Data Quality and Data Privacy

Spring 2025

Y. Chandramouli

mchandra@cmi.ac.in

Office: Room 408

Tuesday and Thursday – 3.30 PM -4.45 PM

Course Outline

- Data Quality & Data Privacy
- 2 credit elective course
- Tuesday and Thursday 3.30 PM – 4.50 PM
  - Review of techniques
  - Implementation of algorithms

Course grading

- Quiz & Homework 40/100
- Final examination 30/100
- Course project and presentation 30/100

Data Quality References

- Corinna Cichy and Stefan Rass, An overview of data quality frameworks, 2019
- Lisa Ehrlinger and Wolfram Wöß. A survey of data quality measurement and monitoring tools. Frontiers in big data, page 28, 2022
- Nuno Laranjeiro; Seyma Nur Soydemir; Jorge Bernardino, A Survey on Data Quality: Classifying Poor Data, 2015
  - More references during lectures

What is Data Quality

- Definition of Data Quality
- How to measure Data Quality ?

- What are the challenges ?
- Review of some of techniques
  - Data Deduplication
  - Data Consistency
  - Data Accuracy
  - Information completeness

- How can ML algorithms ML help for Data quality ? In relation to traditional data quality frameworks ?
-

A Conventional Definition of Data Quality

Good quality data are:

ACUUC

- **Accurate, Complete, Unique, Up-to-date**, and **Consistent** ; meaning …

A Conventional Definition of Data Quality

Accurate: This refers to how the data were recorded in the first place.

What might be the *inaccurately recorded* datum in the following table?

Barratt	John	22	Maths	BSc	Male
Burns	Robert	24	CS	BSc	Male
Carter	Laura	20	Physics	MSc	Female
Davies	Michael	12	CS	BSc	Male

age small

A Conventional Definition of Data Quality

- **Complete:** This refers to whether or not the database really contains everything it is *supposed* to contain.

E.g. a patient's medical records should contain references to *all* medication prescribed to date for that patient.

- The Licensing DB should contain an entry for every licensee in the country. Does it?

A Conventional Definition of Data Quality

- **Unique:** Every separate datum appears only once. How many `Data Quality errors' can you find in the following table, and what types are they?

These persons are same, thus uniqueness is to established.

format non uniform

Surname	*Firstname*	*DoB*	*Driving test passed:*
Smith	J.	17/12/85	17/12/05
Smith	Jack	17/12/85	17/12/2005
Smith	Jock	17/12/95	17/12/2005

A Conventional Definition of Data Quality

- **Up-to-date:** The data are kept up to date.

  The post office has just changed my postcode from 600 041 to 600 045. Why does this make it difficult for me to get a sensible quote for home insurance or car insurance?

Can you think of a DB where it *doesn't matter* whether or not the data are kept up to date??

→ No, data needs to be updated.

A Conventional Definition of Data Quality

Consistent: The data contains no logical errors or impossibilities. It makes sense in and of itself.

Why is the following mini DB inconsistent?

Sales - Returns = Net income

Date	*Sales*	*Returns*	*Net income*
23rd Nov	£25,609	£1,003	£24,506
24th Nov	£26,202	£1,601	£24,601
25th Nov	£28,936	£1,178	£25,758

Why there are Data Quality Problems

- **Gathering – Source**
  - Manual data entry (how can we improve this?) → By employing careful/sincere ppl
  - Lack of uniform standards for format and content.
  - Duplicates arising from parallel entry
  - Approximations, alternatives, entries altered in order to cope with s/w and/or h/w constraints.
  - Measurement errors

Why there are Data Quality Problems

- **Transmission**
  - Multiple hops from source to DB – problems can happen anywhere
  - Transmission problems: buffer overflows, checks (did all files arrive, and all correctly?)

Why there are Data Quality Problems

- **Storage**
  - Poor, out of date or inappropriate metadata
  - Missing timestamps
  - conversion to storage format (e.g. to excel files, to higher/lower precision

GTS ⇒ Gathering Transmission Storage.

Where DQ problems occur (integration)

This is the business of combining datasets – e.g. from different parts of a company, from (previously) different companies following an acquisition; from different government agencies, etc.

merging data sets.

- Different keys, different fields, different formats
- Different *definitions* (`customer', `income', …)
- Sociological factors: reluctance to share!

Where DQ problems occur (retrieval/analysis)

- The problem here is usually the quality of DBs that store the retrieved data, or the use of the retrieved data in general.

- Problems arise because:
  - The source DB is not properly understood!
  - Straightforward mistakes in the queries that retrieve the relevant data.

The Several Problems with the Conventional (or any?) Definition

- We can define Data Quality (DQ) in a way that makes it clear what kinds of issues we are thinking about. But is the definition any use beyond that? Can it be used:
- *to Measure* DQ, so that we can say whether or not one DB is better than another, or so we can gauge improvements in DQ over time.
- *on a wide range of different DBs*, of widely different purposes, sizes, etc.

What is the solution then?

DQ Def problems: measurability

- It is not clear how we might measure quality according to some of these items.
  - Completeness: How will we know??
  - Uniqueness: It is hard to tell whether two entries are similar, or duplicates!
  - Up-to-date-ness: How do we know ?
  - Consistent: consistency errors can be very hard to find, especially in a very large DB

The Data Quality Continuum

- It's rare that a datum is entered once into a DB and then left alone. Usually, a datum has a long and varied life, into which errors can arise at each and every stage. The continuum is:
  - Data gathering
  - Data delivery
  - Data storage
  - Data integration
  - Data retrieval
  - Data analysis
- So, if we want to *monitor* DQ, we need to monitor it at each of these stages

= GDSIRA

Agenda

- Last Class
  - Introduction to Data Quality
- Today
  - March 13 – Data Privacy – Introduction
  - Introduction to Data Privacy

What is Data Privacy

- Data privacy generally means the ability of a person to determine for themselves when, how, and to what extent personal information about them is shared with or communicated to others.

- This personal information can be one's name, location, contact information, or online or real-world behavior. Just as someone may wish to exclude people from a private conversation, many online users want to control or prevent certain types of personal data collection.

Data Privacy

- Data Privacy
- Need for privacy
- Review of some of the techniques
- What is differential data privacy ?

Need for Data Privacy

True!!

- Every time we use a service, we have to hand over some of the personal information

- Even without our knowledge some information is generated and captured by companies that we are likely to have never interacted with

Data Privacy

- Definition of Data Privacy

  - Data privacy associated with personally identifiable information (PII), such as names, addresses, Social Security numbers and credit card numbers. This idea also extends to other valuable or confidential data, including financial data, intellectual property and personal health information.

What is Privacy?

What Isn't Privacy?

- Privacy isn't restricting questions to large populations.
  - "What is the average salary of Penn faculty?"
  - "What is the average salary of Penn faculty not named Aaron Roth?"

What Isn't Privacy?

- Privacy isn't "Anonymization"
  - Anonymization isn't enough
    - Collection of medical records from a specific urgent care center and date might correspond to only a small collection of medical conditions.
    - Knowledge (from a neighbor?) that Alice went to that urgent care center doesn't identify her record, but implies she has one of a small number of conditions.

Data Privacy

- Data Privacy issues can arise
  - Healthcare records
  - Criminal justice investigation
  - Financial institutions
  - Biological traits
  - Residence and Geographic records
  - Web surfing behaviour
  - Persistent cookies

HCF BRWP

Data Privacy Breach

- In 1997, when Massachusetts began making health records of state employees available to medical researchers, the government removed patients' names, addresses, and Social Security numbers. William Weld, then the governor, assured the public that identifying individual patients in the records would be impossible.

- Although the state had removed all obvious identifiers, it had left each patient's date of birth, sex and ZIP code. By cross-referencing this information with voter-registration records, Latanya Sweeney was able to pinpoint Weld's records.

Data Privacy References

- https://www.scientificamerican.com/article/privacy-by-the-numbers-a-new-approach-to-safeguarding-data

- Cynthia Dwork. A firm foundation for private data analysis. Communications of the ACM, 54(1):86–95, 2011.

An example implementation

- https://github.com/google/differential-privacy

Later on

- https://github.com/tensorflow/privacy
-

Data Protection Act Principles

The Data Protection Act is the law that protects us against illegal and inappropriate use of our personal information without our consent, and the same applies to us using the information of others

Anyone who processes personal information must comply with eight principles of the Data Protection Act, which make sure that personal information is:

- Fairly and lawfully processed
- Processed for limited purposes
- Adequate, relevant and not excessive
- Accurate and up to date
- Not kept for longer than is necessary
- Processed in line with your rights
- Secure
- Not transferred to other countries without adequate protection

Data Security vs Data Privacy

- Companies must ensure data privacy since the data is an asset to the company
- Data Security is simply the means to the desired end
- No data security policy can overcome the willingness to sell the data

Data Protection Laws

- As of August 2014, over 100 countries around the world have enacted comprehensive data protection legislation, and several other countries are in the process of passing such laws.

- The strongest and most comprehensive laws are in the countries of the European Union and European Economic Area that have implemented the 1995 Data Protection Directive.

- Canada is another leading example with two separate pieces of legislation applying at the national level to government and industry.

Technical Threats

Non-existent Security Architecture

- Some organizations do not have an established security architecture in place, leaving their networks vulnerable to exploitation and the loss of personally identifiable information (PII).

- Inadequate network protection results in increased vulnerability of the data, hardware, and software, including susceptibility to malicious software.

Phishing and Targeted Attacks ("Spear Phishing")

- One way malicious individuals or criminals (e.g., hackers) target individuals and organizations to gain access to personal information is through emails containing malicious code this is referred to as phishing. Once infected emails are opened, the user's machine can be compromised.

Mitigation:

- To reduce vulnerability to phishing and other e-mail security scams, organizations should install professional enterprise-level e-mail security software.

Internet Websites

- Malicious code can be transferred to a computer through browsing webpages that have not undergone security updates.
- Simply browsing the internet and visiting compromised or unsecured websites could result in malicious software being downloaded to an organization's computers and network.
- **Mitigation**: To prevent threats from compromised websites, employ firewalls and antivirus software to help identify and block potentially risky web pages.

Removable media

- The use of removable media on an organization's network poses a significant security threat.
- Without proper protection, these types of media provide a pathway for malware to move between networks or hosts.
- Following proper security measures when using removable media devices is necessary to decrease the risk of infecting organization's machines or the entire network.

pdf - 03

Data Science

Data Science: The Conventional View

A data scientist operating **alone**, on **one static dataset** at a time, with a **clean** "rectangular" shape and fitting in main-memory, employing various statistical and ML algorithms on **predefined objectives**.

- From Data 100
- Also the view reinforced by "popular" Machine Learning, e.g., leaderboards, Kaggle, ...

A lot of data engineering must happen to support the conventional view!

Data Engineering

Nowadays, Data Science also involves **Data Engineering**:

A set of activities that include collecting, collating, extracting, moving, transforming, cleaning, integrating, organizing, representing, storing, and processing data.

- **Messy** (often non-rectangular), **dynamic**, and **large** datasets
- One **team generates** the data, another **team consumes** **it**
- **Unclear** and ill-defined **objectives**
- Necessary precursor to real-world data science & ML
- etc.

[3/4] Why Learn Data Engineering?

1. Data science projects largely focus on data engineering.
2. Data engineer roles >> data scientist roles.
3. Data engineering is essential to ML/AI.

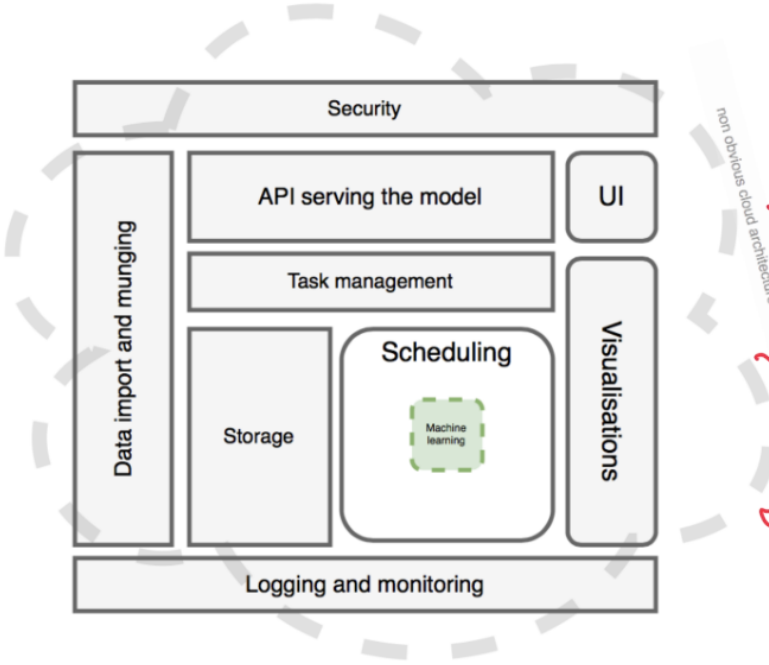
> ...junior people get to the market...come in with an **unrealistic set of expectations** about what data science work will look like. Everyone thinks they're going to be doing machine learning, deep learning, ...

Data science is different now

Feb 13, 2019

Vicky Boykis, 2019.[blog]

NEW: Machine Learning Engineer



Tomasz Dudek,, 2018 [blog].

"ML Engineer": a specialization of data engineer focused on **operationalizing ML**.

"A need for a person that would reunite two warring parties...

One being fluent just enough in both fields [Data Science and Software Engineering] to get the product up and running...

...taking **data scientists' code** and making it more effective and scalable. ..."

(Scalability will be an important focus for us!)

[3/4] Why Learn Data Engineering?

1. Data science projects largely focus on data engineering.
2. Data engineer roles >> data scientist roles.
3. Data engineering is essential to ML/AI.

