Introduction to Focus Areas in Bioinformatics (19405152)

Lecturer: Priyanka Banerjee (Charite)

Session 4: Big Query (Genomics)

Slides acknowledgment : Tim Conrad, FU Course **(19405152)** WS20/21

- BigQuery
- Speeding up Computations: SPARK
- Advanced Table Queries: SQL
- Problematic Datasets:
  non-complete and non-balanced data
- How can this be used in practice: Project 4

- 10:15 – 12:00 Introduction

- Break -

- 12:30 – 13:45 Discussion, Presentations & Project Work

Reports

Remember:

If you go over the time-limit, you can stop working on the project. However, the report must be written and – in that case – contain a discussion, why you hit the time limit. (Missing programming experience does not count.)

If you fail a report, you can resubmit once within a week of notification.

For the resubmission: you need to fix those things that I mentioned in the feedback email.

If everything is fixed send me an email with the new version.

PREVIOUS PROJECTS

RECAP:

The main idea so far was to practically do analysis of life-science data.

Project Goals:

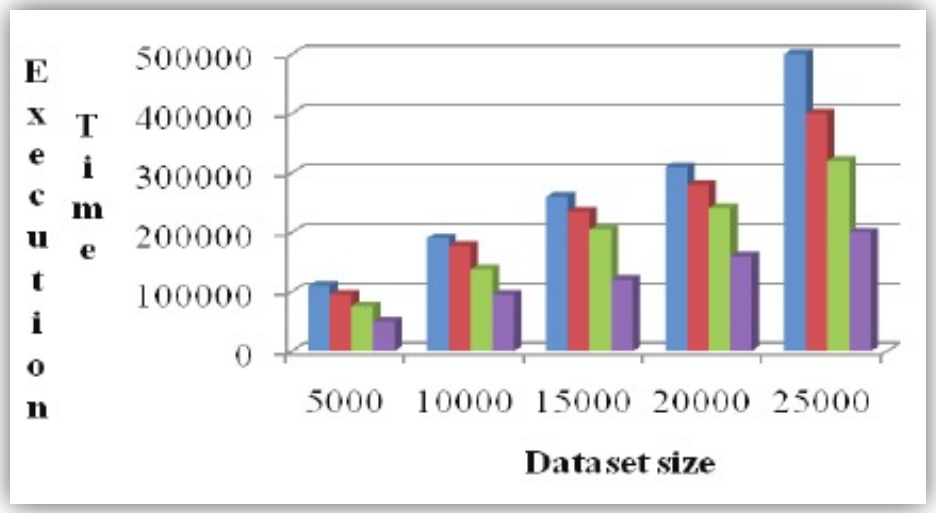
1. Develop a classifier to diagnose a heart disease - using tabular data.
2. Develop, **evaluate and analyze** a classifier to diagnose a biopsy - using tabular data extracted from an image.
3. Develop, evaluate and analyze a classifier to diagnose a biopsy - **using the raw image.**

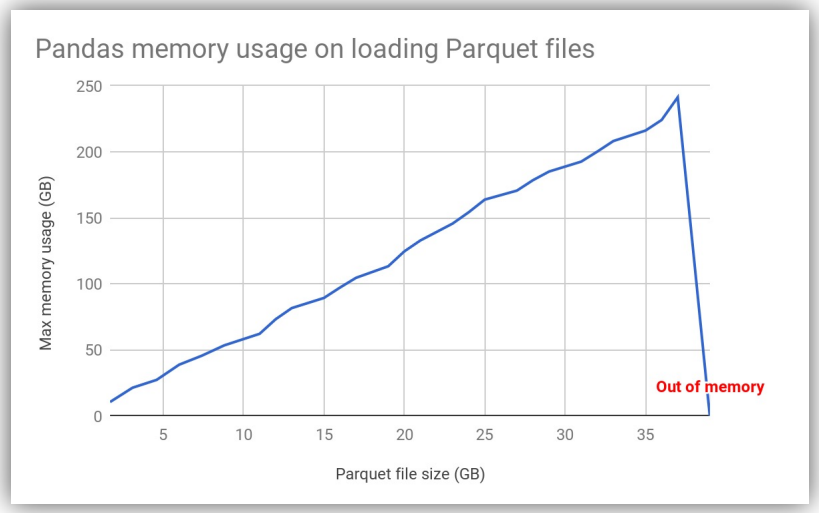
Realistic?

- Although the data and approaches taken were quite realistic already, one thing was certainly not:

Realistic?

- Although the data and approaches taken were quite realistic already, one thing was certainly not: **the size of the date-set**.
- This influences the runtime and the available tools.





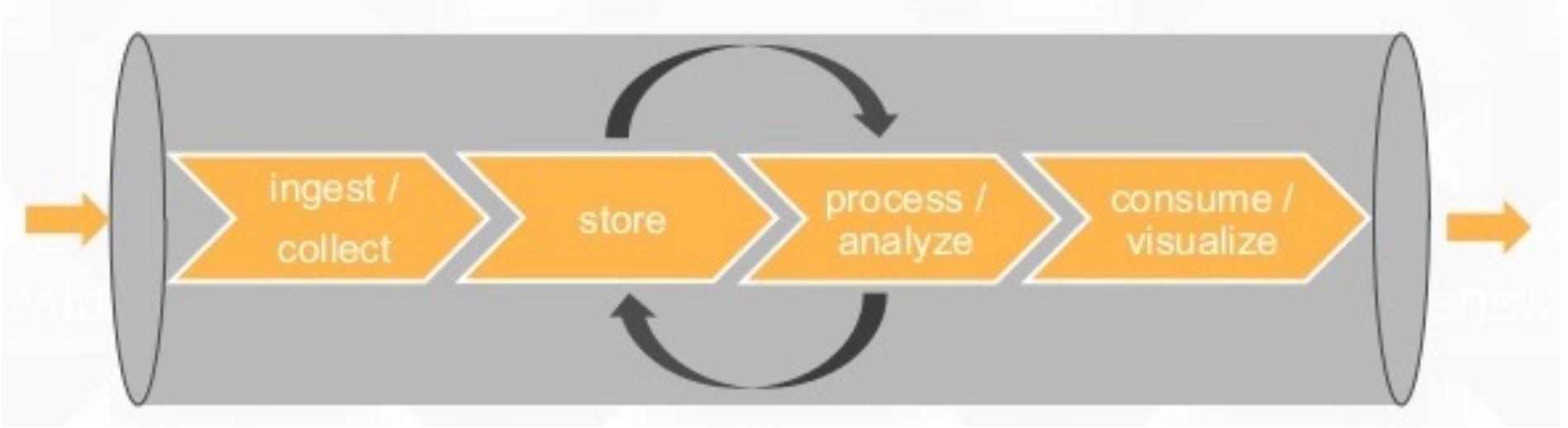
Bars (colors) represent different algorithms.)

