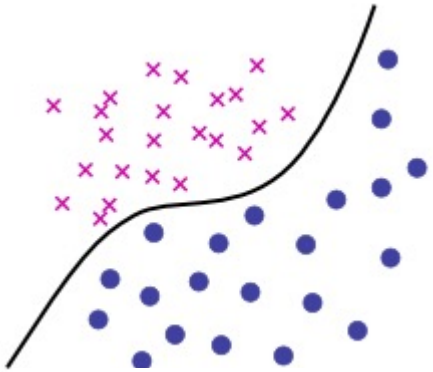
MATH + ALGORITHM = MACHINE LEARNING

Examples

- Given the historical data of power consumption in Europe can you **predict** the power consumption for 2021?
- Given a dataset of housing prices in Munich can you **estimate** the price of my house?
- Can we automatically **detec**t the tone of Elon Musk tweets?
- Can we **differentiate** roads/buildings from satellite images?
- We have various sensors in our factory can we automatically **detect** if anything goes wrong?
- We have a dataset of the users of our service. Can we **segment** them into a few groups to offer more personalized service to similar groups?
- Can we train our robot to reach from A to B in the **least amount** of time while avoiding the obstacles?

Machine Learning Problem Categories

- **Regression:** The goal is to estimate the value for a target variable
  - **predict something**
  - **estimate something**



- **Classification:** The goal is to divide the data into a number of groups
  - **categorize something**
  - **detect something**

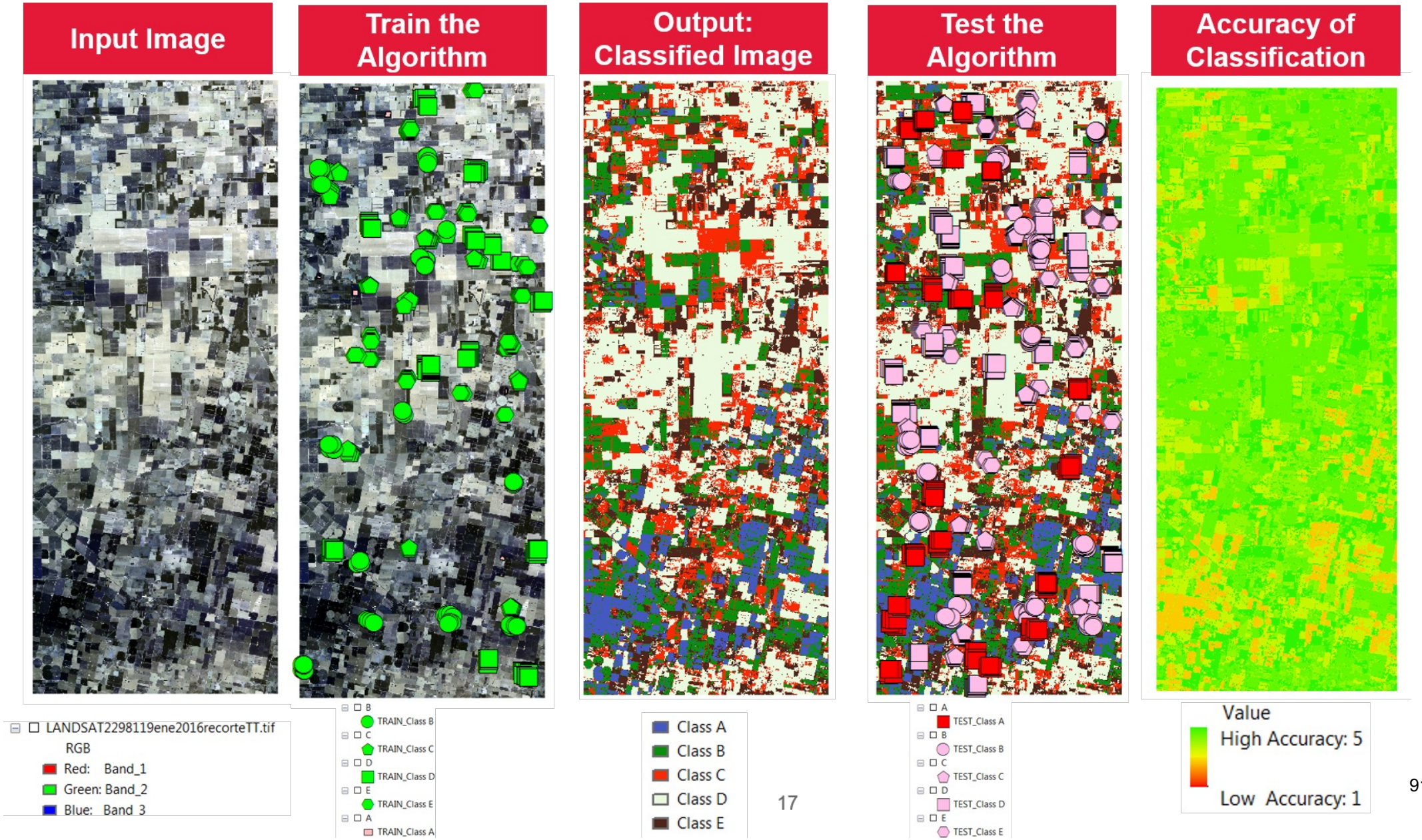
- **Optimization:** The goal is to minimize/maximize some function
  - **best performance**
  - **least resources**