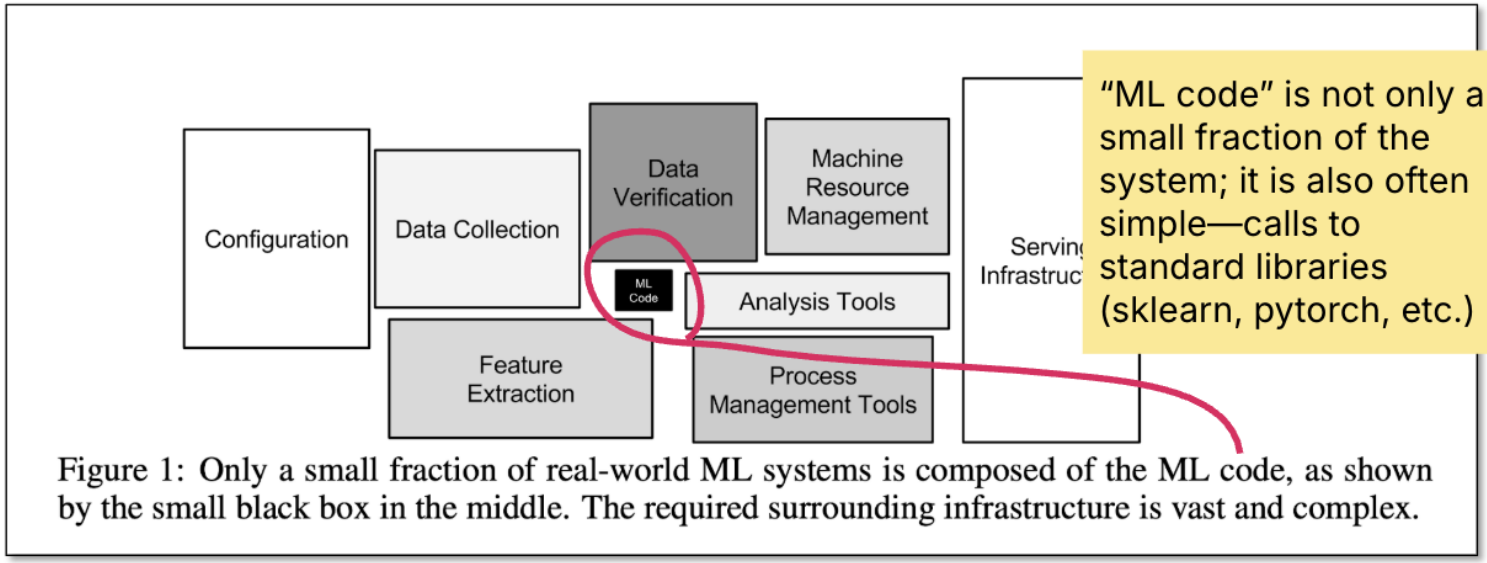
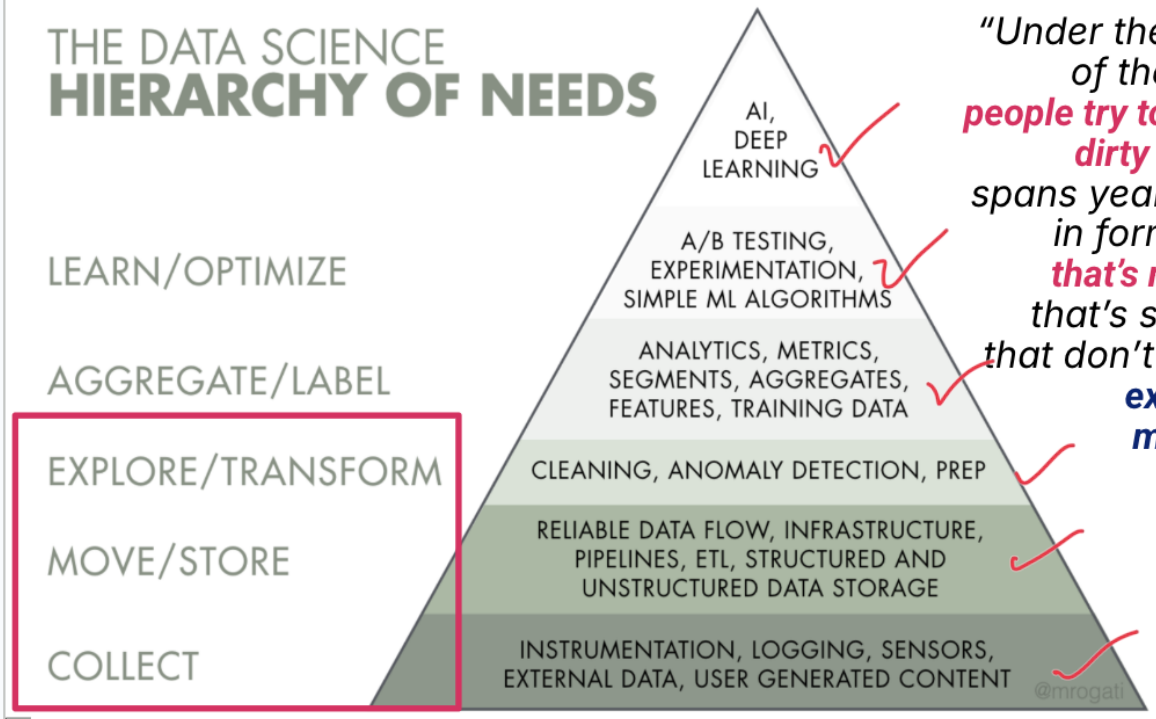


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

"ML code" is not only a small fraction of the system; it is also often simple—calls to standard libraries (sklearn, pytorch, etc.)

Sculley et al., SE4ML 2014 [google research].

Data Engineering is Essential in ML/AI

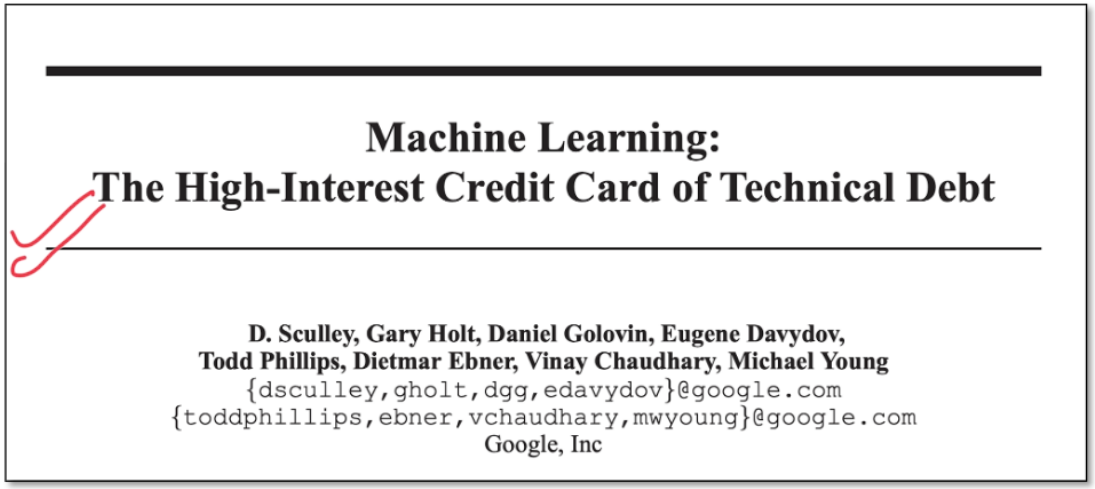


"Under the strong influence of the current AI hype, ***people try to plug in data that's dirty & full of gaps****, that spans years while changing in format and meaning,* ***that's not understood yet****, that's structured in ways that don't make sense, and* ***expect those tools to magically handle it****."*

Monica Rogati, 2017 [blog].

[3/4] Why Learn Data Engineering?

1. Data science projects largely focus on data engineering.
2. Data engineer roles >> data scientist roles.
3. Data engineering is essential to ML/AI.



Machine Learning:
The High-Interest Credit Card of Technical Debt

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov,
Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young
{dsculley,gholt,dgg,edavydov}@google.com
{toddphillips,ebner,vchaudhary,mwyoung}@google.com
Google, Inc

Sculley et al., SE4ML 2014 [google research].

Data Quality outline

- Definition of Data Quality
- How to measure Data Quality ?

- What are the challenges ?
- Review of some of techniques
  - Data Deduplication
  - Data Consistency
  - Data Accuracy
  - Information completeness

- How can ML algorithms ML help for Data quality ? In relation to traditional data quality frameworks ?
-

Consequences of Bad Data Quality

- **Loss of Revenue**
- Reduced Operational Efficiency
- **Flawed Analytics and Decision-Making**
- **Compliance Risks**
- Missed Opportunities
- Reputational Damage

LRF CMR.

Overview

- The meaning of data quality (1)
- The data quality continuum
- The meaning of data quality (2)
- Data quality metrics
- Technical tools
  - Management
  - Statistical
  - Database
  - Metadata
- Case Study
- Research directions

The Meaning of Data Quality (1)

Conventional Definition of Data Quality

- Accuracy
  - The data was recorded correctly.
- Completeness
  - All relevant data was recorded.
- Uniqueness
  - Entities are recorded once.
- Timeliness
  - The data is kept up to date.
    - Special problems in federated data: time consistency.
- Consistency
  - The data agrees with itself.

Meaning of Data Quality (1)

- Generally, you have a problem if the data doesn't mean what you think it does, or should
  - Data not up to spec : garbage in, glitches, etc.
  - You don't understand the spec : complexity, lack of metadata.
- Many sources and manifestations
  - As we will see.
- Data quality problems are expensive and pervasive
  - DQ problems cost hundreds of billion $$$ each year.
  - Resolving data quality problems is often the biggest effort in a data mining study.

Example

T.Das|97336o8327|24.95|Y|-|0.0|1000
Ted J.|973-360-8779|2000|N|M|NY|1000

- Can we interpret the data?
  - What do the fields mean?
  - What is the key? The measures?
- Data glitches
  - Typos, multiple formats, missing / default values
- Metadata and domain expertise
  - Field three is Revenue. In dollars or cents?
  - Field seven is Usage. Is it *censored*?
    - Field 4 is a censored flag. How to handle censored data?

Data Glitches

- Systemic changes to data which are external to the recorded process.
  - Changes in data layout / data types
    - Integer becomes string, fields swap positions, etc.
  - Changes in scale / format
    - Dollars vs. euros
  - Temporary reversion to defaults
    - Failure of a processing step
  - Missing and default values
    - Application programs do not handle NULL values well …
  - Gaps in time series
    - Especially when records represent incremental changes.

Problems …

- Unmeasurable
  - Accuracy and completeness are extremely difficult, perhaps impossible to measure.
- Context independent
  - No accounting for what is important. E.g., if you are computing aggregates, you can tolerate a lot of inaccuracy.
- Incomplete
  - What about interpretability, accessibility, metadata, analysis, etc.
- Vague
  - The conventional definitions provide no guidance towards practical improvements of the data.

Finding a modern definition

- We need a definition of data quality which
  - Reflects the use of the data
  - Leads to improvements in processes
  - Is measurable (we can define metrics)

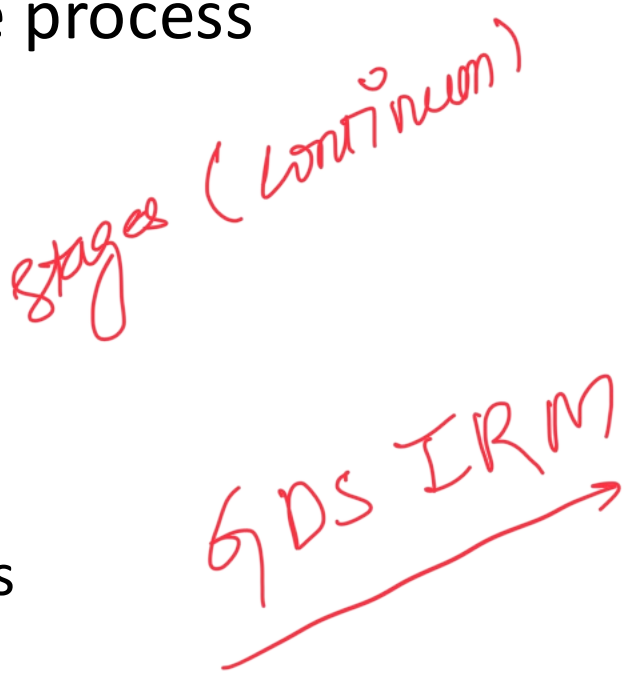
- First, we need a better understanding of how and where data quality problems occur
  - The data quality continuum

Journey of data from raw form to its useful analyzed form.

The Data Quality Continuum

The Data Quality Continuum

- Data and information is not static, it flows in a data collection and usage process
  - Data gathering
  - Data delivery
  - Data storage
  - Data integration
  - Data retrieval
  - Data mining/analysis



Data Gathering

- How does the data enter the system?
- Sources of problems:
  - Manual entry
  - No uniform standards for content and formats
  - Parallel data entry (duplicates)
  - Approximations, surrogates – SW/HW constraints
  - Measurement errors.

Solutions

- Potential Solutions:
  - Preemptive:
    - Process architecture (build in integrity checks)
    - Process management (reward accurate data entry, data sharing, data stewards)
  - Retrospective:
    - Cleaning focus (duplicate removal, merge/purge, name & address matching, field value standardization)
    - Diagnostic focus (automated detection of glitches).

Data Delivery

- Destroying or mutilating information by inappropriate pre-processing
  - Inappropriate aggregation
  - Nulls converted to default values
- Loss of data:
  - Buffer overflows
  - Transmission problems
  - No checks

Solutions

- Build reliable transmission protocols
  - Use a relay server
- Verification
  - Checksums, verification parser
  - Do the uploaded files fit an expected pattern?
- Relationships
  - Are there dependencies between data streams and processing steps
- Interface agreements
  - Data quality commitment from the data stream supplier.

Data Storage

- You get a data set. What do you do with it?
- Problems in physical storage
  - Can be an issue, but terabytes are cheap.
- Problems in logical storage (ER → relations)
  - Poor metadata.
    - Data feeds are often derived from application programs or legacy data sources. What does it mean?
  - Inappropriate data models.
    - Missing timestamps, incorrect normalization, etc.
  - Ad-hoc modifications.
    - Structure the data to fit the GUI.
  - Hardware / software constraints.
    - Data transmission via Excel spreadsheets, Y2K

Solutions

- Metadata
  - Document and publish data specifications.
- Planning
  - Assume that everything bad will happen.
  - Can be very difficult.
- Data exploration
  - Use data browsing and data mining tools to examine the data.
    - Does it meet the specifications you assumed?
    - Has something changed?

Data Integration

- Combine data sets (acquisitions, across departments).
- Common source of problems
  - Heterogenous data : no common key, different field formats
    - Approximate matching
  - Different definitions
    - What is a customer: an account, an individual, a family, …
  - Time synchronization
    - Does the data relate to the same time periods? Are the time windows compatible?
  - Legacy data
    - IMS, spreadsheets, ad-hoc structures
  - Sociological factors
    - Reluctance to share – loss of power.

Solutions

- Commercial Tools
  - Significant body of research in data integration
  - Many tools for address matching, schema mapping are available.
- Data browsing and exploration
  - Many hidden problems and meanings : must extract metadata.
  - View before and after results : did the integration go the way you thought?