PROCESSING LARGE DATA E.G. FOR MACHINE LEARNING

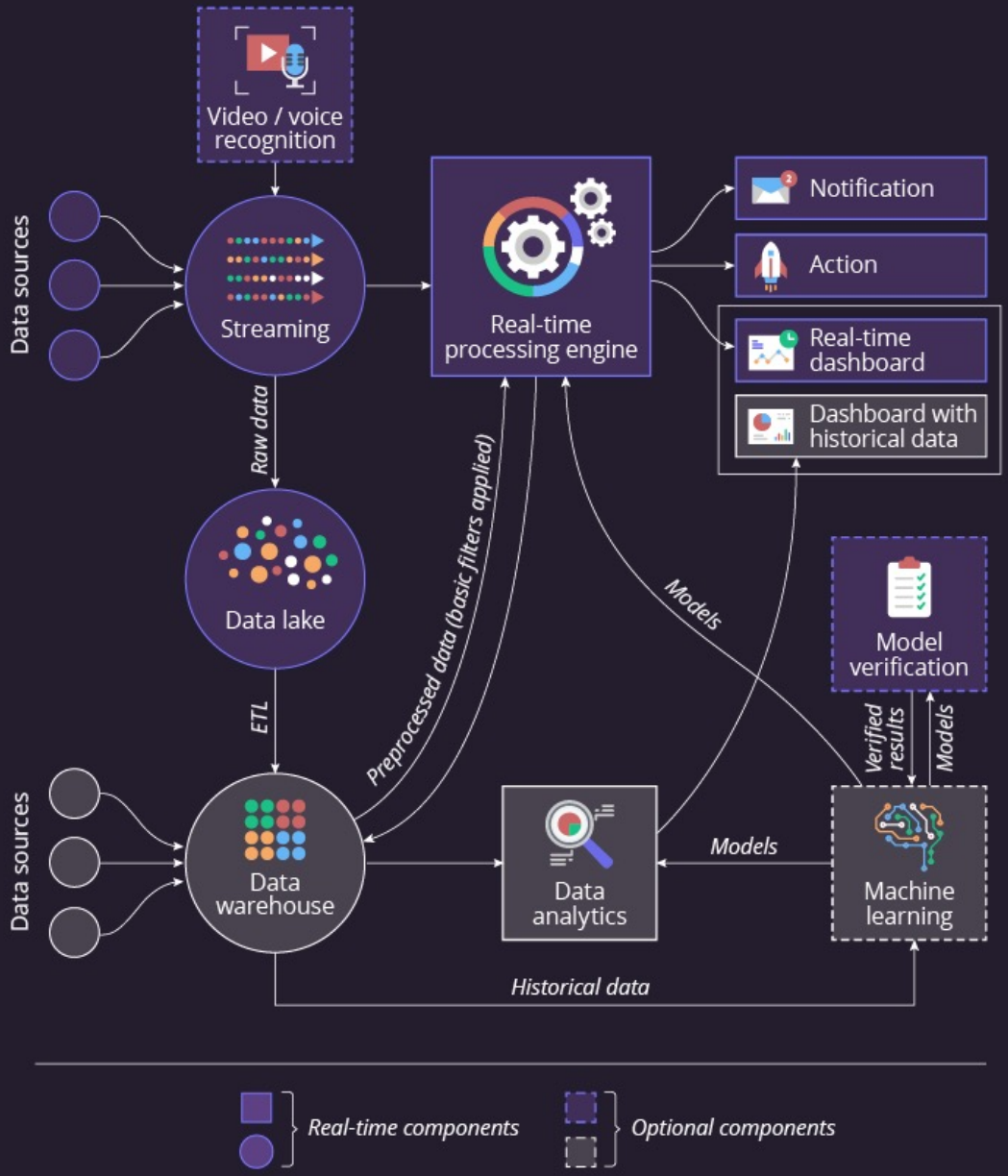
Reality?

- Although the data and approaches taken were quite realistic already, in real-world applications it looks a bit different.

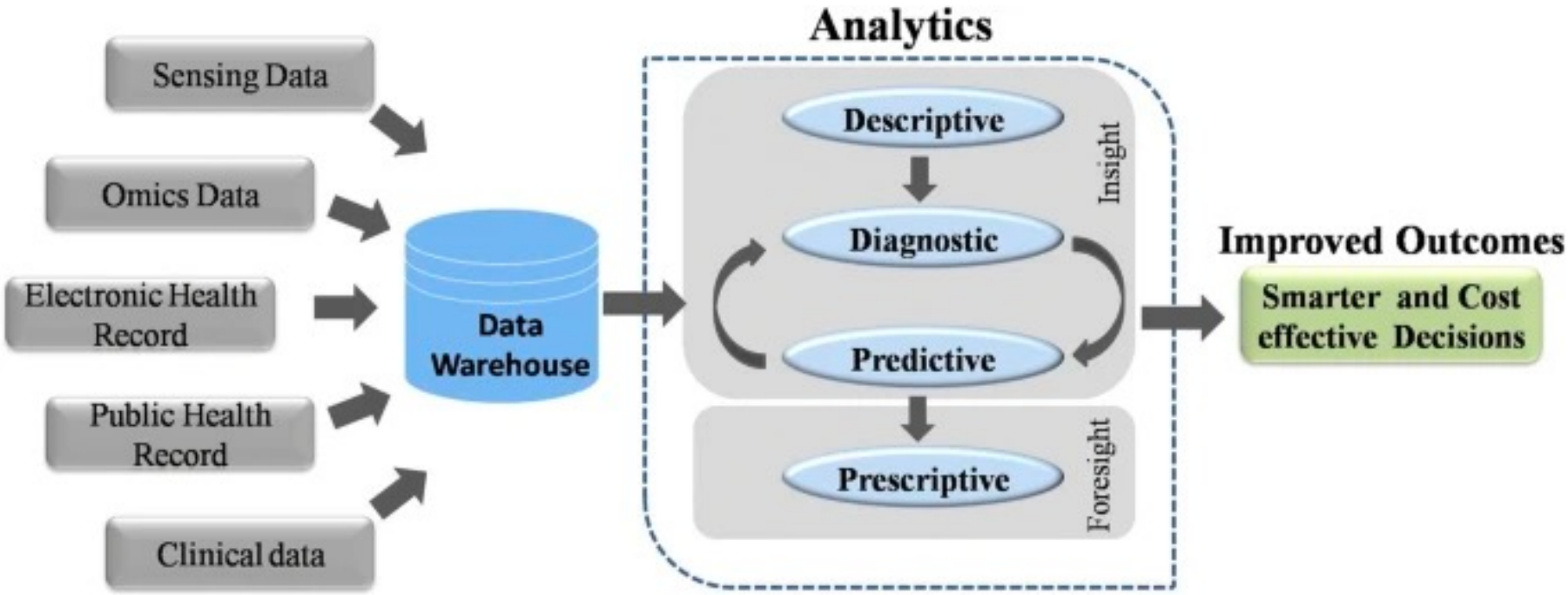
- One common workflow is:



REAL-TIME BIG DATA ANALYTICS ARCHITECTURE



Big data in healthcare: management, analysis and future prospects

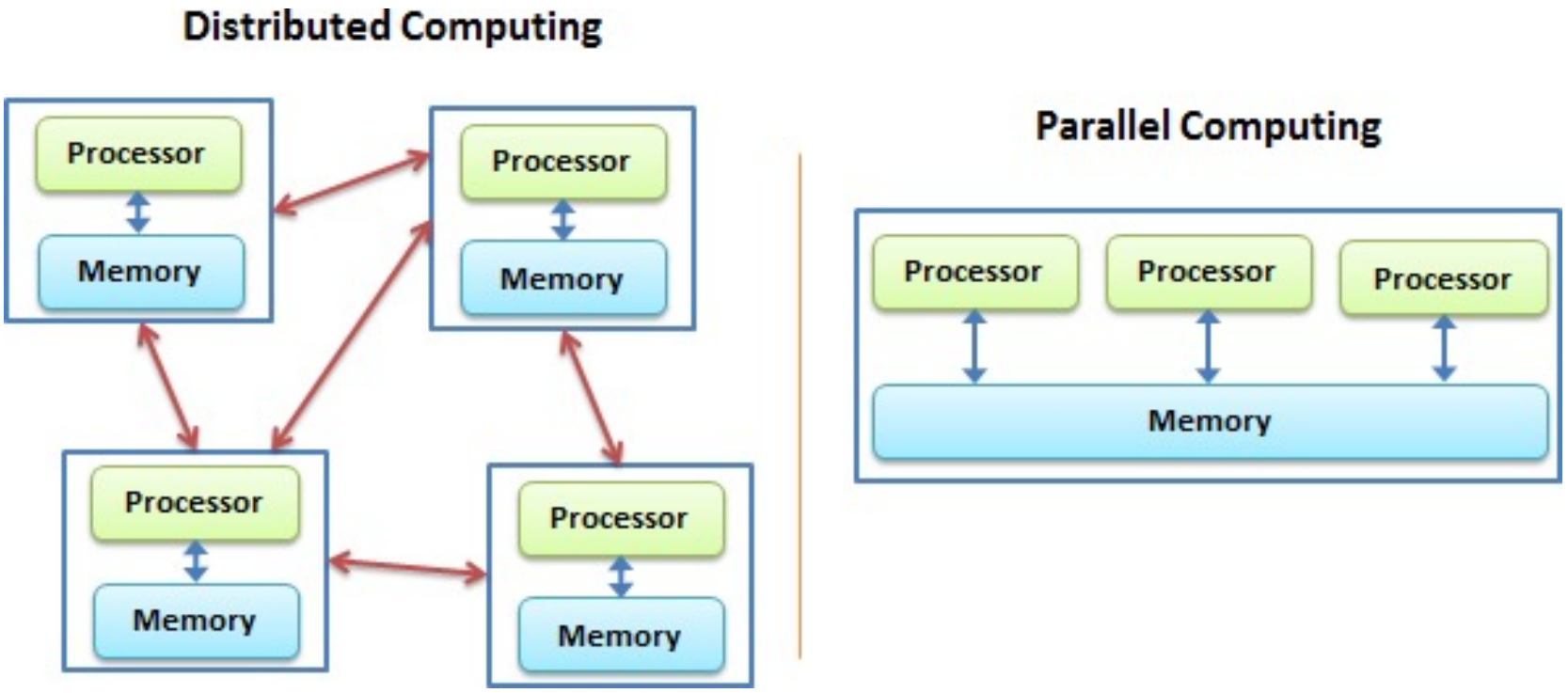


https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0217-0

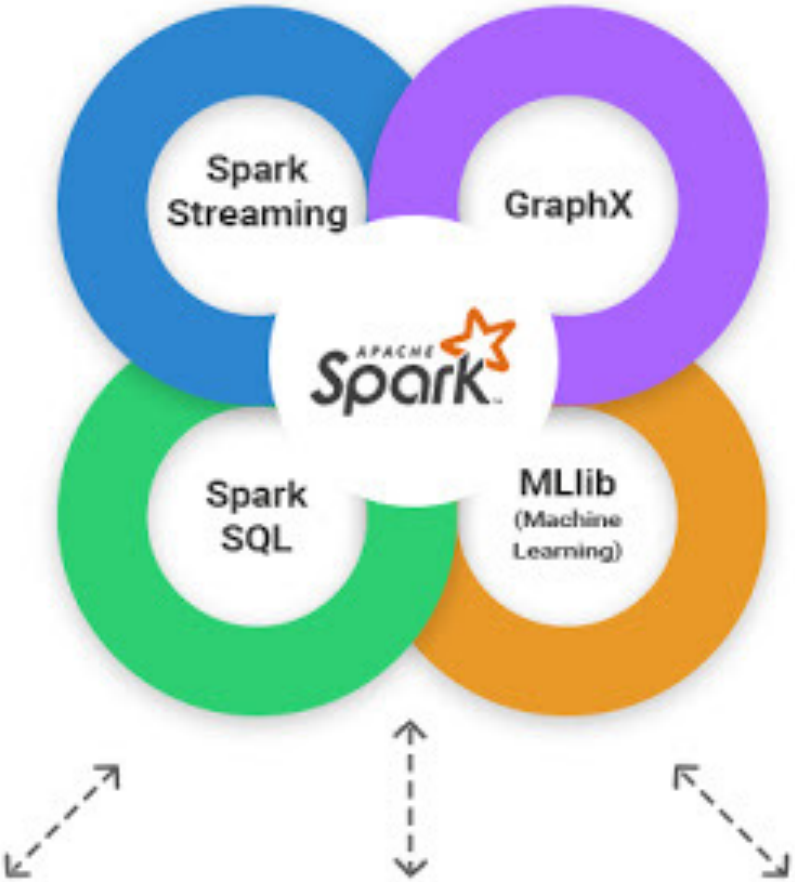
So what's the main difference?

- Work is distributed and organized in workflows.
- Parts of the workflow run on different machines for load balancing.
- Same part (or: task, e.g. data cleaning) can happen on many machines in parallel.
- **BIG CHALLENGE: how can the data be distributed efficiently such that all "parts" can access it?**