How do we make Machines Learn?

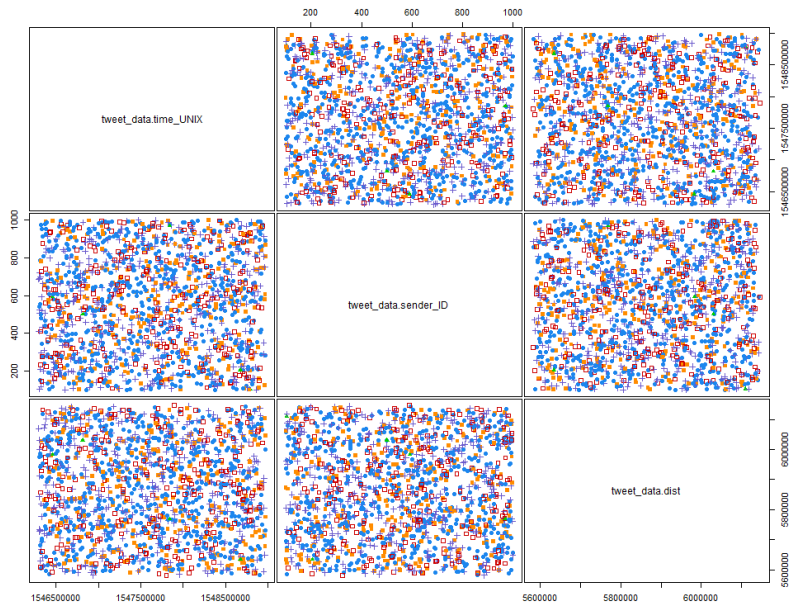
- Define a Rule/Model to make the **Machine learn**:
  - **Supervise the Learning:** If these are the conditions, this is the result..
  - **Unsupervised Learning:** Sometimes use the data and let machines figure out the relationships themselves

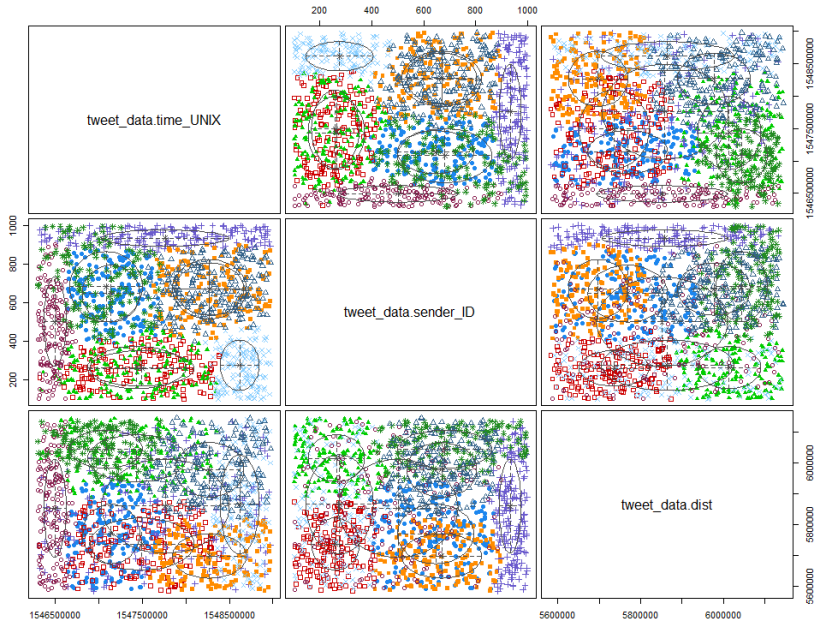
- Define a task for the **Machine to Do**:
  - Based on the rule/model that machine has learnt:
    - Predict something
    - Classify something