Data Retrieval

- Exported data sets are often a view of the actual data. Problems occur because:
  - Source data not properly understood.
  - Need for derived data not understood.
  - Just plain mistakes.
    - Inner join vs. outer join
    - Understanding NULL values
- Computational constraints
  - E.g., too expensive to give a full history, we'll supply a snapshot.
- Incompatibility
  - Ebcdic?

Data Mining and Analysis

- What are you doing with all this data anyway?
- Problems in the analysis.
  - Scale and performance
  - Confidence bounds?
  - Black boxes and dart boards
    - "fire your Statisticians"
  - Attachment to models
  - Insufficient domain expertise
  - Casual empiricism

Solutions

- Data exploration
  - Determine which models and techniques are appropriate, find data bugs, develop domain expertise.
- Continuous analysis
  - Are the results stable? How do they change?
- Accountability
  - Make the analysis part of the feedback loop.

Meaning of Data Quality (2)

- There are many types of data, which have different uses and typical quality problems
  - Federated data
  - High dimensional data
  - Descriptive data
  - Longitudinal data
  - Streaming data
  - Web (scraped) data
  - Numeric vs. categorical vs. text data

write

Meaning of Data Quality (2)

- There are many uses of data
  - Operations
  - Aggregate analysis
  - Customer relations …
- Data Interpretation : the data is useless if we don't know all of the *rules* behind the data.
- Data Suitability : Can you get the answer from the available data
  - Use of proxy data
  - Relevant data is missing

Data Quality Constraints

- Many data quality problems can be captured by *static* constraints based on the schema.
  - Nulls not allowed, field domains, foreign key constraints, etc.
- Many others are due to problems in workflow, and can be captured by *dynamic* constraints
  - E.g., orders above $200 are processed by Biller 2
- The constraints follow an 80-20 rule
  - A few constraints capture most cases, thousands of constraints to capture the last few cases.
- Constraints are measurable. Data Quality Metrics?

Data Quality Metrics

- We want a measurable quantity
  - Indicates what is wrong and how to improve
  - Realize that DQ is a messy problem, no set of numbers will be perfect
- Types of metrics
  - Static vs. dynamic constraints
  - Operational vs. diagnostic
- Metrics should be *directionally correct* with an improvement in use of the data.
- A very large number metrics are possible
  - Choose the most important ones.

Examples of Data Quality Metrics

- Conformance to schema
  - Evaluate constraints on a snapshot.
- Conformance to business rules
  - Evaluate constraints on changes in the database.
- Accuracy
  - Perform inventory (expensive), or use proxy (track complaints). Audit samples?
- Accessibility
- Interpretability
- Glitches in analysis
- Successful completion of end-to-end process

Data Quality Metrics

Data Quality Metrics

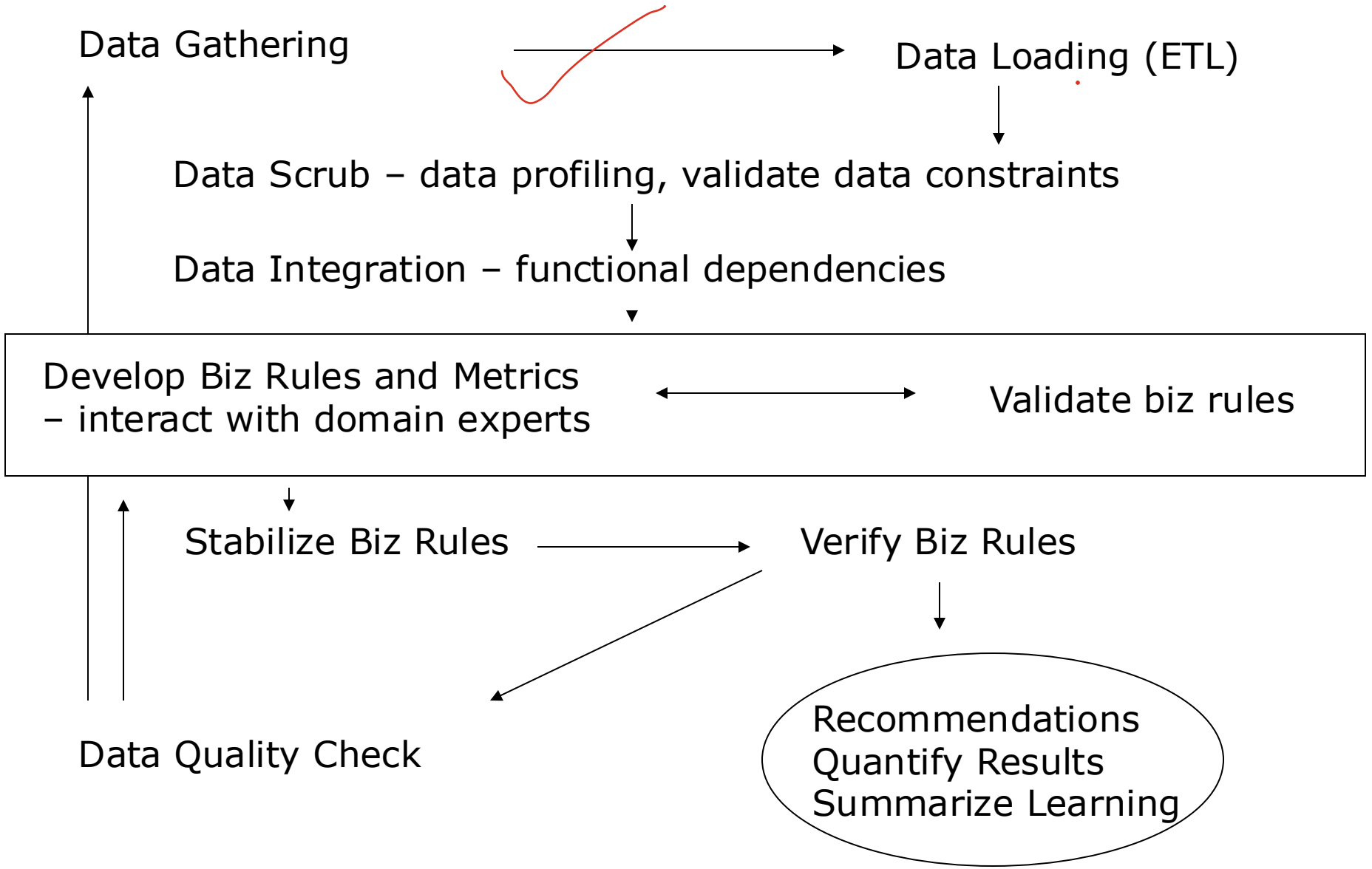
- We want a measurable quantity
  - Indicates what is wrong and how to improve
  - Realize that DQ is a messy problem, no set of numbers will be perfect
- Types of metrics
  - Static vs. dynamic constraints
  - Operational vs. diagnostic
- Metrics should be *directionally correct* with an improvement in use of the data.
- A very large number metrics are possible
  - Choose the most important ones.

Examples of Data Quality Metrics

- Conformance to schema
  - Evaluate constraints on a snapshot.
- Conformance to business rules
  - Evaluate constraints on changes in the database.
- Accuracy
  - Perform inventory (expensive), or use proxy (track complaints). Audit samples?
- Accessibility
- Interpretability
- Glitches in analysis
- Successful completion of end-to-end process

write

Data Quality Process



Technical Tools

Technical Approaches

- We need a multi-disciplinary approach to attack data quality problems
  - No one approach solves all problem
- Process management
  - Ensure proper procedures
- Statistics
  - Focus on analysis: find and repair anomalies in data.
- Database
  - Focus on relationships: ensure consistency.
- Metadata / domain expertise
  - What does it mean? Interpretation

Process Management

- Business processes which encourage data quality.
  - Assign dollars to quality problems
  - Standardization of content and formats
  - Enter data once, enter it correctly (incentives for sales, customer care)
  - Automation
  - Assign responsibility : data stewards
  - End-to-end data audits and reviews
    - Transitions between organizations.
  - Data Monitoring
  - Data Publishing
  - Feedback loops

write

Feedback Loops

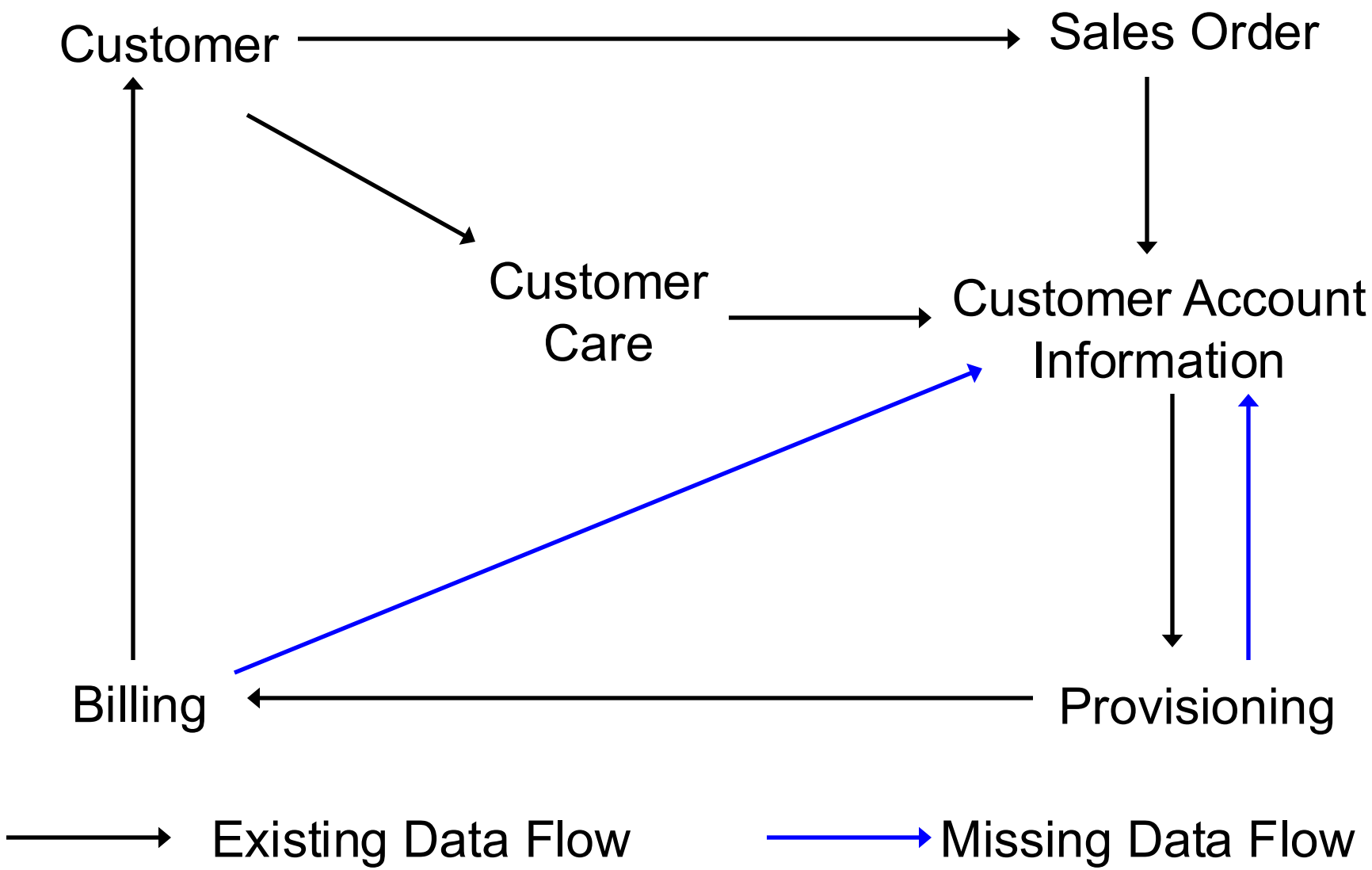
- Data processing systems are often thought of as open-loop systems.
  - Do your processing then throw the results over the fence.
  - Computers don't make mistakes, do they?
- Analogy to control systems : *feedback loops*.
  - *Monitor* the system to detect difference between actual and intended
  - *Feedback loop* to correct the behavior of earlier components
  - Of course, data processing systems are much more complicated than linear control systems.

Example

- Sales, provisioning, and billing for telecommunications service
  - Many stages involving handoffs between organizations and databases
  - Simplified picture
- *Transition between organizational boundaries is a common cause of problems*.
- Natural feedback loops
  - Customer complains if the bill is to high
- Missing feedback loops
  - No complaints if we undercharge.

write

Example



Monitoring

- Use data monitoring to add missing feedback loops.
- Methods:
  - Data tracking / auditing
    - Follow a sample of transactions through the workflow.
    - Build secondary processing system to detect possible problems.
  - Reconciliation of incrementally updated databases with original sources.
  - Mandated consistency with a Database of Record (DBOR).
  - Feedback loop sync-up
  - Data Publishing

Data Publishing

- Make the contents of a database available in a readily accessible and digestible way
  - Web interface (universal client).
  - Data Squashing : Publish aggregates, cubes, samples, parametric representations.
  - Publish the metadata.
- Close feedback loops by getting a lot of people to look at the data.
- Surprisingly difficult sometimes.
  - Organizational boundaries, loss of control interpreted as loss of power, desire to hide problems.

Statistical Approaches

- No explicit DQ methods
  - Traditional statistical data collected from carefully designed experiments, often tied to analysis
  - But, there are methods for finding anomalies and repairing data.
  - Existing methods can be adapted for DQ purposes.
- Four broad categories can be adapted for DQ
  - Missing, incomplete, ambiguous or damaged data e.g truncated, censored
  - Suspicious or abnormal data e.g. outliers
  - Testing for departure from models
  - Goodness-of-fit

Missing Data

- Missing data - values, attributes, entire records, entire sections
- Missing values and defaults are indistinguishable
- Truncation/censoring - not aware, mechanisms not known
- Problem: Misleading results, bias.

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; check at various levels of granularity since aggregates can be misleading.

Missing data detection (cont.)

- Hidden damage to data
  - Values are truncated or censored - check for spikes and dips in distributions and histograms
  - Missing values and defaults are indistinguishable - too many missing values? metadata or domain expertise can help
  - Errors of omission e.g. all calls from a particular area are missing - check if data are missing randomly or are localized in some way

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

Missing Value Imputation - 1

- Standalone imputation
  - 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 - 2

- Better imputation - use attribute relationships
- Assume : all prior attributes are populated
  - That is, *monotonicity* in missing values.

X1	X2	X3	X4	X5
1.0	20	3.5	4	.
1.1	18	4.0	2	.
1.9	22	2.2	.	.
0.9	15	.	.	.