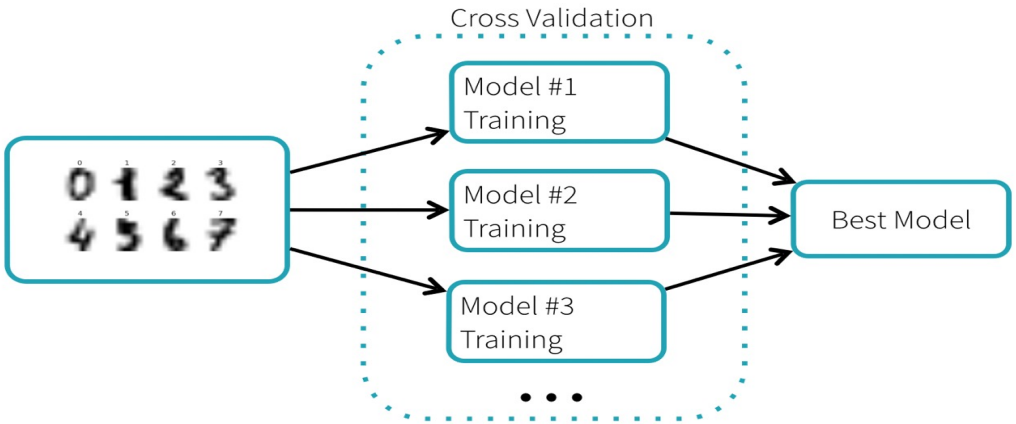
Goal: Use more compute power

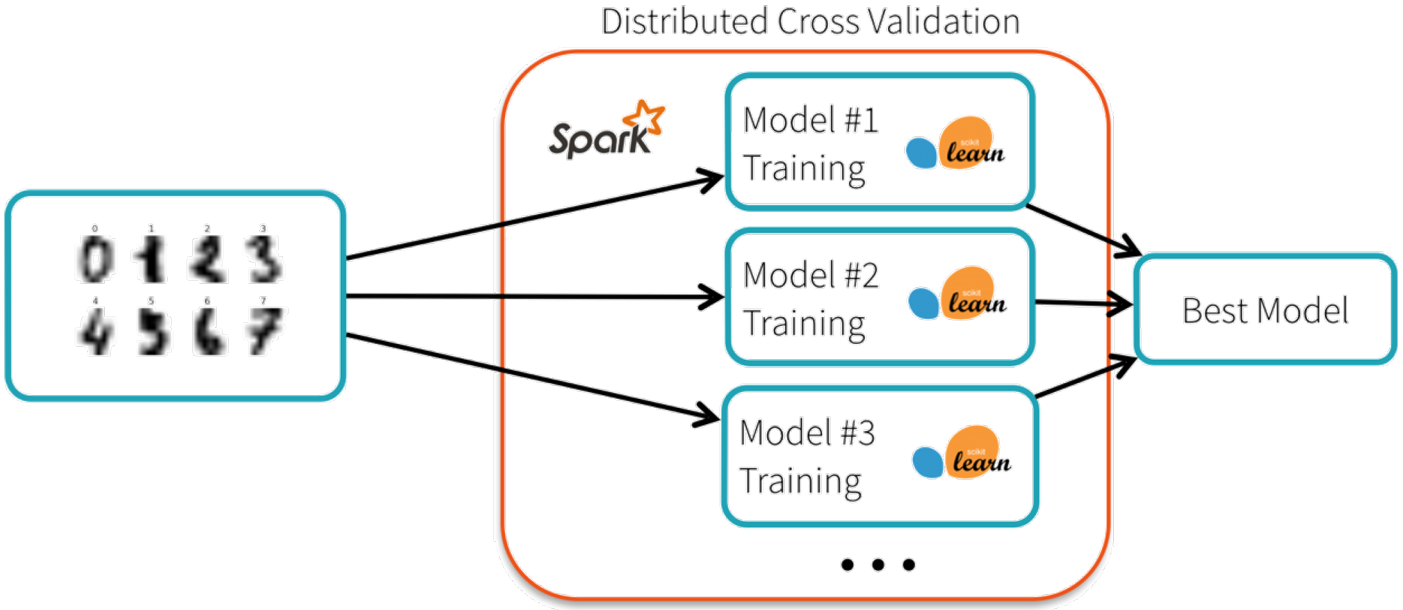



DATA & AI LANDSCAPE 2019
FRAMEWORKS
Spark
Flink
YARN
TEZ
MESOS
kubernetes
docker
CDAP
Red Hat
HELIX
QUERY / DATA FLOW
Spark SQL
presto
APACHE DRILL
SLAMDATA
GraphQL
Flink
DATA ACCESS & DATABASES
cassandra
mongoDB
redis
Cockroach LABS
druid
CouchDB
OPENTSDB
SciDB
riak
HBASE
Cloud Spanner
ACCUMULO
ORCHESTRATION & MGMT
talend
Apache Zookeeper
Apache Ambari
Apache Airflow
MESOS
etcd
Kong
STREAMING & MESSAGING
Spark
nifi
Flink
beam
kafka
STORM
Apache RocketMQ

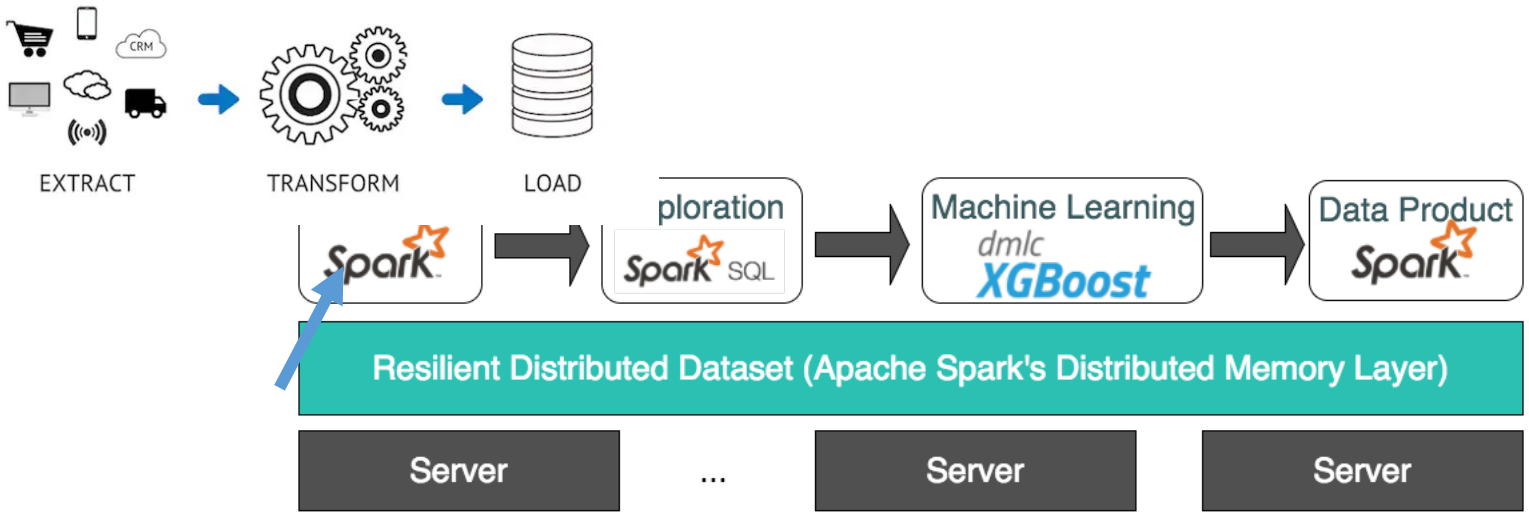

Spark Streaming
GraphX
APACHE Spark
Spark SQL
MLlib (Machine Learning)
DATA SOURCE API
PostgreSQL
MySQL
cassandra
hadoop
APACHE HBASE

ML with Spark



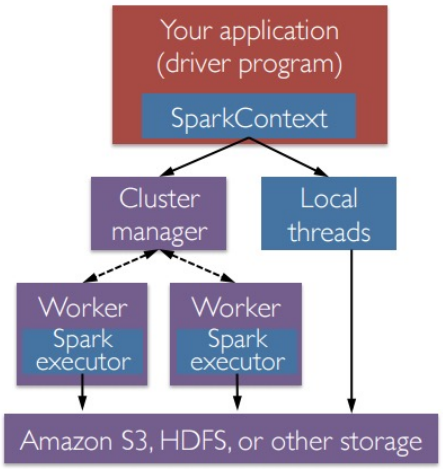


Data Science with Spark



Spark Components

- A Spark program first creates a SparkContext object
  - Tells Spark how and where to access a cluster
  - Connect to several types of cluster managers (e.g., YARN or its own manager)
- Cluster manager:
  - Allocate resources across applications
- Spark executor:
  - Run computations
  - Access data storage



Data Distribution

- Remember the BIG CHALLENGE from earlier:

  How can the data be distributed efficiently such that all "parts" can access it?

- Spark introduced a concept called "Resilient Distributed Datasets" (RDDs).

- "RDD is the primary data abstraction in Apache Spark and the core of Spark. It enables operations on collection of elements in parallel."

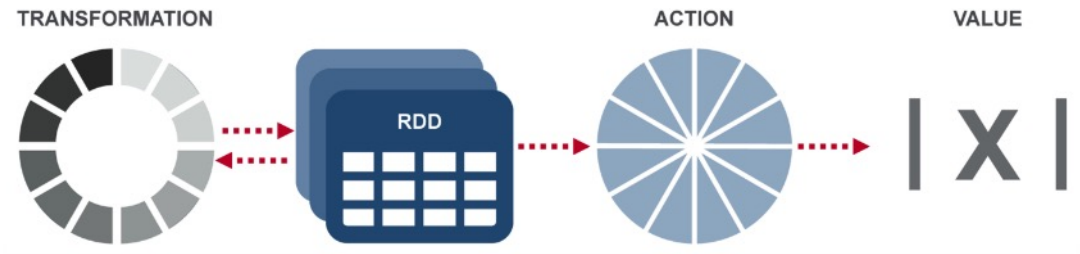
Resilient Distributed Datasets

- *Resilient Distributed Datasets (RDDs)*
  - Distributed collections of objects that can be cached in memory across a compute cluster
  - Manipulated through parallel operators
  - Automatically recomputed on failure
- RDDs can express many parallel algorithms, and capture many current programming models
  - Data flow models: MapReduce, SQL, …
  - Specialized models for iterative apps: Pregel, …

What is RDD

- Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. (Zaharia, et al. NSDI'12)
  - RDD is a **distributed** memory abstraction that lets programmers perform **in-memory** computations on large clusters in a **fault-tolerant** manner.
- **Resilient**
  - Fault-tolerant, is able to recompute missing or damaged partitions due to node failures.
- **Distributed**
  - Data residing on multiple nodes in a cluster.
- **Dataset**
  - A collection of partitioned elements, e.g. tuples or other objects (that represent records of the data you work with).