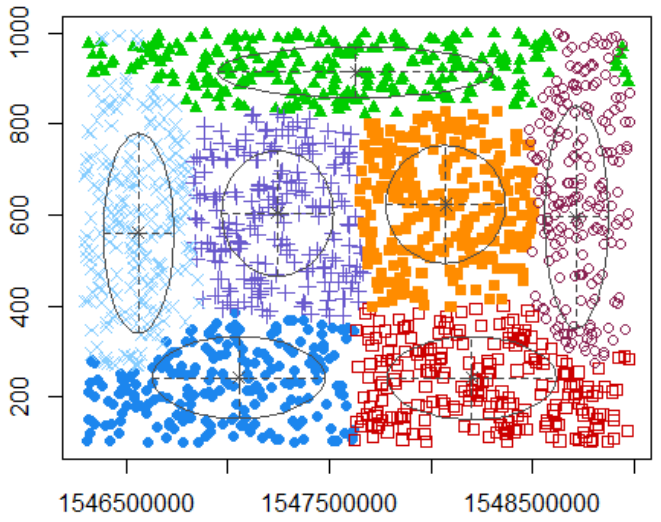
Task: Classify the Satellite Image into different Categories



Task: Classify Tweets based on Topics Automatically







Supervised Learning

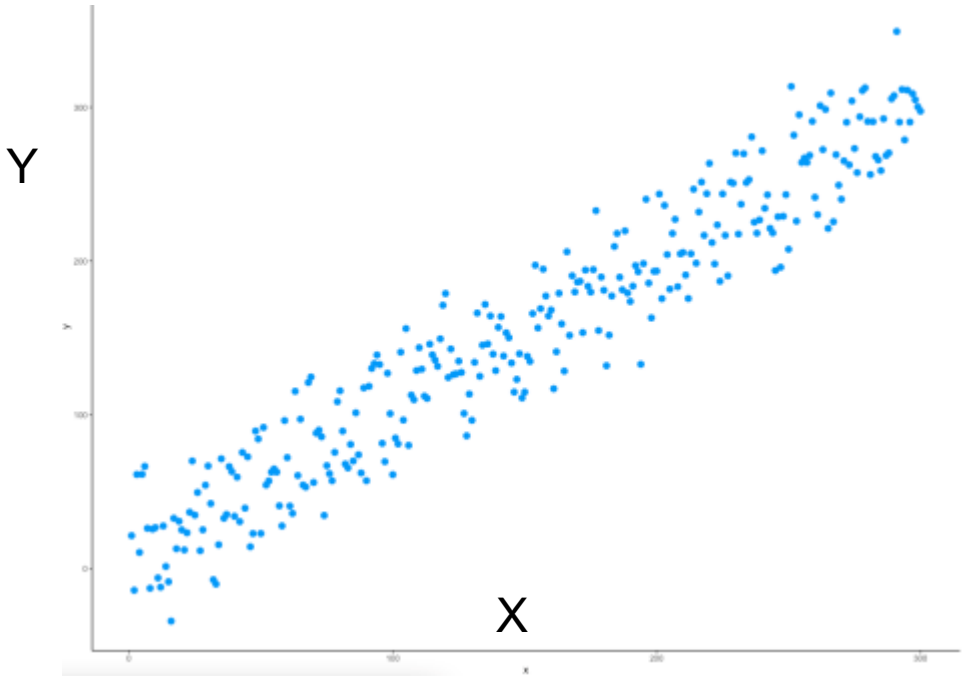
- Regression
- Classification

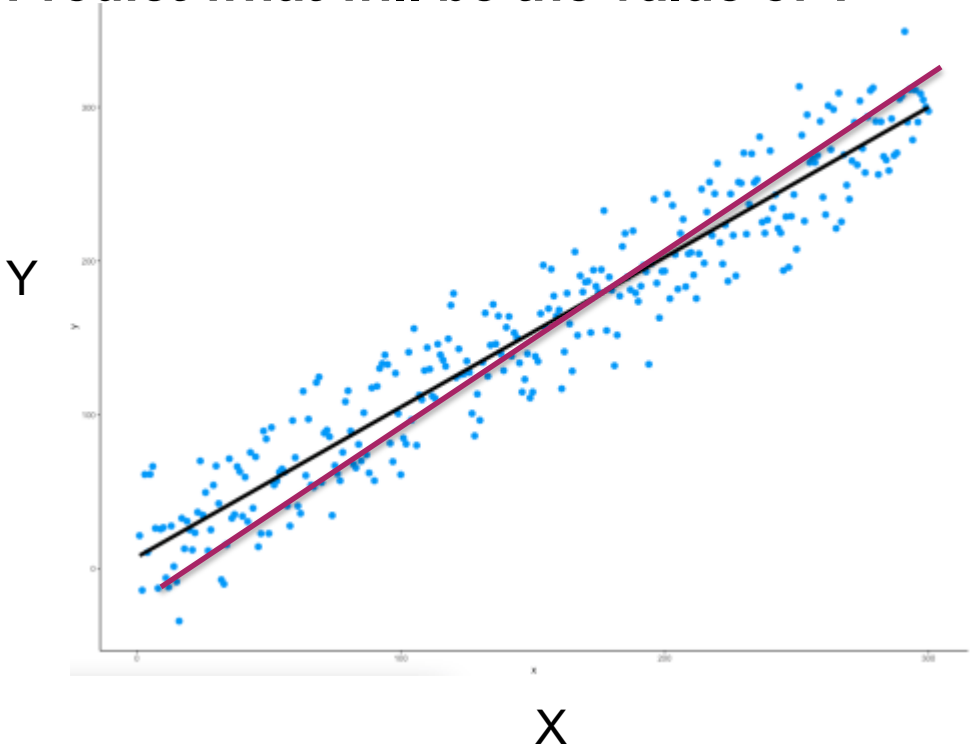
Unsupervised Learning

- Clustering
- Dimensionality Reduction

Linear Regression

Intuition: What happens to Y when X Changes?--- **Predict what will be the value of Y**