- Two techniques
  - Regression (parametric),
  - Propensity score (nonparametric)

Missing Value Imputation –3

- Regression method
  - Use linear regression, sweep left-to-right

    X3=a+b*X2+c*X1;

    X4=d+e*X3+f*X2+g*X1, and so on
  - X3 in the second equation is estimated from the first equation if it is missing

Missing Value Imputation - 3

- Propensity Scores (nonparametric)
  - Let Y_j=1 if X_j is missing, 0 otherwise
  - Estimate P(Y_j =1) based on X_1 through $X_{(j-1)}$ using logistic regression
  - Group by propensity score P(Y_j =1)
  - Within each group, estimate missing X_js from known X_js using approximate Bayesian bootstrap.
  - Repeat until all attributes are populated.

Missing Value Imputation - 4

- Arbitrary missing pattern
  - Markov Chain Monte Carlo (MCMC)
  - Assume data is multivariate Normal, with parameter Θ
  - (1) Simulate missing X, given Θ estimated from observed X ; (2) Re-compute Θ using filled in X
  - Repeat until stable.
  - Expensive: Used most often to induce monotonicity

- Note that imputed values are useful in aggregates but can't be trusted individually

Censoring and Truncation

- Well studied in Biostatistics, relevant to time dependent data e.g. duration

imp

- *Censored* - Measurement is bounded but not precise e.g. Call duration > 20 are recorded as 20
- *Truncated* - Data point dropped if it exceeds or falls below a certain bound e.g. customers with less than 2 minutes of calling per month

Consequences of Bad Data Quality

- Bad data quality can have a significant impact on organizations, some of which include:

- **Inaccurate or unreliable decision making**: poor data quality can lead to incorrect or incomplete information being used to make decisions, resulting in poor outcomes.
- **Reduced productivity and efficiency**: bad data quality can lead to wasted time and resources, as employees may have to spend time correcting errors or searching for missing information.
- **Increased costs:** poor data quality can lead to increased costs, such as the cost of correcting errors or re-doing work that was based on incorrect information.

Consequences of Bad Data Quality

- **Loss of trust and credibility:** bad data quality can damage an organization's reputation and lead to loss of trust from clients, customers, and other stakeholders.

- **Compliance issues:** bad data quality can lead to non-compliance with regulations and laws, such as GDPR, HIPAA, and SOX.

- Inability to effectively use data analytics and business intelligence: poor data quality can make it difficult to extract insights from data and make it difficult to use data to improve decision making.

Consequences of Bad Data Quality

- **Poor ML models :** bad data quality can have a significant impact on machine learning models, reducing their accuracy, making them more complex and harder to maintain, and making it difficult to train, evaluate, and interpret their results. It is important to ensure that data is of good quality, accurate, and unbiased, before using it in any ML model.

- **Difficulty in integrating data from different systems:** poor data quality can make it difficult to merge data from multiple sources, which can limit the insights that can be gained from the data.

Great Expectations Overview

Great Expectations (GX) is one of the most popular data quality tools. The core idea behind creating Great Expectations was "instead of just testing code, and we should be testing data. After all, that's where the complexity lives."

The creators of GX were on the money. They built the tool on an expectation (of data quality) that can be tested by running pre-

Great Expectations

Soda Core

Soda Core Overview

Soda Core is an open-source Python library built to enable data reliability in your data platform. It comes with its command-line tool. It supports SodaCL (Soda Checks Language), a domain-specific, YAML-compatible language written with reliability in mind. Soda Core can connect to data sources and workflows to

Technical Tools

Data Quality Tools

- Commercial Tools

  - Informatica Data Quality
  - Trifacta
  - Microsoft

Data Quality Tools

- Open Source Tools

  - Trifacta
  - Deequ (Amazon)
  - Google Data Quality
  - Pandas

Ydata Profiling

- The primary goal is to provide a one-line Exploratory Data Analysis (EDA) experience in a consistent and fast solution. Like pandas df.describe() function, that is so handy, ydata-profiling delivers an extended analysis of a DataFrame while allowing the data analysis to be exported in different formats such as html and json.

Ydata Profiling

Use case	Description
Comparing datasets	Comparing multiple version of the same dataset
Profiling a Time-Series dataset	Generating a report for a time-series dataset with a single line of code
Profiling large datasets	Tips on how to prepare data and configure `ydata-profiling` for working with large datasets
Handling sensitive data	Generating reports which are mindful about sensitive data in the input dataset
Dataset metadata and data dictionaries	Complementing the report with dataset details and column-specific data dictionaries
Customizing the report's appearance	Changing the appearance of the report's page and of the contained visualizations
Profiling Databases	For a seamless profiling experience in your organization's databases, check Fabric Data Catalog, which allows to consume data from different types of storages such as RDBMs (Azure SQL, PostGreSQL, Oracle, etc.) and object storages (Google Cloud Storage, AWS S3, Snowflake, etc.), among others.

Great Expectations

Great Expectations Overview

- The creators of GX were on the money. They built the tool on an expectation (of data quality) that can be tested by running pre-defined and templated tests by connecting to your data sources. In the official integration guides, find more about GX integrations with tools and platforms like Databricks, Flyte, Prefect, and EMR.

- Great Expectation is actively maintained and is known to be used by Vimeo, Calm, ING, Glovo, Avito, DeliveryHero, Atlan, and Heineken, among others.

Great Expectations Features

- GX has an exhaustive list of Expectations that prescribe the "expected state of the data." GX's integrations with the data sources mean that all the data quality checks are done in place, and no data is moved out of the data source.
- GX also supports data contracts by automating data quality checks, recording the results over time, and giving you a human-readable summary of the test runs.
- On top of data sources, such as databases and data warehouses, GX also directly connects with source metadata aggregators and data catalogs, and orchestration engines, such as Airflow, Meltano, and Dagster.
- GX is flexible with storage backends, i.e., you can store Expectations, Validation Results, and Metrics in AWS S3, Azure Blob Storage, Google Cloud Storage, PostgreSQL, or a file system.

Soda Core

- **Soda Core Overview**

Soda Core is an open-source Python library built to enable data reliability in your data platform. It comes with its command-line tool. It supports SodaCL (Soda Checks Language), a domain-specific, YAML-compatible language written with reliability in mind. Soda Core can connect to data sources and workflows to ensure data quality within and outside your data pipelines.

Soda Core

- **Soda Core Overview**
- With an extensive range of data sources, connectors, and test types, Soda Core provides one of the most comprehensive test surface area coverages among open-source data quality tools. In addition to the standard connectors, Soda Core supports new and trending connectors to sources like Dask, DuckDB, Trino, and Dremio.

Data Quality with PyDeequ

- PyDeequ is a Python library that provides a high-level API for using Deequ, an open-source library for data quality assessment, constraint verification, and data profiling. Amazon initially developed Deequ and later contributed to the open-source community.

PyDeequ Features

- Metrics Computation
  - We can better understand the dataset's health and reliability by computing these data quality metrics.
- Constraint Verification
- Constraint suggestion

Metrics Computation

- Completeness: Measures the percentage of non-null values for a given column or set of columns.
- b. Uniqueness: Determines the percentage of unique values for a specific column or set of columns, helping to identify potential duplicate entries.
- c. Distinctness: Evaluates the percentage of distinct values among all the rows in a dataset.
- d. Approximate Count Distinct: Calculates an approximate count of distinct values in a column, which is useful for large datasets where an exact count might be computationally expensive.

Metrics Computation

- e. Entropy: Measures the uncertainty or randomness in a column, indicating how evenly distributed the values are.
- f. Mutual Information: Measures the degree of dependency between two columns, providing insights into the relationship between them.
- g. Functional Dependency: Determining if one column functionally depends on another, indicating potential data quality issues.
- h. Compliance: Checks if the data complies with predefined business rules and constraints.

Data Quality at Scale using Pydeequ

- `import os os.environ["SPARK_VERSION"] = '3.3'`
- `pip install pydeequ==1.2.0`
- `import sagemaker_pyspark`
- `import pydeequ`

- `pip install pydeequ==1.2.0`
- `pip install sagemaker_pyspark`

- `import sagemaker_pyspark import pydeequ`

Data Quality at Scale using Pydeequ

```
import os os.environ["SPARK_VERSION"] = '3.3’
from pyspark.sql import SparkSession, Row, DataFrame
import json
import pandas as pd
classpath = ":".join(sagemaker_pyspark.classpath_jars())
spark = (SparkSession
    .builder
    .config("spark.driver.extraClassPath", classpath)
    .config("spark.jars.packages", pydeequ.deequ_maven_coord)
    .config("spark.jars.excludes", pydeequ.f2j_maven_coord)
    .config("spark.driver.memory", "15g")
    .config("spark.sql.parquet.int96RebaseModeInRead", "CORRECTED")
    .getOrCreate())
```

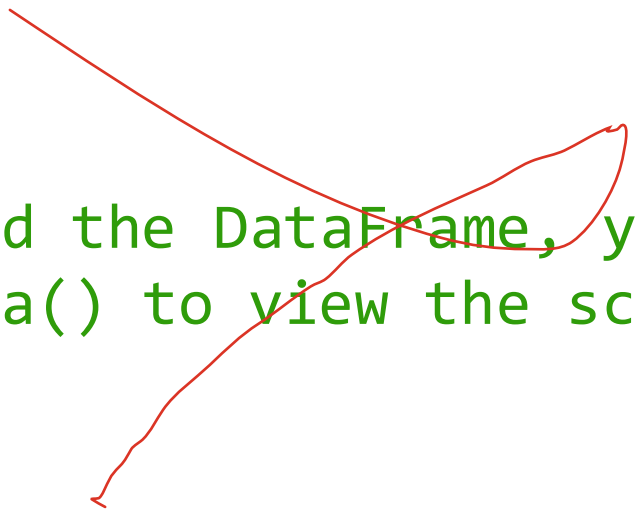
Data Quality at Scale using Pydeequ

Read the dataset

Read the dataset with the following code:

- df = spark.read.parquet("s3a://aws-bigdata-blog/generated_synthetic_reviews/data/product_category=Electronics/")

- After you load the DataFrame, you can run df.printSchema() to view the schema of the dataset



Metrics Computation

Name	Instance	Value
Distinctness	review_id	0.99266
Completeness	review_id	1
Compliance	top star_rating	0.74999
Correlation	helpful_votes,total_votes	0.98179
Correlation	total_votes,star_rating	-7.3881 not corr
Mean	star_rating	3.99999
Size	*	3010972

Metrics Computation

From this, we learn the following:

- `review_id` has no missing values and approximately 99.27% of the values are distinct
- 74.99% of reviews have a `star_rating` of 4 or higher
- `total_votes` and `star_rating` are not correlated
- `helpful_votes` and `total_votes` are strongly correlated
- The average `star_rating` is 3.99
- The dataset contains 3,010,972 reviews

Sometimes, you may want to run multiple metrics on a single column. For example, you want to check that all reviews were written either after 1996 or before 2017. In this case, it's helpful to provide a name for each metric in order to distinguish the results in the output:

Metrics Computation

```
from pydeequ.checks import *
from pydeequ.verification import *

check = Check(spark, CheckLevel.Warning, "Synthetic Product Reviews")

checkResult = VerificationSuite(spark) \
    .onData(df) \
    .addCheck(
        check.hasSize(lambda x: x >= 3000000) \
        .hasMin("star_rating", lambda x: x == 1.0) \
        .hasMax("star_rating", lambda x: x == 5.0)  \
        .isComplete("review_id")  \
        .isUnique("review_id")  \
        .isComplete("marketplace")  \
        .isContainedIn("marketplace", ["US", "UK", "DE", "JP", "FR"]) \
        .isNonNegative("year") \
        .hasMin("review_year", lambda x: x == '1996') \
        .hasMax("review_year", lambda x: x == '2017')) \
    .run()
```

Metrics Computation

Sometimes, you may want to run multiple metrics on a single column. For example, you want to check that all reviews were written either after 1996 or before 2017. In this case, it's helpful to provide a name for each metric in order to distinguish the results in the output:

```
analysisResult = AnalysisRunner(spark) \
                    .onData(df) \
                    .addAnalyzer(Compliance("after-1996 review_year",
"review_year >= 1996")) \
                    .addAnalyzer(Compliance("before-2017 review_year",
"review_year <= 2017")) \
                    .run()
analysisResult_pd_df = AnalyzerContext.successMetricsAsDataFrame(spark,
analysisResult, pandas=True)
analysisResult_pd_df
```

Data Governance

- Financial institutions such as FINRA, Nasdaq, and National Australia Bank have built data lakes on AWS to collect, store, and analyze increasing amounts of data at speed and scale. A data lake allows organizations to break down data silos and store all of their data – structured, semi-structured, and unstructured – in a centralized repository at any scale.

isolated repositories of information within an organization

Data Governance

- As an example, in order to comply with Anti Money Laundering (AML) compliance requirement, a financial institution needs to detect and report suspicious activities, such as security fraud and market manipulation. Criminals use money laundering techniques to conceal their activities. In order to properly track, trace, and uncover such activities, a financial institution should collect and centralize transactions from all lines of business and their respective applications, create a 360-degree view of the customer, product, and transactions, and apply AML detective analytical scenarios.

Data Governance

- One of the key components to creating and maintaining an effective data lake is data governance. Data governance refers to a framework that includes people, processes, and technology that enables business users to work collaboratively with technologists to drive clean, certified, and trusted data. Without governance, a data lake becomes a data swamp where data continues to be consumed in a siloed manner without consistency and accuracy. As a result, the original issue is not fixed, it just resurfaces on a new platform.

Data Governance

- A data governance framework consists of multiple components, including data quality, data ownership, data catalog, data lineage, operation, and compliance. In this blog we will be focused on data quality.

Data Governance

- There are two types of data quality issues that can arise in a data lake.

- The first is operational where information such as a customer's birth date or address is entered incorrectly by a person or a system. This type of data quality issue should be assigned to the source application owner and be remediated at the source system. One example of remediation would be scrubbing, which includes reaching out to the customers to confirm certain information or referencing the original onboarding documents. Another form of remediation would be to leverage data quality metrics to enhance business processes. For example, if the majority of customer birth dates are null, the source system's application can be updated so that it requires a valid date of birth before allowing a customer to be onboarded. Additionally, machine learning capabilities such as computer vision and optical character recognition (OCR) could be used to automatically extract a customer's name, address, and birth date from their driver license, and therefore, minimizing errors resulting from manual data entry.

Data Governance

- There are two types of data quality issues that can arise in a data lake.