RDD Operations



- **Transformation:** returns a new RDD.
  - Nothing gets evaluated when you call a Transformation function, it just takes an RDD and return a new RDD.
  - Transformation functions include *map, filter, flatMap, groupByKey, reduceByKey, aggregateByKey, filter, join, etc.*
- **Action:** evaluates and returns a new value.
  - When an Action function is called on a RDD object, all the data processing queries are computed at that time and the result value is returned.
  - Action operations include *reduce, collect, count, first, take, countByKey, foreach, saveAsTextFile, etc.*

Spark Transformations

- Create new datasets from an existing one
- Use lazy evaluation: results not computed right away – instead Spark remembers set of transformations applied to base dataset
  - Spark optimizes the required calculations
  - Spark recovers from failures
- Some transformation functions

Transformation	Description
`map(`*`func`*`)`	return a new distributed dataset formed by passing each element of the source through a function *func*
`filter(`*`func`*`)`	return a new dataset formed by selecting those elements of the source on which *func* returns true
`distinct([`*`numTasks`*`]))`	return a new dataset that contains the distinct elements of the source dataset
`flatMap(`*`func`*`)`	similar to map, but each input item can be mapped to 0 or more output items (so *func* should return a Seq rather than a single item)

More Information

A nice introduction to ***Apache Spark in Python*** can be found here:

https://towardsdatascience.com/a-neanderthals-guide-to-apache-spark-in-python-9ef1f156d427

DATA EXPLORATION USING SQL (FOR SPARK)

Based on slides by Tova Milo

Working with Data in Python

- Pandas is an excellent Python data analysis library which provides high-performance, easy-to-use data structures and data analysis tools.
- However, when it comes to so-called BIG DATA, Pandas has its limitations and this is where other approaches are needed.
- We will have a look at Apache Spark.

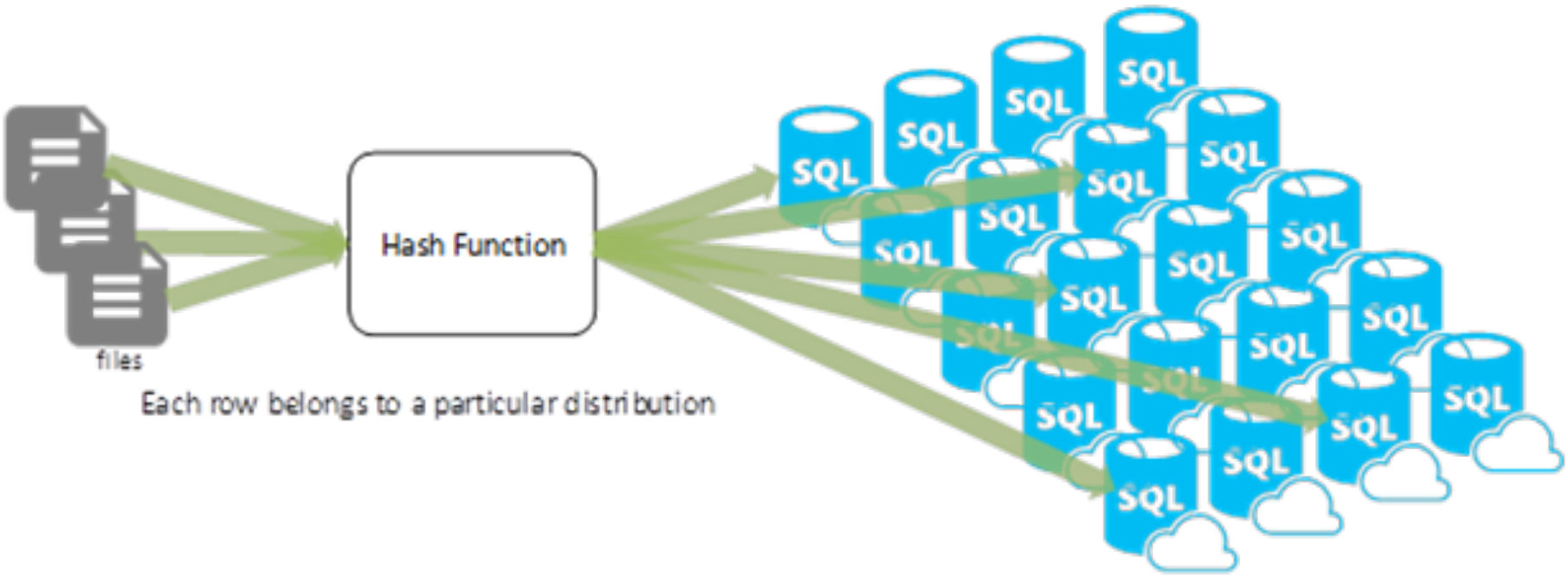
Distributed Data & Queries

- Imagine, you have some REALLY BIG files. (Think of Terabytes.)
- The data in these files is structured. (Think of Excel tables.)
- You want to search these files and perform some very interesting analyses.
- If you would do this on a single computer, just going through these files over and over again is just too time consuming.

Distributed Data & Queries

A common strategy is to split the data to many computers.

But then: how to query the now distributed data efficiently?



Distributed Data & Queries

A common strategy is to split the data to many computers.

But then: how to query the now distributed data efficiently?

Spark and it's extension "Spark SQL" are designed for this task.

Distributed Data & Queries

https://www.youtube.com/watch?v=27axs9dO7AE

SQL Introduction

Standard language for querying and manipulating data

Structured **Q**uery **L**anguage

Many standards out there:

- ANSI SQL
- SQL92 (a.k.a. SQL2)
- SQL99 (a.k.a. SQL3)
- Vendors support various subsets of these
- What we discuss is common to all of them

SQL

- Data Definition Language (DDL)
  - Create/alter/delete tables and their attributes
- Data Manipulation Language (DML)
  - Query one or more tables – discussed next !
  - Insert/delete/modify tuples in tables
- Transact-SQL
  - Idea: package a sequence of SQL statements - server

Data in SQL

1. Atomic types, a.k.a. data types
2. Tables built from atomic types

Unlike XML, no nested tables, only flat tables are allowed!

- We will see later how to decompose complex structures into multiple flat tables

Data Types in SQL

- Characters:
  - CHAR(20) -- fixed length
  - VARCHAR(40) -- variable length
- Numbers:
  - BIGINT, INT, SMALLINT, TINYINT
  - REAL, FLOAT -- differ in precision
  - MONEY
- Times and dates:
  - DATE
  - DATETIME -- SQL Server
- Others... All are simple

Tables in SQL

Table name

Attribute names

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Tuples or rows

Tables Explained

- A tuple = a record
  - Restriction: all attributes are of atomic type
- A table = a set of tuples
  - Like a list…
  - …but it is unordered: no **first()**, no **next()**, no **last()**.

Tables Explained

- The *schema* of a table is the table name and its attributes:

Product(PName, Price, Category, Manfacturer)

- A *key* is an attribute whose values are unique; we underline a key

Product(PName, Price, Category, Manfacturer)

Simple Queries in SQL

SQL Query

Basic form: (plus many many more bells and whistles)

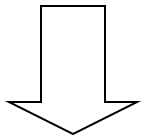
```
SELECT  attributes
FROM    relations (possibly multiple)
WHERE   conditions (selections)
```

Simple SQL Query

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

```
SELECT  *
FROM    Product
WHERE   category='Gadgets'
```



PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks

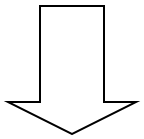
"selection"

Simple SQL Query

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

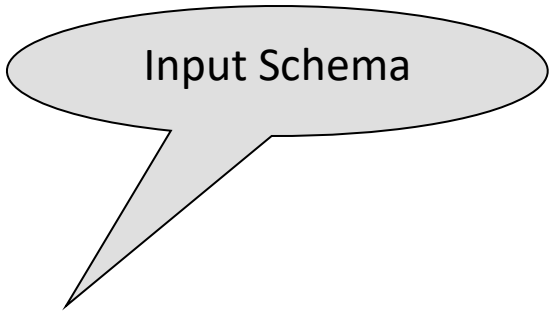
```
SELECT  PName, Price, Manufacturer
FROM    Product
WHERE   Price > 100
```



"selection" and "projection"

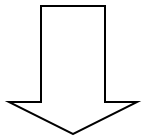
PName	Price	Manufacturer
SingleTouch	$149.99	Canon
MultiTouch	$203.99	Hitachi

A Notation for SQL Queries

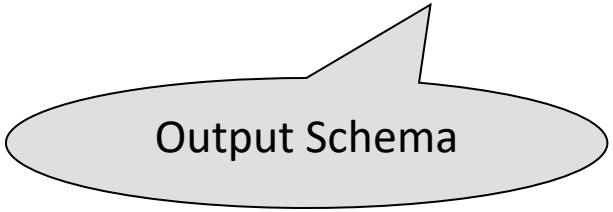


Product(PName, Price, Category, Manfacturer)

```
SELECT   PName, Price, Manufacturer
FROM     Product
WHERE    Price > 100
```



Answer(PName, Price, Manfacturer)



Selections

What goes in the WHERE clause:

- x = y, x < y, x <= y, etc
  - For number, they have the usual meanings
  - For CHAR and VARCHAR: lexicographic ordering
    - Expected conversion between CHAR and VARCHAR
  - For dates and times, what you expect...
- Pattern matching on strings...