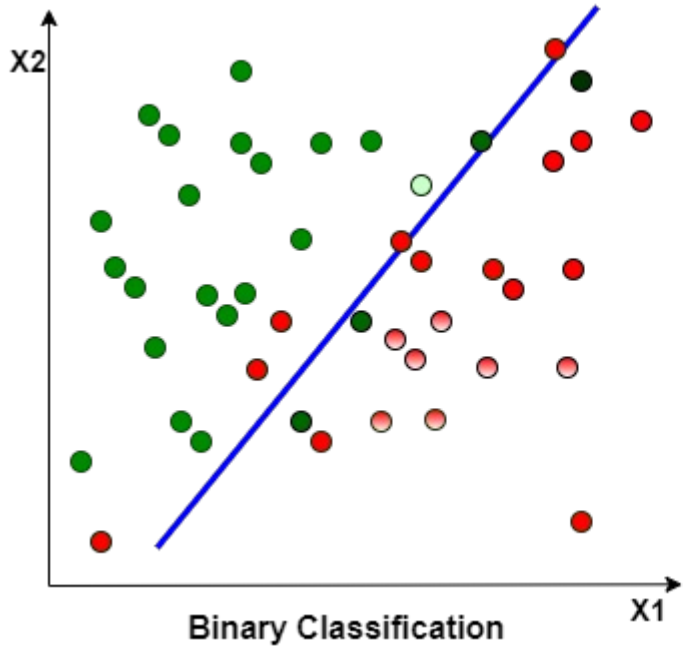




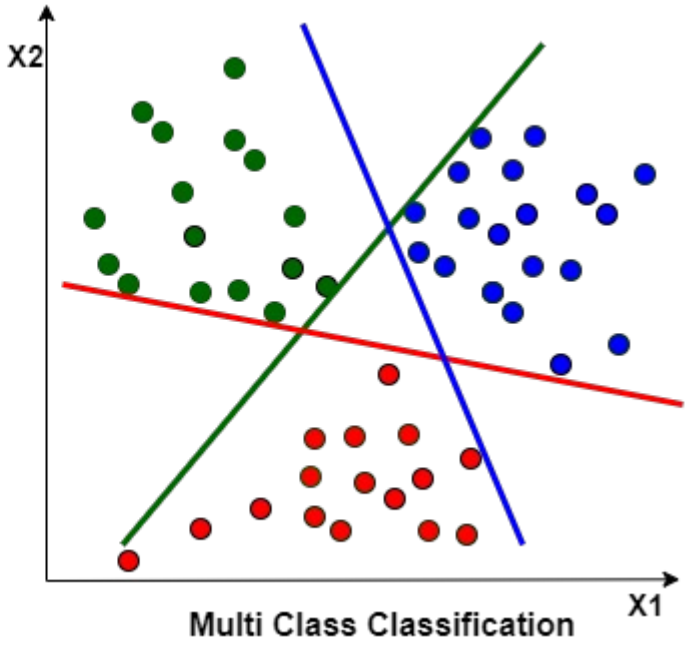
$$\hat{Y} = WX + b$$

Classification

Intuition: Categorize the data into various categorizes.—**Predict what will be the category of Y (either X1 or X2)**



Binary Classification



Multi Class Classification