- The second type of data quality issue is introduced when data siloes are being integrated into the data lake. During this integration process, data from various source systems are fed into a centralized data lake, often resulting in the same attributes having different values and formats. These attributes are maintained on siloed databases at the source system as their usage is directly linked to the operation of the source system. Therefore, it is natural for the various data siloes to have inconsistent values for the same attributes. While this scenario is considered a data quality issue for the data lake, it cannot be considered the same for the source system. This type of data quality issue requires the integration job to include additional logic to conform critical data elements to standardized, consistent values if a change in source system is not feasible. For example, if the customer's birth date is not populated for their loan application, the bank can leverage another data source and populate this field from the customer's deposit record.

Data Governance

- To address both types of data quality issues discussed earlier, it is imperative for customers that are leveraging data lake architectures to have a well-defined data quality framework. This framework should include:

  - An end-to-end data quality lifecycle
  - Responsibility model
  - The technology to enable the objective

Data Governance

Data Governance

- **Data Discovery:** Requirement gathering, source application identification, data collection, organization, and data quality report classification

- **Data Profiling:** Initial examination, sample data quality check, rule suggestion, and approval of final data quality rule

- **Data Rules**: Execution of final business rule to examine accuracy of the data, and its fit for purpose

- **Data Distribution and Remediation:** Process of distributing the data quality reports to the responsible parties and start of remediation process

- **Data Monitoring:** Ongoing monitoring of remediation process, and creation of data quality dashboards and score cards

Technical Tools

Data Quality Tools

- Commercial Tools
  - Informatica Data Quality
  - Trifacta
  - Microsoft

Data Quality Tools

- Open Source Tools

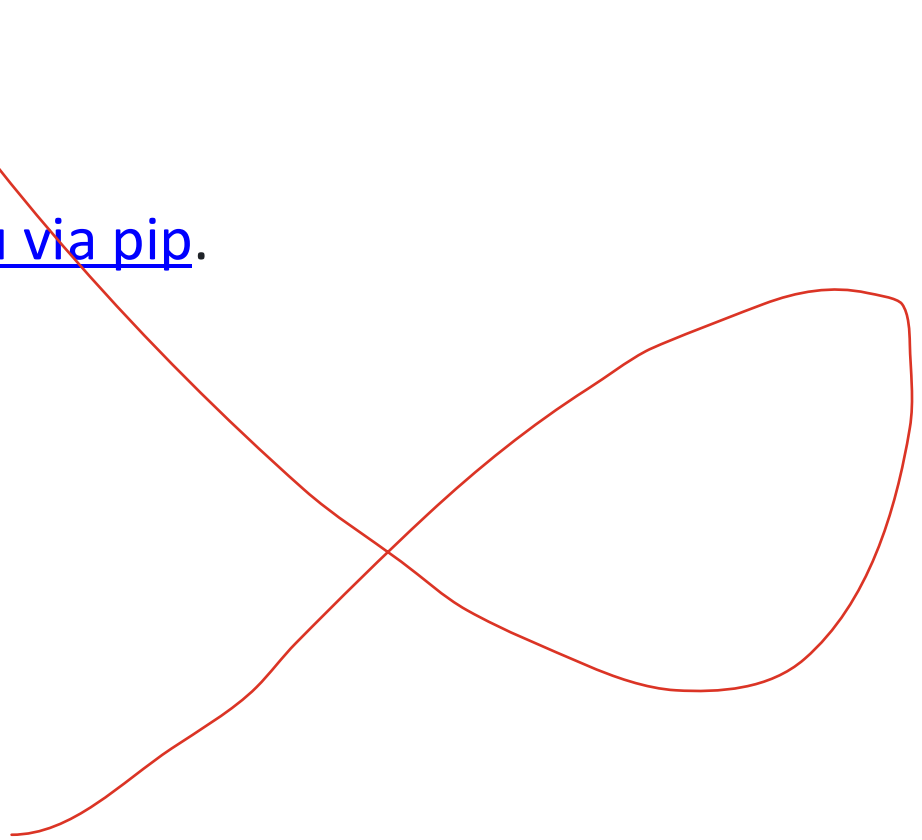
  - Trifacta
  - Deequ (Amazon)
  - Google Data Quality
  - Pandas

Data Quality at Scale using Pydeequ

Quickstart

Installation

- You can install PyDeequ via pip.
- pip install pydeequ



Data Quality at Scale using Pydeequ

PyDeequ, an open source Python wrapper over Deequ (an open source tool developed and used at Amazon).

Deequ is written in Scala, whereas PyDeequ allows you to use its data quality and testing capabilities from Python and PySpark, the language of choice for many data scientists.

PyDeequ democratizes and extends the power of Deequ by allowing you to use it alongside the many data science libraries that are available in that language. Furthermore, PyDeequ allows for fluid interface with pandas DataFrames as opposed to restricting within Apache Spark DataFrames.

Data Quality at Scale using Pydeequ

Deequ allows you to

- calculate data quality metrics for your dataset,
- define and verify data quality constraints

Deequ supports you by suggesting checks for you. Deequ is implemented on top of Apache Spark and is designed to scale with large datasets (billions of rows) that typically live in a data lake, distributed file system, or a data warehouse.

PyDeequ gives you access to this capability, but also allows you to use it from the familiar environment of your Python Jupyter notebook.

Data Quality at Scale using Pydeequ

- Deequ is used internally at Amazon to verify the quality of many large production datasets. Dataset producers can add and edit data quality constraints.

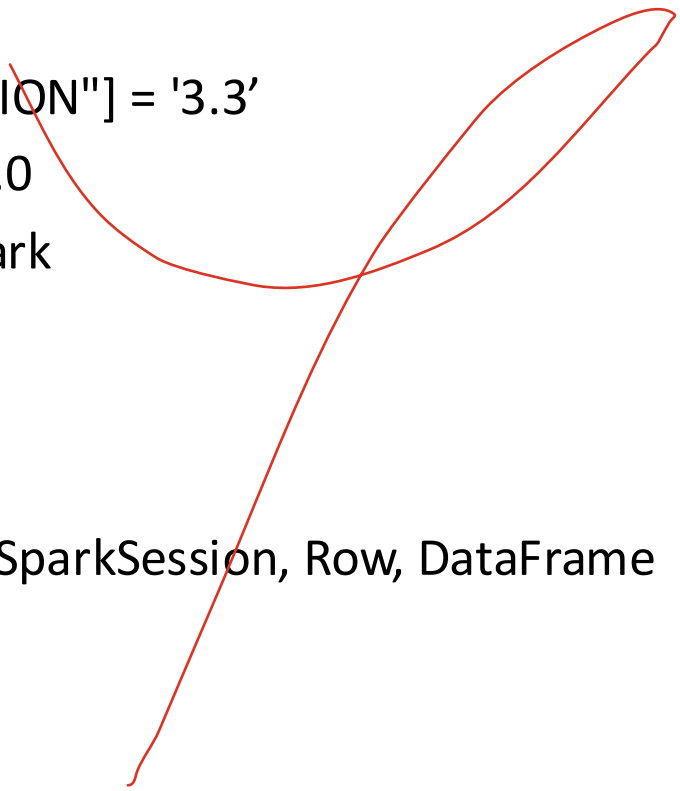
- The system computes data quality metrics on a regular basis (with every new version of a dataset), verifies constraints defined by dataset producers, and publishes datasets to consumers in case of success.

- In error cases, dataset publication can be stopped, and producers are notified to take action. Data quality issues don't propagate to consumer data pipelines, reducing their area of impact.

Pydeequ Setup

Setup

```
import os
os.environ["SPARK_VERSION"] = '3.3’
pip install pydeequ==1.2.0
import sagemaker_pyspark
import pydeequ
```

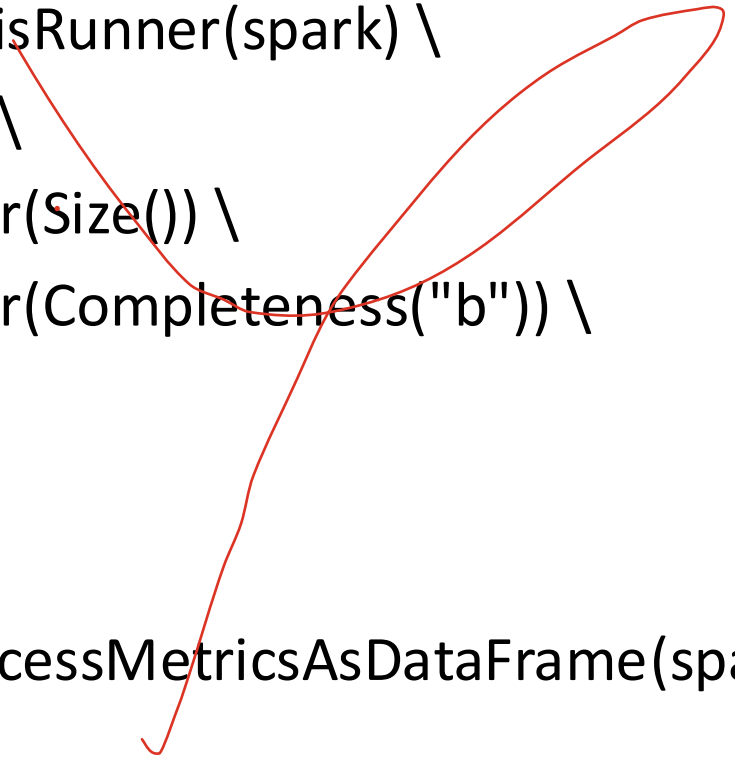


Start a PySpark Session

```
from pyspark.sql import SparkSession, Row, DataFrame
import json
import pandas as pd

classpath = ":".join(sagemaker_pyspark.classpath_jars())
```

Data Quality at Scale using Pydeequ

```
from pydeequ.analyzers import *
analysisResult = AnalysisRunner(spark) \
                    .onData(df) \
                    .addAnalyzer(Size()) \
                    .addAnalyzer(Completeness("b")) \
                    .run()

analysisResult_df =
    AnalyzerContext.successMetricsAsDataFrame(spark,
    analysisResult)
analysisResult_df.show()
```



Read the dataset

```
df = spark.read.parquet("s3a://aws-bigdata-
    blog/generated_synthetic_reviews/data/product_category=El
    ectronics/")
```

df.printSchema()

```
root
 |-- marketplace: string (nullable = true)
 |-- customer_id: string (nullable = true)
 |-- review_id: string (nullable = true)
 |-- product_id: string (nullable = true)
 |-- product_title: string (nullable = true)
 |-- star_rating: long (nullable = true)
 |-- helpful_votes: long (nullable = true)
 |-- total_votes: long (nullable = true)
 |-- insight: string (nullable = true)
 |-- review_headline: string (nullable = true)
 |-- review_body: string (nullable = true)
 |-- review_date: timestamp (nullable = true)
 |-- review_year: long (nullable = true)
```

Data Analysis

```
from pydeequ.analyzers import *

analysisResult = AnalysisRunner(spark) \
                    .onData(df) \
                    .addAnalyzer(Size()) \
                    .addAnalyzer(Completeness("review_id")) \
                    .addAnalyzer(Distinctness("review_id")) \
                    .addAnalyzer(Mean("star_rating")) \
                    .addAnalyzer(Compliance("top star_rating", "star_rating >= 4.0")) \
                    .addAnalyzer(Correlation("total_votes", "star_rating")) \
                    .addAnalyzer(Correlation("total_votes", "helpful_votes")) \
                    .run()

analysisResult_df = AnalyzerContext.successMetricsAsDataFrame(spark,
    analysisResult)
```

Data Analysis

```
from pydeequ.analyzers import *

analysisResult = AnalysisRunner(spark) \
                    .onData(df) \
                    .addAnalyzer(Size()) \
                    .addAnalyzer(Completeness("review_id")) \
                    .addAnalyzer(Distinctness("review_id")) \
                    .addAnalyzer(Mean("star_rating")) \
                    .addAnalyzer(Compliance("top star_rating", "star_rating >= 4.0")) \
                    .addAnalyzer(Correlation("total_votes", "star_rating")) \
                    .addAnalyzer(Correlation("total_votes", "helpful_votes")) \
                    .run()

analysisResult_df = AnalyzerContext.successMetricsAsDataFrame(spark,
    analysisResult)
```

Results

can give to interpret

Out[5]:

	entity	instance	name	value
0	Column	review_id	Completeness	1.000000e+00
1	Column	review_id	ApproxCountDistinct	3.160409e+06
2	Mutlicolumn	total_votes,star_rating	Correlation	-7.388090e-04
3	Dataset	*	Size	3.010972e+06
4	Column	star_rating	Mean	3.999997e+00
5	Column	top star_rating	Compliance	7.499993e-01
6	Mutlicolumn	total_votes,helpful_votes	Correlation	9.817923e-01

Results

code written afterwards

At least 3 million rows in total

review_id is never null

review_id is unique

star_rating has a minimum of 1.0 and maximum of 5.0

marketplace only contains US, UK, DE, JP, or FR

year does not contain negative values

year is between 1996 and 2017*

Results

repeated

- review_id has no missing values and approximately 99.27% of the values are distinct
- 74.99% of reviews have a star_rating of 4 or higher

- **total_votes and star_rating are not correlated**
- **helpful_votes and total_votes are strongly correlated**

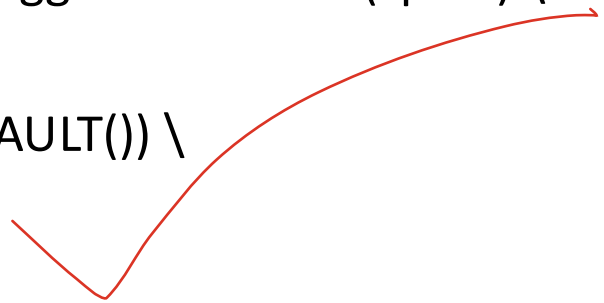
- The average star_rating is 3.99
- The dataset contains 3,010,972 reviews

Verification

```
checkResult = VerificationSuite(spark) \
    .onData(df) \
    .addCheck(
        check.hasSize(lambda x: x >= 3000000) \
        .hasMin("star_rating", lambda x: x == 1.0) \
        .hasMax("star_rating", lambda x: x == 5.0)  \
        .isComplete("review_id")  \
        .isUnique("review_id")  \
        .isComplete("marketplace")  \
        .isContainedIn("marketplace", ["US", "UK", "DE", "JP", "FR"]) \
        .isNonNegative("year") \
        .hasMin("review_year", lambda x: x == '1996') \
        .hasMax("review_year", lambda x: x == '2017')) \
    .run()

checkResult_df = VerificationResult.checkResultsAsDataFrame(spark,
checkResult, pandas=True)
```