The **LIKE** operator

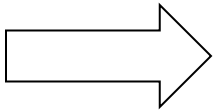
- s **LIKE** p: pattern matching on strings
- p may contain two special symbols:
  - % = any sequence of characters
  - _ = any single character

Product(PName, Price, Category, Manufacturer)
Find all products whose name mentions 'gizmo':

```
SELECT  *
FROM    Products
WHERE   PName LIKE '%gizmo%'
```

Eliminating Duplicates

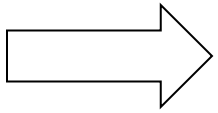
```
SELECT   DISTINCT category
FROM     Product
```



Category
Gadgets
Photography
Household

Compare to:

```
SELECT   category
FROM     Product
```



Category
Gadgets
Gadgets
Photography
Household

Ordering the Results

```
SELECT   pname, price, manufacturer
FROM     Product
WHERE    category='gizmo' AND price > 50
ORDER BY  price, pname
```

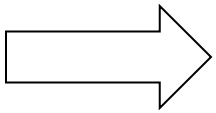
Ordering is ascending, unless you specify the DESC keyword.

Ties are broken by the second attribute on the ORDER BY list, etc.

Ordering the Results

```
SELECT   category
FROM     Product
ORDER BY pname
```

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi



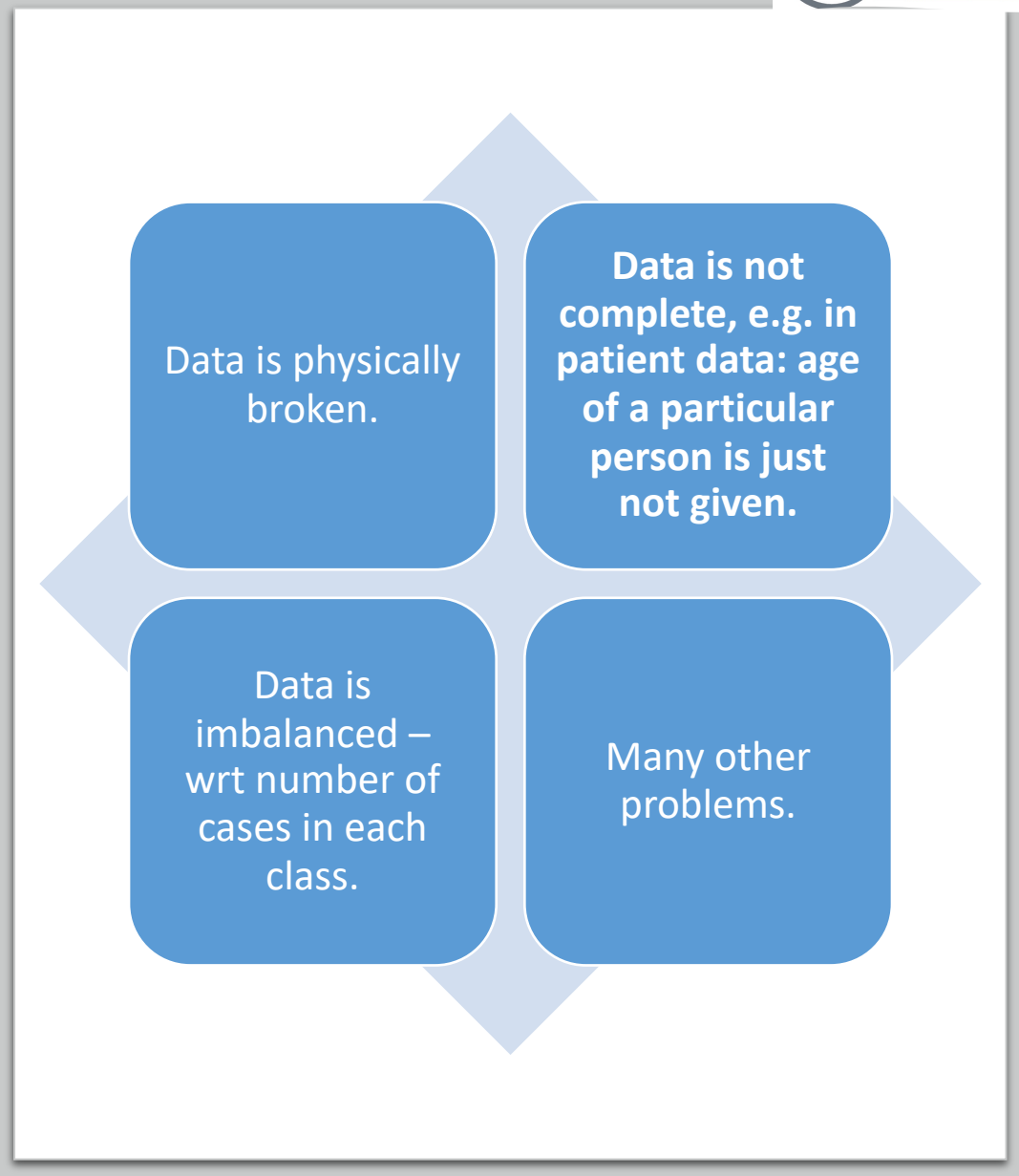
?

Advanced SQL

- **See e.g. “SQL for Web Nerds”** by Philip Greenspun
  http://philip.greenspun.com/sql/

ONCE AGAIN: PROBLEMS WITH THE DATA

Problems with the data



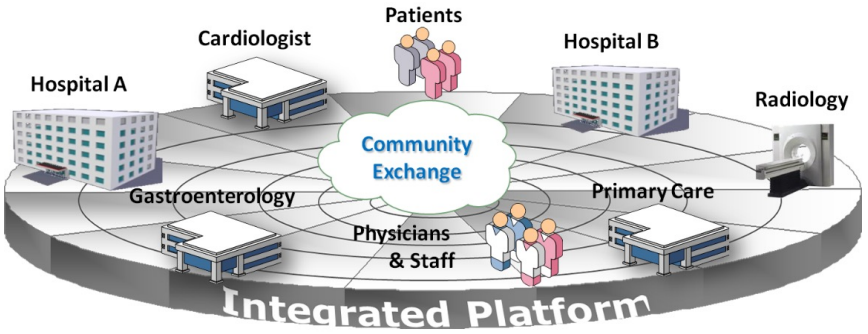
Detecting Missing Data

- Overtly missing data
  - Match data specifications against data - are all the attributes present?
  - Scan individual records - are there gaps?
  - Rough checks: number of files, file sizes, number of records, number of duplicates
  - Compare estimates (averages, frequencies, medians) with “expected” values and bounds.