Examples:

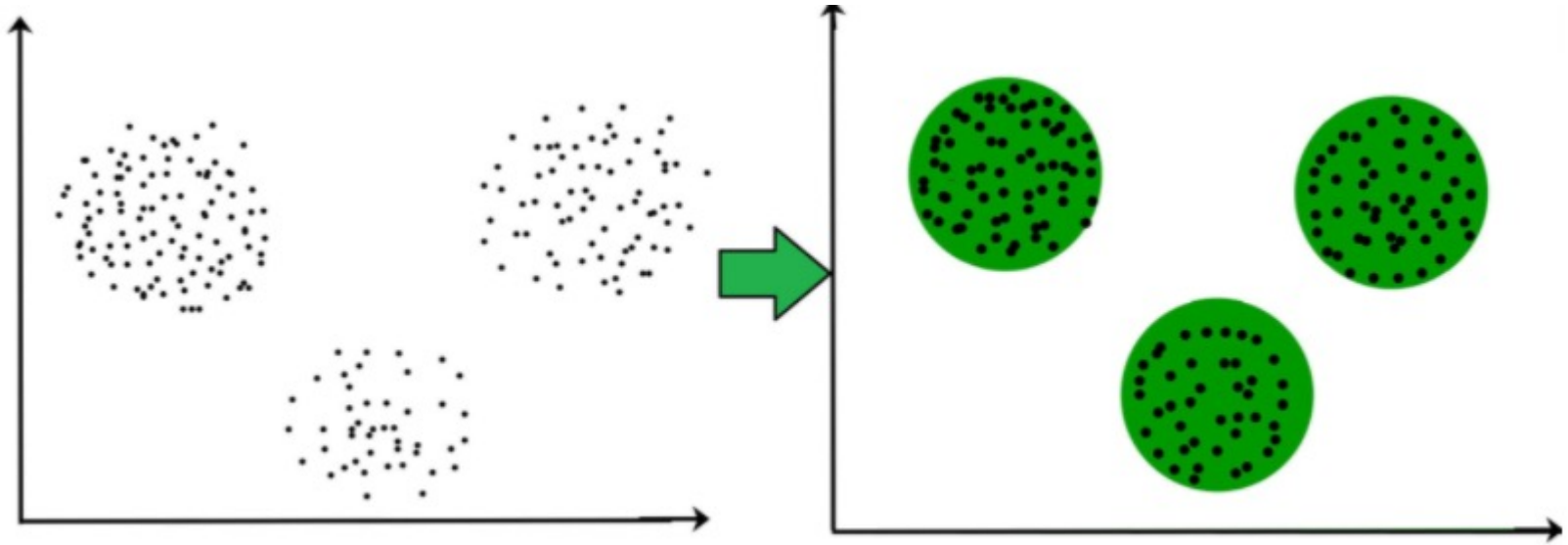
Image- and object-recognition: Supervised learning algorithms can be used to locate, isolate, and categorize objects out of videos or images, making them useful when applied to various computer vision techniques and imagery analysis.