Automated Constraint Suggestion

```
from pydeequ.suggestions import *

suggestionResult = ConstraintSuggestionRunner(spark) \
        .onData(df) \
        .addConstraintRule(DEFAULT()) \
        .run()

# Constraint Suggestions in JSON format
print(json.dumps(suggestionResult, indent=2))
```



Automated Constraint Suggestion


```
{
  "constraint_suggestions": [
    {
      "constraint_name": "ComplianceConstraint(Compliance('insight' has value range 'N', 'Y',`insight` IN ('N', 'Y'),None))",
      "column_name": "insight",
      "current_value": "Compliance: 1",
      "description": "'insight' has value range 'N', 'Y'",
      "suggesting_rule": "CategoricalRangeRule(com.amazon.deequ.suggestions.rules.CategoricalRangeRule$$$Lambda$4119/0x000000080197e840@74f276b0)",
      "rule_description": "If we see a categorical range for a column, we suggest an IS IN (...) constraint",
      "code_for_constraint": ".isContainedIn(\"insight\", [\"N\", \"Y\"])"
    }
```

Automated Constraint Suggestion


```
{
  "constraint_name":
  "CompletenessConstraint(Completeness(insight,None))",
  "column_name": "insight",
  "current_value": "Completeness: 1.0",
  "description": "'insight' is not null",
  "suggesting_rule": "CompleteIfCompleteRule()",
  "rule_description": "If a column is complete in the sample, we suggest a
  NOT NULL constraint",
  "code_for_constraint": ".isComplete(\"insight\")"
},
```

Automated Constraint Suggestion

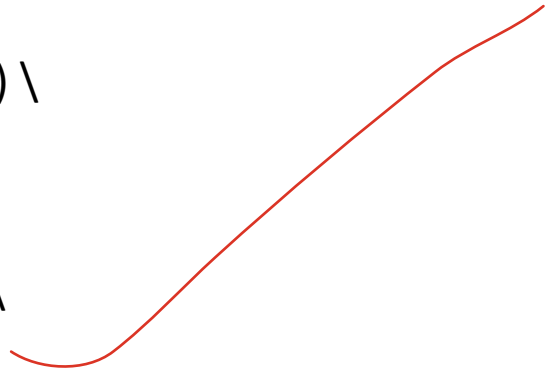

```
{
  "constraint_name":
  "CompletenessConstraint(Completeness(review_id,None))",
  "column_name": "review_id",
  "current_value": "Completeness: 1.0",
  "description": "'review_id' is not null",
  "suggesting_rule": "CompleteIfCompleteRule()",
  "rule_description": "If a column is complete in the sample, we suggest a
  NOT NULL constraint",
  "code_for_constraint": ".isComplete(\"review_id\")"
},
```

Automated Constraint Suggestion


```
{
    "constraint_name": "ComplianceConstraint(Compliance('helpful_votes'
    has no negative values,helpful_votes >= 0,None))",
    "column_name": "helpful_votes",
    "current_value": "Minimum: 3.0",
    "description": "'helpful_votes' has no negative values",
    "suggesting_rule": "NonNegativeNumbersRule()",
    "rule_description": "If we see only non-negative numbers in a column, we
    suggest a corresponding constraint",
    "code_for_constraint": ".isNonNegative(\"helpful_votes\")"
  },
```

Constraint Verification

```
from pydeequ.checks import *
from pydeequ.verification import *

check = Check(spark, CheckLevel.Warning, "Review Check")

checkResult = VerificationSuite(spark) \
    .onData(df) \
    .addCheck(
        check.hasSize(lambda x: x >= 3) \
        .hasMin("b", lambda x: x == 0) \
        .isComplete("c")  \
        .isUnique("a")  \
        .isContainedIn("a", ["foo", "bar", "baz"]) \
        .isNonNegative("b")) \
    .run()

checkResult_df = VerificationResult.checkResultsAsDataFrame(spark, checkResult)
checkResult_df.show()
```



Google Data Quality Monitor

- Data is the most important part of a modern business strategy. However, it's hard to maintain the robust foundation necessary for supporting data-driven decisions.

- Data Quality Monitor (DQM) aims to empower clients with an easy way to monitor their data. It runs on Google Cloud Platform (GCP) and can act on any data sitting in BigQuery, including exports from various Google Ads & Marketing Platform connectors.

- The checks/rules are configured with a simple JSON file and managed with scheduled Cloud Workflows. The output are logs that can be visualized and monitored for subsequent action. We also provide templates for common use cases.

Google Data Quality Monitor

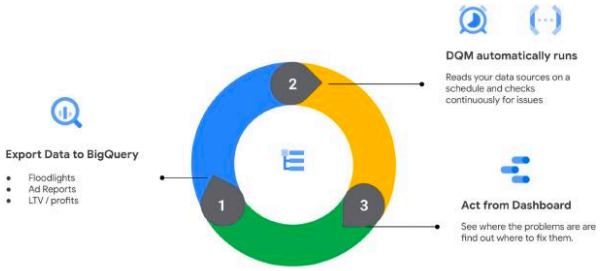
Data Quality Monitor

What Is It

Data Quality Monitor enables advertisers to continuously monitor the quality of their data in the cloud. It provides simple "rules" that you can configure to check that your data is the right type, format, and value. The deployment is simplified, and it can scale to innumerable tables. It is a Python application that runs on the Google Cloud Platform.

Outcome examples

- Discover and fix data quality issues related to bidding and reporting
- Discover and fix reporting discrepancies in global reporting
- Resolves data integration barriers for customer concerned about exposing incorrect or low-quality data.



Prerequisites

- Data to monitor (on BigQuery)
- Access to GCP project
- Budget for GCP usage

Resources

Client

- Cloud Software Engineer for deployment
- Business Intelligence team for use case selection, visualisation, and activation

Google

- Customer Solutions Engineer
- Engagement Manager

Milestones & Timeline

- Deployment (~1 hr, GCP access) - code
- Use cases selection
- Activation setup

Google Data Quality Monitor

- Data Freshness
- KPI monitoring
- Model drift
- Log visualization

Google Data Quality Monitor

- Search Engine Results quality
- Google Maps data quality
- Google Ads
- YouTube Content

Google Data Quality Monitor

imp

1. Simple Freshness Calculation

One of the most common ways to calculate freshness is:

$$\text{Freshness} = \text{Current Time} - \text{Last Updated Time}$$

where:

- **Current Time** = The time when the freshness is being checked.
- **Last Updated Time** = The timestamp when the data was last updated.

This gives freshness in seconds, minutes, hours, or days.

Google Data Quality Monitor

2. Freshness Score (Normalized)

If you need a score to compare different datasets, normalize the freshness to a scale (e.g., 0 to 1):

imp

$$\text{Freshness Score} = 1 - \frac{\text{Current Time} - \text{Last Updated Time}}{\text{Maximum Acceptable Age}}$$

- If **Freshness Score = 1**, the data is perfectly fresh.
- If **Freshness Score = 0**, the data is outdated beyond the acceptable limit.

If the calculated value is negative, you can cap it at 0.

Pandas – Profiling

- ydata-profiling can be used for a quick Exploratory Data Analysis on time-series data. This is useful for a quick understanding on the behaviour of time dependent variables regarding behaviours such as time plots, seasonality, trends, stationary and data gaps.

- Combined with the profiling reports compare, you're able to compare the evolution and data behaviour through time, in terms of time-series specific statistics such as PACF and ACF plots. It also provides the identification of gaps in the time series, caused either by missing values or by entries missing in the time index.

Pandas – Profiling

Stationarity

- In the realm of time-series analysis, a stationary time-series is a dataset where statistical properties, such as mean, variance, and autocorrelation, remain constant over time.

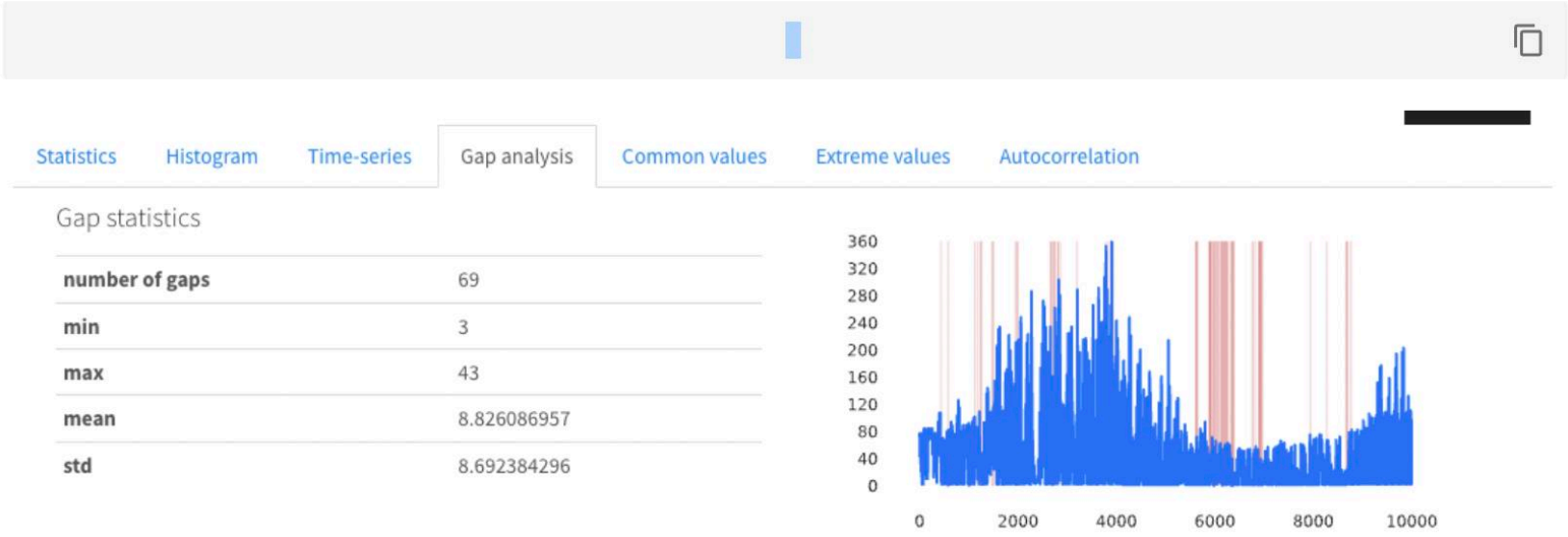
- Seasonality

- Time-series missing gaps

Pandas – Time Series Profiling

```
import pandas as pd

from ydata_profiling.utils.cache import cache_file from ydata_profiling import
ProfileReport

file_name = cache_file(  "pollution_us_2000_2016.csv",
"https://query.data.world/s/mz5ot3l4zrgvldncfgxu34nda45kvb",)

df = pd.read_csv(file_name, index_col=[0])

# Filtering time-series to profile a single site
site = df[df["Site Num"] == 3003]

#Enable tsmode to True to automatically identify time-series variables
#Provide the column name that provides the chronological order of your time-series
profile = ProfileReport(df, tsmode=True, sortby="Date Local", title="Time-Series EDA")

profile.to_file("report_timeseries.html")
```

Pandas – Time Series Profiling



Time-series missing data visualization

Great Expectations

Great Expectations Overview

- The creators of GX were on the money. They built the tool on an expectation (of data quality) that can be tested by running pre-defined and templated tests by connecting to your data sources. In the official integration guides, find more about GX integrations with tools and platforms like Databricks, Flyte, Prefect, and EMR.

- Great Expectation is actively maintained and is known to be used by Vimeo, Calm, ING, Glovo, Avito, DeliveryHero, Atlan, and Heineken, among others.

Great Expectations Features

- GX has an exhaustive list of Expectations that prescribe the "expected state of the data." GX's integrations with the data sources mean that all the data quality checks are done in place, and no data is moved out of the data source.
- GX also supports data contracts by automating data quality checks, recording the results over time, and giving you a human-readable summary of the test runs.
- On top of data sources, such as databases and data warehouses, GX also directly connects with source metadata aggregators and data catalogs, and orchestration engines, such as Airflow, Meltano, and Dagster.
- GX is flexible with storage backends, i.e., you can store Expectations, Validation Results, and Metrics in AWS S3, Azure Blob Storage, Google Cloud Storage, PostgreSQL, or a file system.

Great Expectations Features

- A data contract outlines how data can get exchanged between two parties. It defines the structure, format, and rules of exchange in a distributed data architecture. These formal agreements make sure that there aren't any uncertainties or undocumented assumptions about data.