Imputing Values to Missing Data

- In federated data, between 30%-70% of the data points will have at least one missing attribute - data wastage if we ignore all records with a missing value
- Remaining data is seriously biased
- Lack of confidence in results
- Understanding pattern of missing data unearths data integrity issues

Data federation is an aspect of **data** virtualization where the **data** stored in a heterogeneous set of autonomous **data** stores are made accessible to **data** consumers as one integrated **data** store by using on-demand **data** integration. Source: sciencedirect.com



Missing Value Imputation

- Simple approach: STANDALONE imputation
  - Use mean, median, other point estimates
  - Assume: Distribution of the missing values is the same as the non-missing values.
  - Does not take into account inter-relationships
  - Introduces bias
  - Convenient, easy to implement

Missing Value Imputation

More on this:

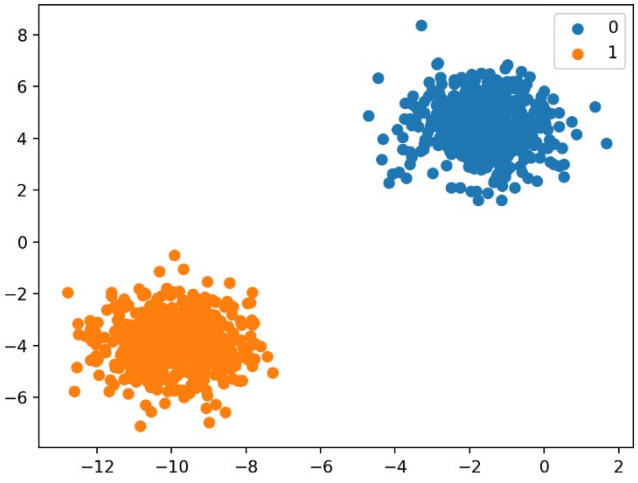
https://machinelearningmastery.com/handle-missing-data-python/

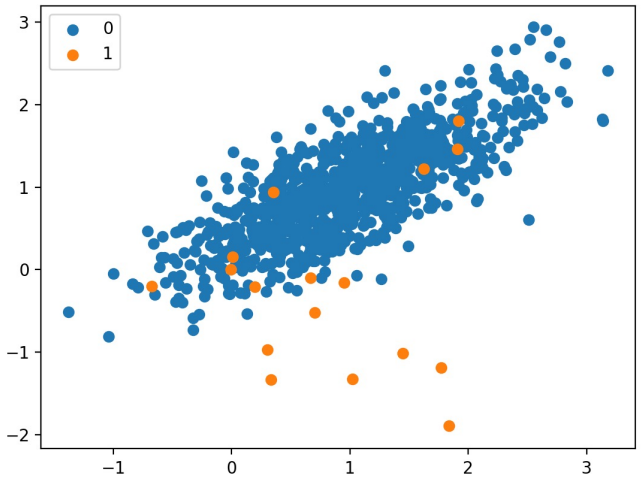
Problems with the data

- Data is physically broken.
- Data is not complete, e.g. in patient data: age of a particular person is just not given.
- **Data is imbalanced – wrt number of cases in each class.**
- Many other problems.

Imbalanced Classification

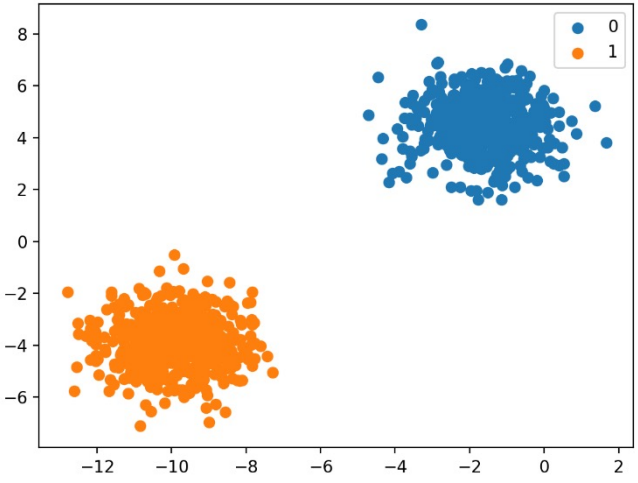
- Imbalanced classification refers to classification tasks where the number of examples in each class is unequally distributed.

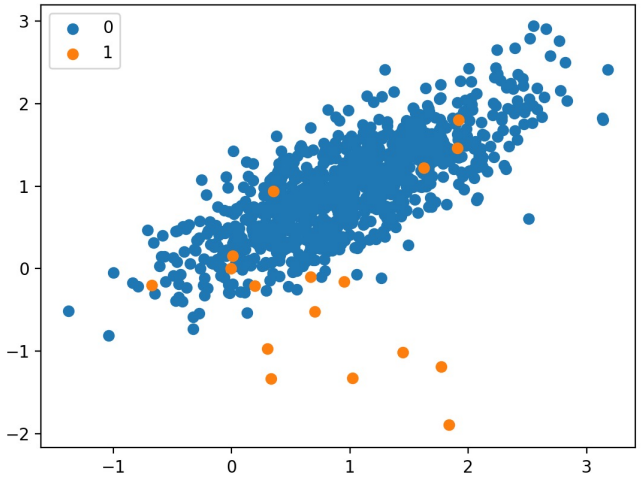




Imbalanced Classification

- Typical for binary classification tasks where the majority of examples in the training dataset belong to the normal class and a minority of examples belong to the abnormal class.

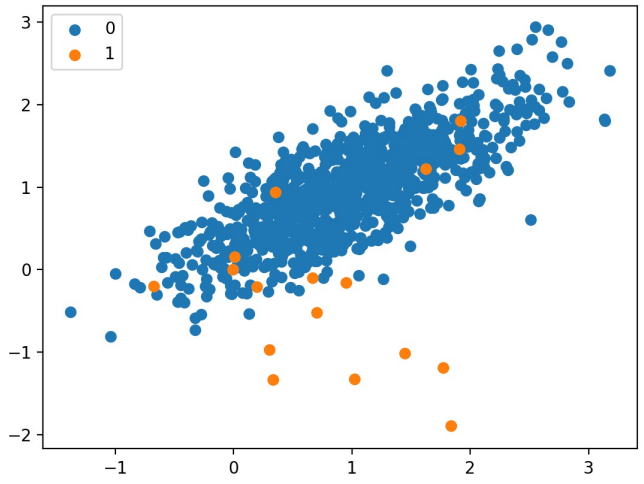




Imbalanced Classification

- Typical for binary classification tasks where the majority of examples in the training dataset belong to the normal class and a minority of examples belong to the abnormal class.

- Classifiers optimizing accuracy would predict only the blue class for any given input with very high accuracy.



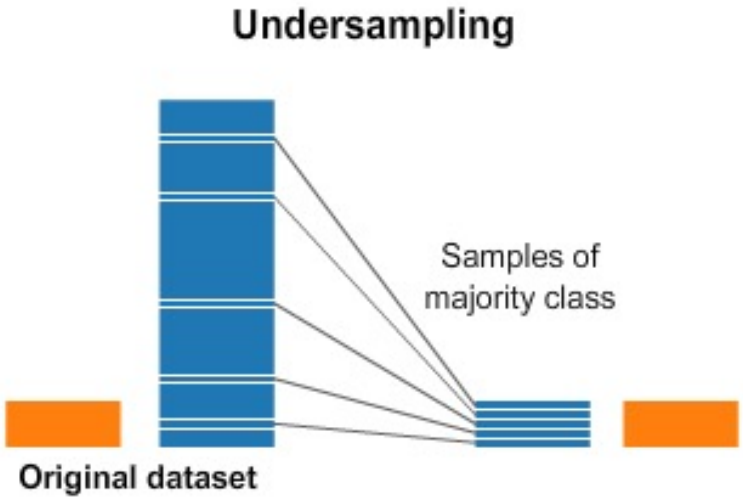
How is an imbalance presented in your dataset?