Customer sentiment analysis: Using supervised machine learning algorithms, organizations can extract and classify important pieces of information from large volumes of data—including context, emotion, and intent—with very little human intervention.

Spam detection: Spam detection is another example of a supervised learning model. Using supervised classification algorithms, organizations can train databases to recognize patterns or anomalies in new data to organize spam and non-spam-related correspondences effectively.

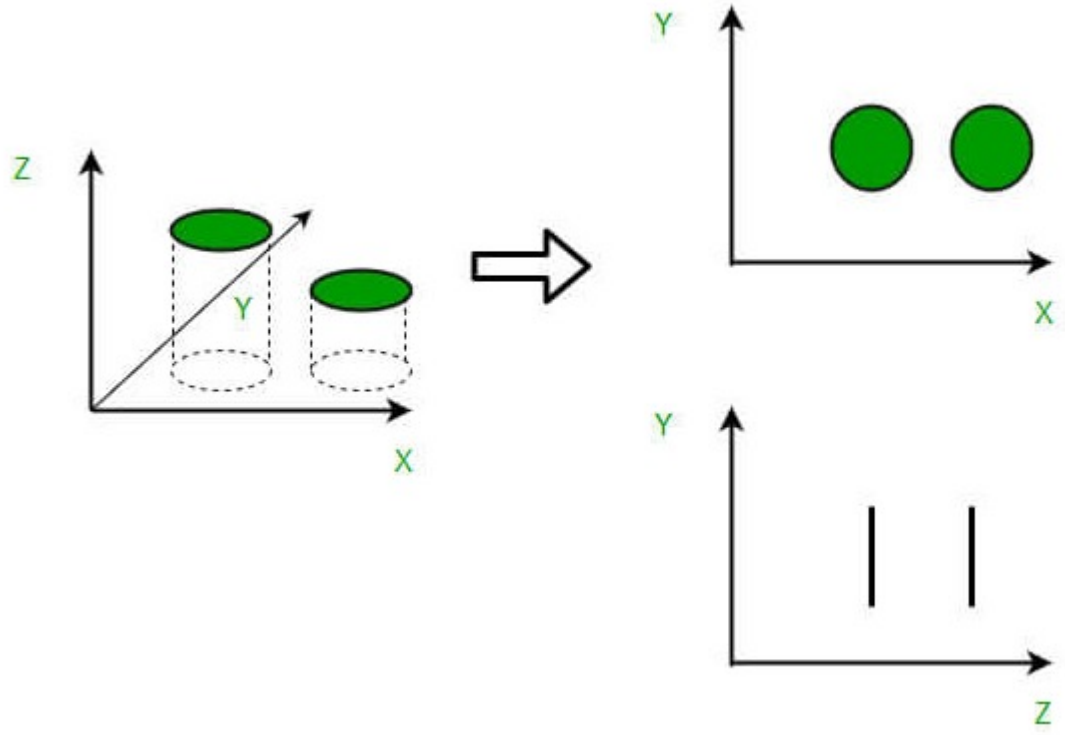
Unsupervised Learning

Clustering: When we do not have any pre-defined data, we can use Clustering Algorithms to cluster the data based on a particular property: Such as Distance



Unsupervised Learning

Dimensionality Reduction: When we have lot of data, we can select only those which are relevant to address our problems.