- What is inside a data contract?
  - In addition to general agreements about intended use, ownership, and provenance, data contracts include agreements about:
  - Schema
  - Semantics
  - Service level agreements (SLA)
  - Metadata (data governance)

Great Expectations Features

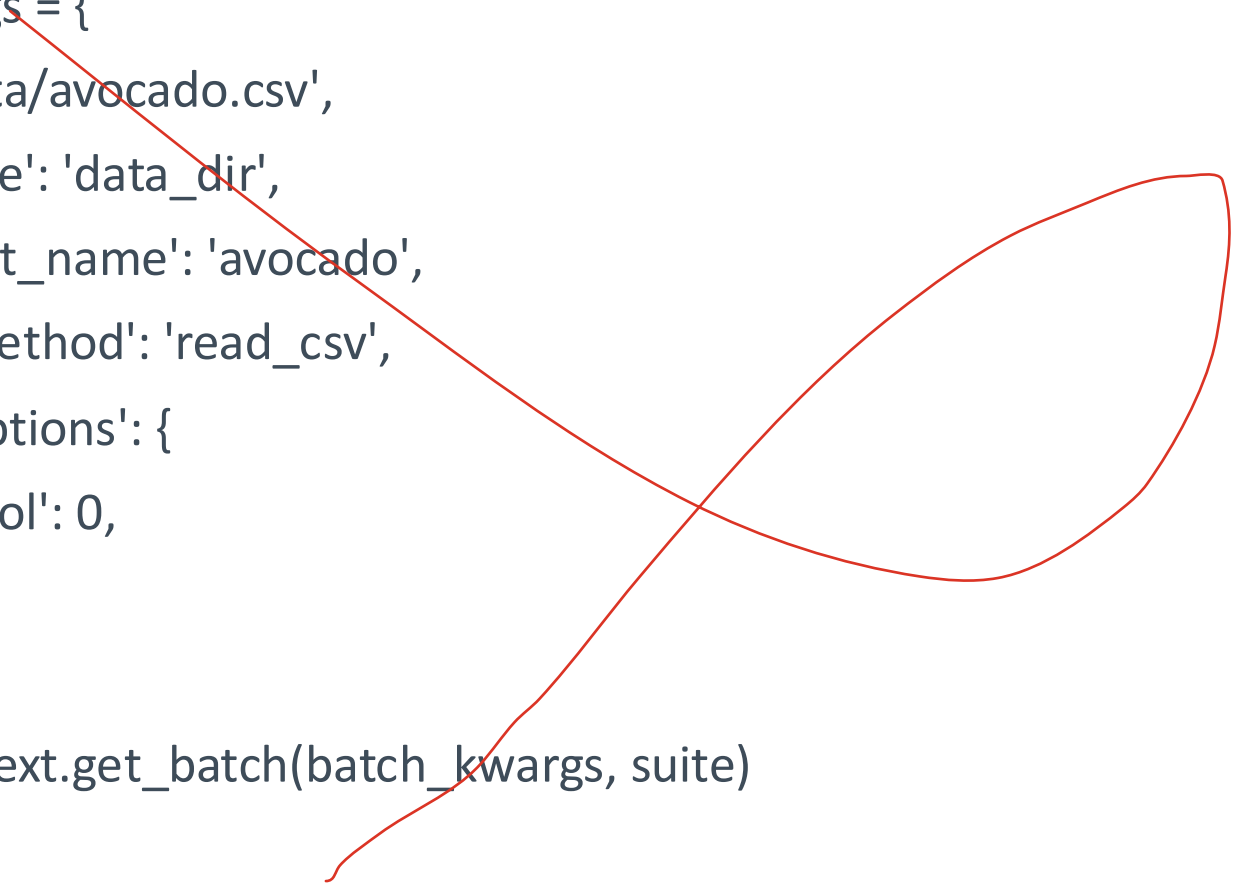
- %%bash
- if [[! -d great_expectations]]
- then
- git init
- git remote add origin https://github.com/datarootsio/tutorial-great-expectations.git
- git pull origin main
- pip install great_expectations==0.13
- apt-get install tree
- mkdir -p great_expectations/checkpoints
- Fi

- import great_expectations as ge
- context = ge.data_context.DataContext()

Great Expectations Features

- suite = context.create_expectation_suite('check_avocado_data', overwrite_existing=True)

- batch_kwargs = {
- 'path': 'data/avocado.csv',
- 'datasource': 'data_dir',
- 'data_asset_name': 'avocado',
- 'reader_method': 'read_csv',
- 'reader_options': {
- 'index_col': 0,
- }
- }
- batch = context.get_batch(batch_kwargs, suite)



Great Expectations Features

- This is the documentation that came with the data:

- Date - The date of the observation
- AveragePrice - the average price of a single avocado
- type - agriculture type: conventional or organic
- Region - the city or region of the observation
- Total Volume - Total number of avocados sold
- 4046 - Total number of avocados with PLU 4046 sold (small Hass)
- 4225 - Total number of avocados with PLU 4225 sold (large Hass)
- 4770 - Total number of avocados with PLU 4770 sold (extra large Hass)

Metrics Computation

- Completeness: Measures the percentage of non-null values for a given column or set of columns.
- b. Uniqueness: Determines the percentage of unique values for a specific column or set of columns, helping to identify potential duplicate entries.
- c. Distinctness: Evaluates the percentage of distinct values among all the rows in a dataset.
- d. Approximate Count Distinct: Calculates an approximate count of distinct values in a column, which is useful for large datasets where an exact count might be computationally expensive.

Metrics Computation

- e. Entropy: Measures the uncertainty or randomness in a column, indicating how evenly distributed the values are.
- f. Mutual Information: Measures the degree of dependency between two columns, providing insights into the relationship between them.
- g. Functional Dependency: Determining if one column functionally depends on another, indicating potential data quality issues.
- h. Compliance: Checks if the data complies with predefined business rules and constraints.

Agenda for today class

- Last Class
  - PqDeequ
  - Panda Yprofiling
  - Google Data Quality
- Today's class
  - Great Expectations
  - Cleanlab
  - Machine Learning

Data Validation

- **Great Expectations**
  - **Great Expectations (GX) is a Python-based open-source tool for managing data quality. It provides data teams with the ability to profile, test, and create reports on data.**

    - **pip install great_expectations**
    - **import great_expectations as gx**
    - **data = gx.read_csv(”data.csv")**
    - **df = gx.from_pandas(data)**

Data quality monitoring Techniques

- **Expectations are declarative statements about your data: the assumptions you have based on previous knowledge of the dataset or your domain knowledge.**

- **Missing values, unique values, and types**
  - **expect_column_values_to_be_unique**
  - **expect_column_values_to_not_be_null**
  - **expect_column_values_to_be_of_type**
  - **expect_column_values_to_be_in_type_list**

Data quality monitoring Techniques

- **Sets and ranges**
  - **expect_column_values_to_be_in_set**
  - **expect_column_values_to_not_be_in_set**
  - **expect_column_values_to_be_between**
  - **expect_column_values_to_be_increasing**
  - **expect_column_values_to_be_decreasing**

- **String matching**
  - **expect_column_value_lengths_to_be_between**
  - **expect_column_value_lengths_to_equal**
  - **expect_column_values_to_match_regex**
  - **expect_column_values_to_not_match_regex**
  - **expect_column_values_to_match_regex_list**
  - **expect_column_values_to_not_match_regex_list**

Trifacta is Now Alteryx Designer Cloud

- **Trifacta is a data-wrangling platform that allows users to discover, prepare, and pipeline data for analytics and machine learning**

Open Refine

OpenRefine is a Java-based power tool that allows you to load data, understand it, clean it up, reconcile it, and augment it with data coming from the web. All from a web browser and the comfort and privacy of your own computer.

- https://github.com/OpenRefine

Data Cleansing

- **cleanlab Python Library**
  - **to quickly identify dataset problems in any Machine Learning project.**
  - ***image,*** **text, audio, tabular data**

- **cleanlab package offers a data-centric AI platform to run many such algorithms and detect common problems in ML datasets like:**
  - **mislabeling**
  - **outliers**
  - **(near) duplicates**
  - **drift**

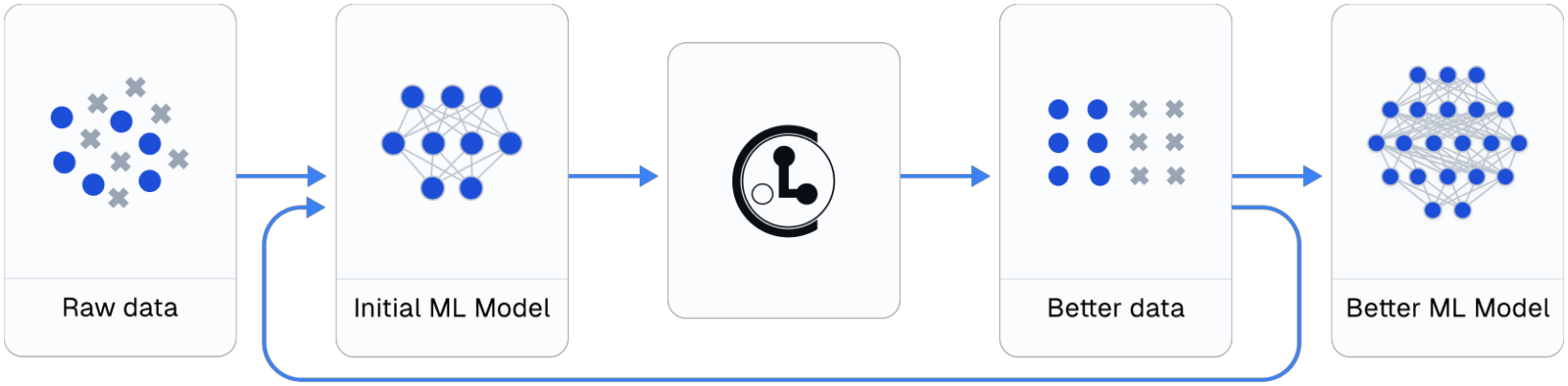
Data Cleansing

- **Flag Null Values: particularly flagging rows with entirely missing observations) which are a common source of trouble for ML models.**

- **Be aware of Imbalanced Classes**
  - **classification dataset is imbalanced**

- **Discover Underperforming Group**
  - **These may correspond to underrepresented groups in the dataset, or subpopulations not well-represented by given feature values.**

Data Cleansing



Machine Learning techniques

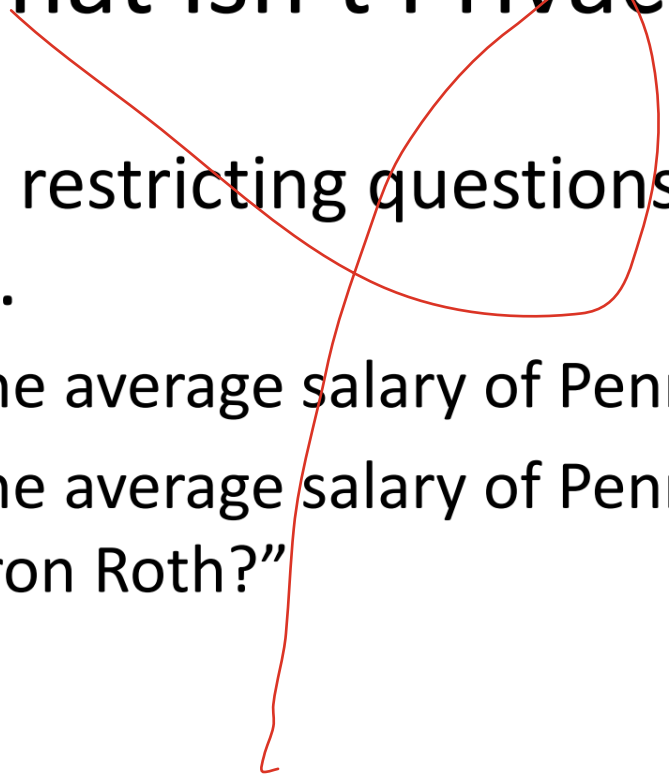
- **Confident Learning: Estimating Uncertainty in Dataset Labels**
  - **Automated Diagnosis of Mislabeled Images in Object Detection Data**
- **Detecting Errors in Numerical Data via a Regression Model**

Data Privacy Outline

- Data Privacy definition
- Need for privacy
- Review of some of algorithms
- K-Anonymity, differential data privacy ?

What Isn't Privacy?

- Privacy isn't restricting questions to large populations.
  - "What is the average salary of Penn faculty?"
  - "What is the average salary of Penn faculty not named Aaron Roth?"



Legal World Views on Privacy

United States: “Privacy is the right to be left alone” - Justice Louis Brandeis

UK: “the right of an individual to be protected against intrusion into his personal life or affairs by direct physical means or by publication of information

Australia: “Privacy is a basic human right and the reasonable expectation of every person”

Recognition of Need for Privacy Guarantees

- By individuals [Cran *et al. '99*]
  - 99% unwilling to reveal their SSN
  - 18% unwilling to reveal their… favorite TV show
- By businesses
  - Online consumers worrying about revealing personal data held back $15 billion in online revenue
- By Federal government
  - Privacy Act of 1974 for Federal agencies
  - Health Insurance Portability and Accountability Act of 1996 (HIPAA)

Technical Privacy Controls

- The risk of reidentification (a threat to anonymity)
  - Types of data in statistical records:
    - Identity data - e.g., name, address, personal number
    - Demographic data - e.g., sex, age, nationality
    - Analysis data - e.g., diseases, habits
  - The degree of anonymity of statistical data depends on:
    - Database size
    - The entropy of the demographic data attributes that can serve as supplementary knowledge for an attacker
  - The entropy of the demographic data attributes depends on:
    - The number of attributes
    - The number of possible values of each attribute
    - Frequency distribution of the values
    - Dependencies between attributes

What is Data Privacy

- Data privacy generally means the ability of a person to determine for themselves when, how, and to what extent personal information about them is shared with or communicated to others.

- This personal information can be one's name, location, contact information, or online or real-world behavior. Just as someone may wish to exclude people from a private conversation, many online users want to control or prevent certain types of personal data collection.

Data Privacy

- Definition of Data Privacy

  - Data privacy associated with personally identifiable information (PII), such as names, addresses, Social Security numbers and credit card numbers. This idea also extends to other valuable or confidential data, including financial data, intellectual property and personal health information.

Need for Data Privacy

- Every time we use a service, we have to hand over some of the personal information
- Even without our knowledge some information is generated and captured by companies that we are likely to have never interacted with

What Isn't Privacy?

- Privacy isn't "Anonymization"
  - Anonymization isn't enough
    - Collection of medical records from a specific urgent care center and date might correspond to only a small collection of medical conditions.
    - Knowledge (from a neighbor?) that Alice went to that urgent care center doesn't identify her record, but implies she has one of a small number of conditions.

Notable Privacy Failure #1: Mass. Grp Insurance (90s)

- Group Insurance Commission published info for medical researchers (left circle)
- Sweeney purchased voter registration info from local government (right circle)
- "87% of the U.S. Population are uniquely identified by (date of birth, gender, ZIP)."