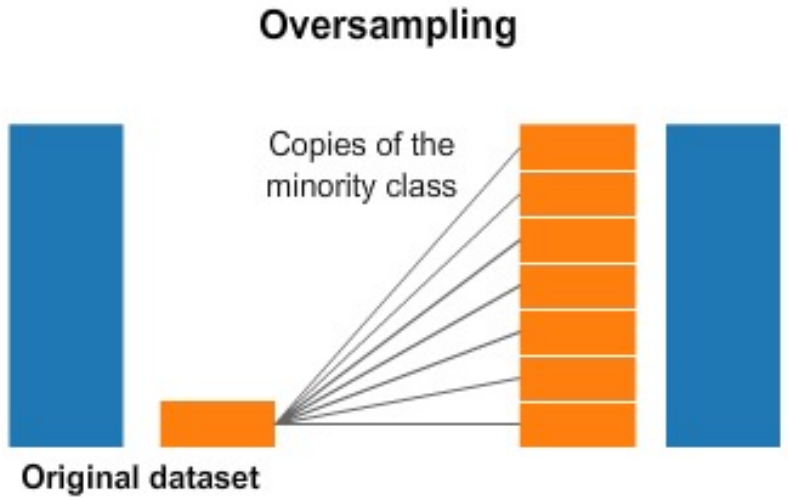
- **Between-class** and **within-class**: As expected in safety related events, it's expected that there are only few binary classes identifying "bad" occurrences. This imbalance also can be presented within classes in the case that some specific "bad" event is even more rare than the others.
- **Intrinsic** vs **extrinsic**: Intrinsic imbalance is due to the nature of the dataset, while extrinsic imbalance is related to time, storage and other factors that limit the dataset or the data analysis. Although we expect to face only intrinsic imbalance, we should not discard the occurrence extrinsic imbalance related problems.

- **Relative imbalance** vs **absolute rarity**: Sometimes the minority class may be outnumbered, however it is not necessarily rare, therefore this can be accurately learned with little disturbance. Note that, although the data present imbalance, it is not be something necessarily bad (and could even positive when using certain classifiers). It is very important to determine whether the imbalance is relative or if it is due to absolute rarity.
- **Small sample size imbalance:** Datasets with high dimensionality and small sample size are quite normal in actual data science problems (face recognition, gene expression, etc…) Limited datasets can also cause specific issues with certain machine learning algorithms.

Imbalanced Classification

- Strategy: change the composition of samples in the training dataset by over- or under-sampling.





Imbalanced Classification

More on this:

https://elitedatascience.com/imbalanced-classes

THIS WEEK'S PROJECT

About McKinsey Analytics Online Hackathon - Healthcare Analytics

Participate in a McKinsey Analytics Hackathon where you will have the opportunity to experience and overcome some of the challenges that leading global organizations face. The best participants will be short listed for interviews with us and additionally the winner will receive an all-expenses paid trip to an international analytics conference of your choice as a McKinsey guest, subject to visa approval and ticket availability.

About McKinsey & Company

McKinsey & Company is a global management consulting firm, deeply committed to helping institutions in the private, public and social sectors achieve lasting success. For over nine decades, our primary objective has been to serve as our clients' most trusted external advisor. With consultants in over 120 locations, in over 60 countries, across industries and functions, we bring unparalleled expertise to clients anywhere in the world. We work closely with teams at all levels of an organization to shape winning strategies, mobilize for change, build capabilities and drive successful execution.

McKinsey Analytics helps clients achieve better performance through data. We work together with clients to build analytics-driven organizations, providing end-to-end support covering strategy, operations, data science, implementation and change management. Our engagements range from use-case specific applications to full-scale analytics transformations. Our teams of consultants, data scientists, and engineers work together with clients to identify opportunities, assess available data, define solutions, establish optimal hosting environments, ingest data, develop cutting-edge algorithms, visualize outputs, and assess impact while building capabilities to sustain and expand it. Learn more at http://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights

Job Description - Data Scientist

Who You'll Work With

You'll work as part of the McKinsey Analytics team in one of our offices worldwide, for instance Waltham, MA, Chicago, New York, Sao Paolo, Madrid, London, Brussels, Dusseldorf, Milan, Wroclaw, Moscow, Gurgaon, Chennai, Bangalore, yet other locations are also available. You will work with data engineers and data translators to deliver the most cutting-edge analytics solutions to our Clients from various industries. Through the combination of strategic insights and advanced analytics technologies, you will be solving the most critical problems leading global organizations face.

What You'll Do

Upon joining McKinsey Analytics, you will have a chance to work and learn from the best in class data scientists, data engineers and analytics translators. Here are some examples of projects you could be engaged in:

- Algorithmic route optimization for revenue improvement of a logistics company.
- Applying analytics in financial institutions' fight against fraud.
- Simulating mining operations for our American gold mining client that identified opportunities to raise capacity by 20-40 percent.
- Using big data to build better predictive models around individual patients to better diagnose and treat disease.

https://datahack.analyticsvidhya.com/contest/mckinsey-analytics-online-hackathon/

Problem Statement

Your Client, a chain of hospitals aiming to create the next generation of healthcare for its patients, has retained McKinsey to help achieve its vision. The company brings the best doctors and enables them to provide proactive health care for its patients. One such investment is a Center of Data Science Excellence.

In this case, your client wants to have study around one of the critical disease “Stroke”. Stroke is a disease that affects the arteries leading to and within the brain. A stroke occurs when a blood vessel that carries oxygen and nutrients to the brain is either blocked by a clot or bursts (or ruptures). When that happens, part of the brain cannot get the blood (and oxygen) it needs, so it and brain cells die.

Over the last few years, the Client has captured several health, demographic and lifestyle details about its patients. This includes details such as age and gender, along with several health parameters (e.g. hypertension, body mass index) and lifestyle related variables (e.g. smoking status, occupation type).

The Client wants you to predict the probability of stroke happening to their patients. This will help doctors take proactive health measures for these patients.

https://dataxboost.wordpress.com/2018/04/17/mckinsey-online-hackathon-on-healthcare/

The Data

Variable	Definition
id	Patient ID
gender	Gender of Patient
age	Age of Patient
hypertension	0 – no hypertension, 1 – suffering from hypertension
heart_disease	0 – no heart disease, 1 – suffering from heart disease
ever_married	Yes/No
work_type	Type of occupation
Residence_type	Area type of residence (Urban/ Rural)
avg_glucose_level	Average Glucose level (measured after meal)
bmi	Body mass index
smoking_status	patient's smoking status
stroke	0 – no stroke, 1 – suffered stroke

Project Overview

- The Data: McKinsey Stroke Dataset
  (available at the course Whiteboard page)
  - This database contains 12 attributes for n=62.001 patients. The “stroke" field refers to the diagnosis.
- The Task:
  - Develop and compare two predictors to predict the probability of a stroke happening to a patient. Make sure to use appropriate data cleaning techniques, such as imputation.
  - You have to use Spark (e.g. pySpark) and SparkSQL for ML and data I/O.
- Available code:
  - https://www.kaggle.com/njalan/healthcare-dataset-stroke-data-pyspark
  - https://github.com/aman1002/McKinseyOnlineHackathon-Healthcare-
- What to deliver: The source code and a report.
- **Deadline: 21.11., 18:00**

Today

- Prepare your system to use PySpark
- HINT: The following tutorials need to be run AFTER the Spark session has been initialized.
- Go through the PySpark ML tutorial and run the examples in the sections „Statistics“, „Pipelines“ and „Regression“
  - https://spark.apache.org/docs/latest/mllib-statistics.html
  - https://spark.apache.org/docs/latest/ml-pipeline.html
  - https://spark.apache.org/docs/latest/mllib-linear-methods.html#linear-least-squares-lasso-and-ridge-regression
- Present your results in class from 13:30.

Spark in Google Colab

[1]	`!apt-get install openjdk-8-jdk-headless -qq > /dev/null`
	`!wget -q https://downloads.apache.org/spark/spark-3.0.1/spark-3.0.1-bin-hadoop3.2.tgz`
	`!tar -xvf spark-3.0.1-bin-hadoop3.2.tgz`
	`!pip install -q findspark`

[2]	`import os`
	`os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"`
	`os.environ["SPARK_HOME"] = "/content/spark-3.0.1-bin-hadoop3.2"`

[3]	`import findspark`
	`findspark.init()`
	`from pyspark.sql import SparkSession`
	
	`spark = SparkSession.builder.master("local[*]").getOrCreate()`

https://colab.research.google.com/drive/1QdEVD6XmYwSXbZcZUlzqL3DjuqW5UwUG?usp=sharing

Freie Universität Berlin

CHARITÉ

Thank You