This much data can tell you everything.. You dont need all data

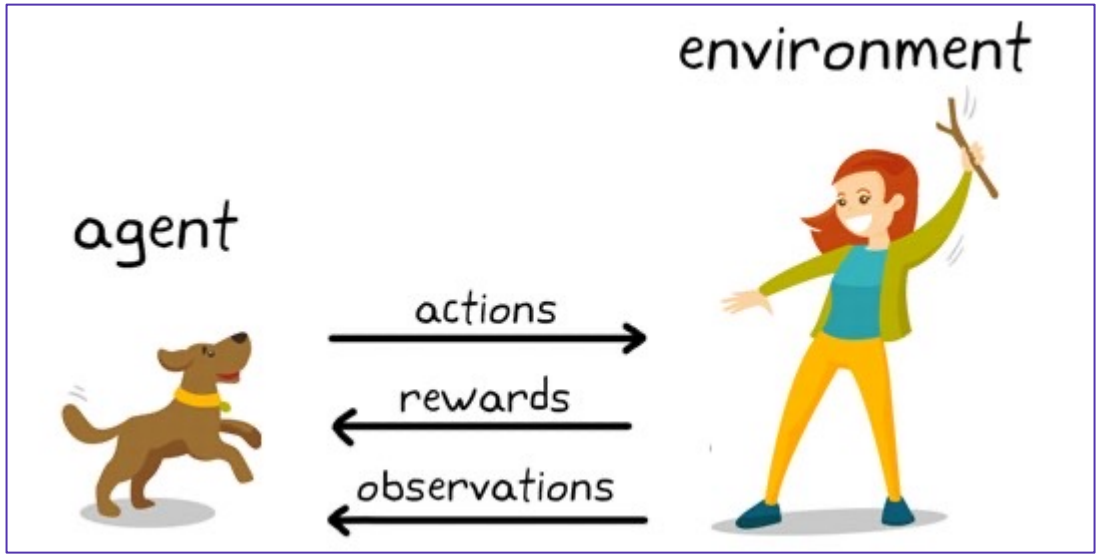
Examples:

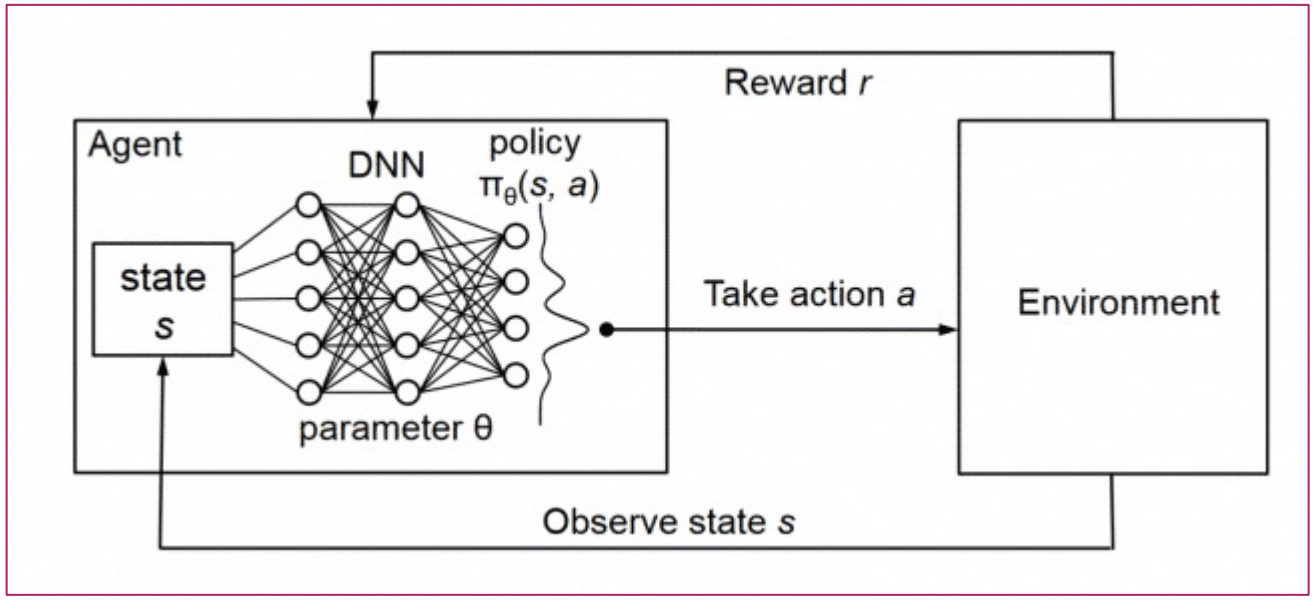
Segment my end users based on their buying preferences/languages/….

Anomaly Detection: Detect which activities are from Bots and which are not

Fraud Detection: Detect which activities are fraudulent..

Reinforcement Learning





Summary

- Given the historical data of power consumption in Europe can you predict the power consumption for 2021?

Prediction problem, supervised learning,

- Given a dataset of housing prices in Munich can you estimate the price of my house?

Regression problem, supervised learning

- Can we automatically detect the tone of Elon Musk tweets?

Multi-label Classification, supervised

Summary

- Can we detect roads/buildings from satellite images?

Multi class Classification, supervised/semi-supervised learning

- We have various sensors in our factory can we automatically detect if anything goes wrong?

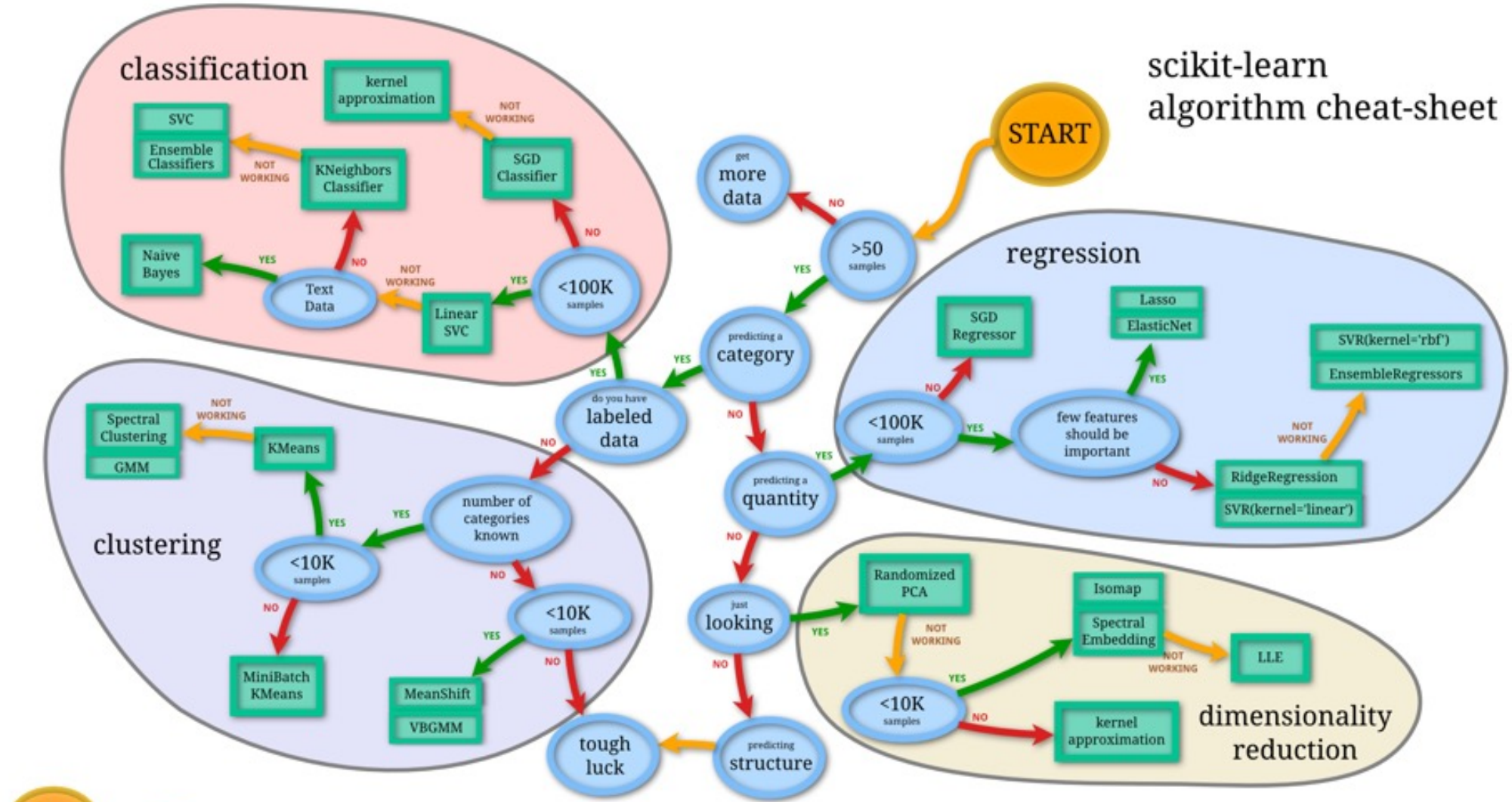
Anomaly detection, supervised/unsupervised learning

- We have a dataset of the users of our service. Can we segment them into a few groups to offer more personalized service to similar groups?

Clustering, Unsupervised Learning

- Can we train our robot to reach from A to B in the least amount of time while avoiding the obstacles?

Reinforcement Learning


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