Latanya Sweeney

Source: Wikipedia

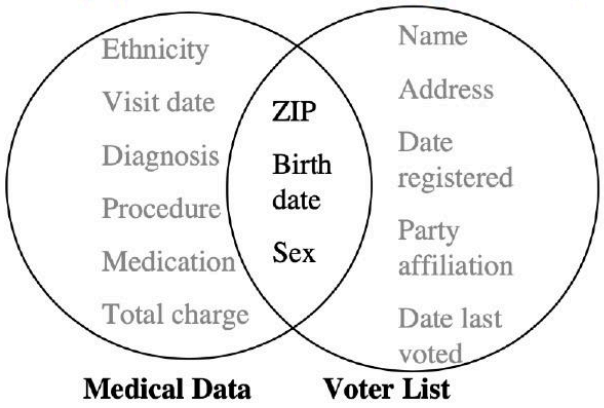
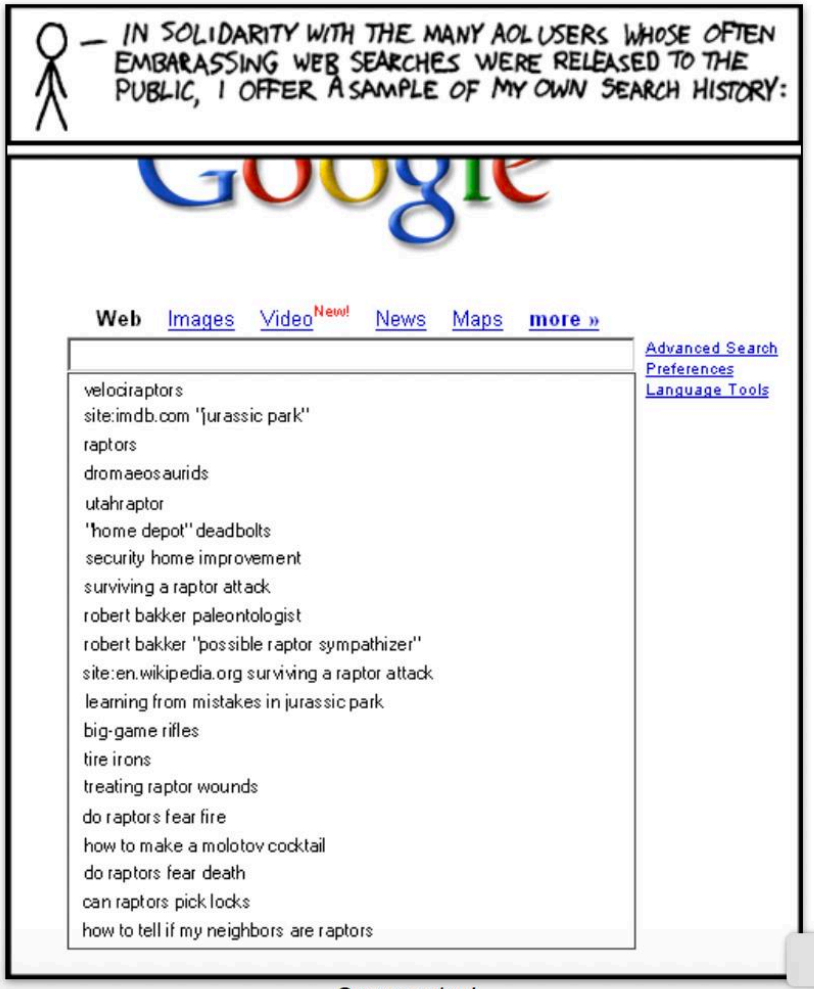


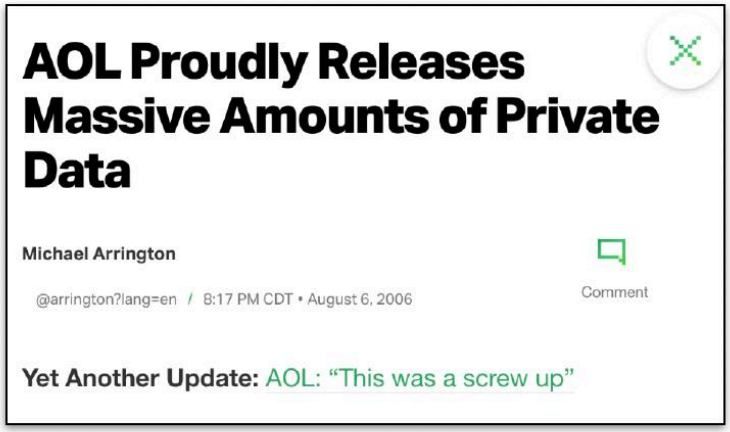
Figure 1 Linking to re-identify data

Source: L. Sweeney. k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10 (5), 2002; 557-570.

Notable Privacy Failure #2: AOL (2006)

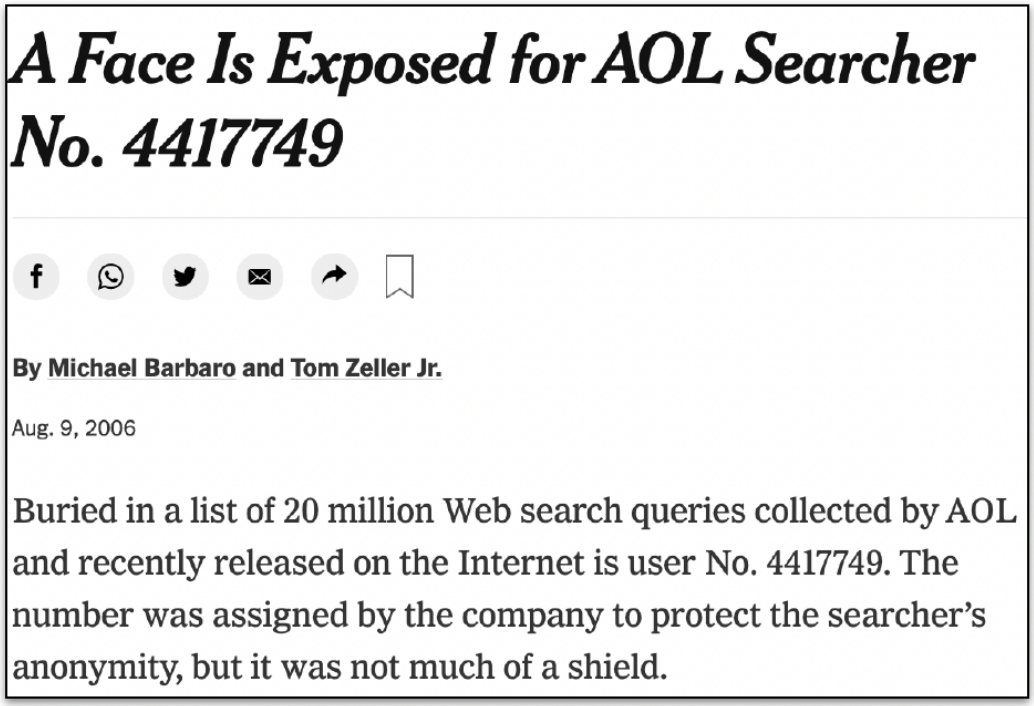
- AOL publishes 20M search queries from 650k users.
- Names deleted, but query histories still associated with individuals





Source: xkcd

Notable Privacy Failure #2: AOL (2006)



A Face Is Exposed for AOL Searcher No. 4417749

By Michael Barbaro and Tom Zeller Jr.

Aug. 9, 2006

Buried in a list of 20 million Web search queries collected by AOL and recently released on the Internet is user No. 4417749. The number was assigned by the company to protect the searcher's anonymity, but it was not much of a shield.

- Several individuals were identified. How would you guess?

Notable Privacy Failure #3: Netflix Prize (2006-2009)

- 2006: Netflix publishes movie rating data of 480K users
  - Meant to be used for recommendation system research
- Q from their FAQ: *"Is there any customer information in the dataset that should be kept private?"*
- *Netflix's answer:*

  *"**No, all customer identifying information has been removed; all that remains are ratings and dates.** This follows our privacy policy, which you can review here. Even if, for example, you knew all your own ratings and their dates you probably couldn't identify them reliably in the data because only a small sample was included (less than one-tenth of our complete dataset) and that data was subject to perturbation. **Of course, since you know all your own ratings that really isn't a privacy problem is it?"***

Notable Privacy Failure #3: Netflix Prize (2006-2009)

name	*Star Wars*	*Casablanca*	*Jurassic Park*	*<other movie>*
Fatma	★★★★, 2/22/99	★★, 7/7/04	★, 8/17/03	★★★★★, 8/22/00
Hong	★★, 5/6/02	★★★★★, 8/9/00	★★★, 6/16/03	★, 3/13/02
Roger	★★★★★, 4/29/98	★, 12/31/99	★★★★, 5/22/95	★, 4/29/00

- Idea: Cross-reference with IMDB
- Arvind+Vitaly: Knowing 8 ratings (w/dates) identifies 90% of users
- People rated movies on Netflix that they did not rate on IMDB.

Robust De-anonymization of Large Sparse Datasets

Arvind Narayanan and Vitaly Shmatikov
The University of Texas at Austin

Source: Wikipedia

RYAN SINGEL SECURITY 03.12.2010 02:48 PM

NetFlix Cancels Recommendation Contest After Privacy Lawsuit

Netflix is canceling its second $1 million Netflix Prize to settle a legal challenge that it breached customer privacy as part of the first contest's race for a better movie-recommendation engine. Friday's announcement came five months after Netflix had announced a successor to its algorithm-improvement contest. The company at the time said it intended to [...]

Notable Privacy Failure #4: NYC Taxi Data (2014)

- NYC releases "anonymized" records of 173M taxi trips to researcher in response to Freedom of Information Act request
- Included start end location and time

10-02-14 | FAST FEED

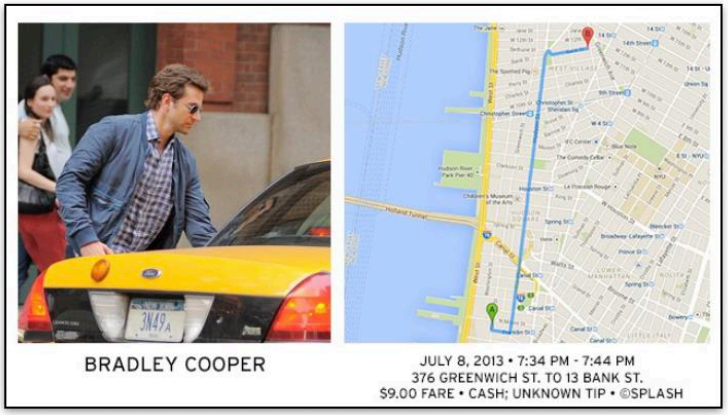
NYC Taxi Data Blunder Reveals Which Celebs Don't Tip–And Who Frequents Strip Clubs

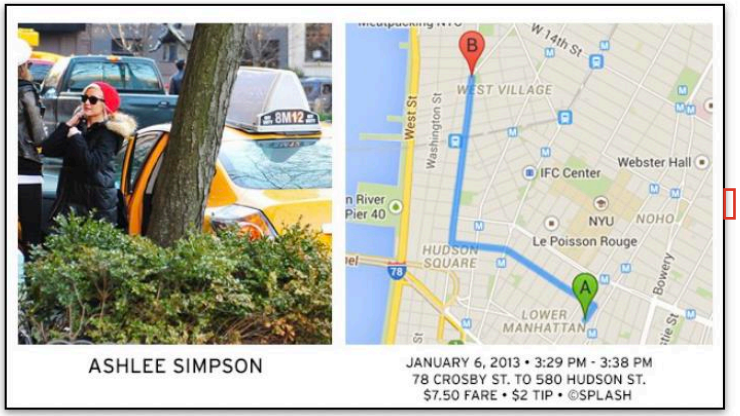
By cross-referencing de-anonymized trip data with paparazzi photos, a privacy research could tell how much Bradley Cooper paid his driver.

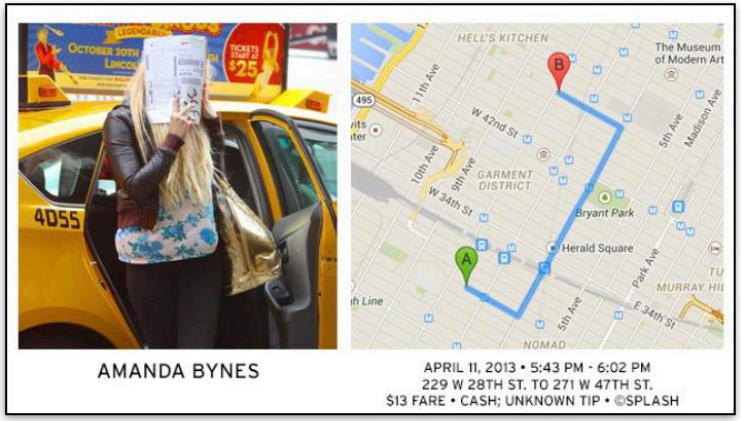
By matching time and location from public photos (like paparazzi photos) to taxi data, people could figure out:

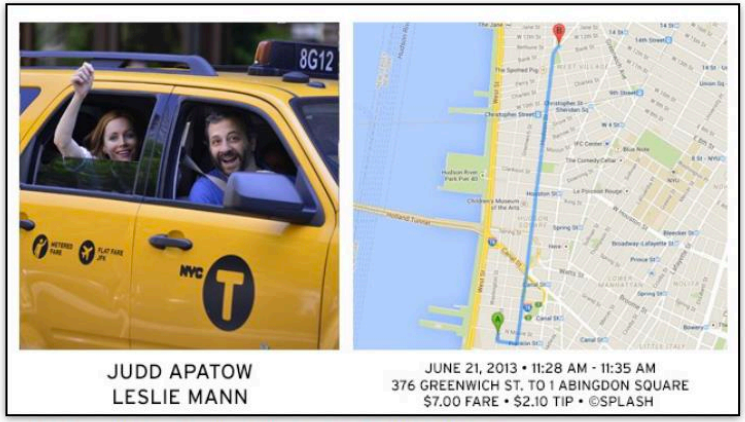
Where someone was picked up
Where they went
How much they paid
When they traveled

Notable Privacy Failure #4: NYC Taxi Data (2014)









Source: https://gawker.com/the-public-nyc-taxicab-database-that-accidentally-track-1646724546

- Also: Dataset had taxi ID replaced with md5(taxiID)…

Privacy Failures: Why is this so hard?

- Hard to anticipate how individuals might be harmed
- Hard to anticipate what *side information* is available for linking
- Hard to anticipate what adversarial analysis is capable of

Latanya Sweeney

Source: Wikipedia

- Sweeney: Take a principled approach!

1. Give precise *definition* of "sufficiently sanitized" data
2. Design sanitization methods that output data meeting definition.

Technical Privacy Controls

- The risk of reidentification (a threat to anonymity)
  - Types of data in statistical records:
    - Identity data - e.g., name, address, personal number
    - Demographic data - e.g., sex, age, nationality
    - Analysis data - e.g., diseases, habits
  - The degree of anonymity of statistical data depends on:
    - Database size
    - The entropy of the demographic data attributes that can serve as supplementary knowledge for an attacker
  - The entropy of the demographic data attributes depends on:
    - The number of attributes
    - The number of possible values of each attribute
    - Frequency distribution of the values
    - Dependencies between attributes

Privacy Metrics

- Problem
  - How to determine that certain degree of data privacy is provided?

- Challenges
  - Different privacy-preserving techniques or systems claim different degrees of data privacy
  - Metrics are usually ad hoc and customized
    - Customized for a user model
    - Customized for a specific technique/system
  - Need to develop uniform privacy metrics
    - To confidently compare different techniques/systems

Data Privacy Outline

Federated Learning :

Instead of sending your private data to Google, Apple, or anyone else,
the model is sent to your device, learns from your data locally, and then only the model updates are sent back — NOT your data

- What does "anonymized data" really mean? How do I actually anonymize data?
- How does federated learning and analysis work?
- Homomorphic encryption sounds great, but is it ready for use?

  You can compute on the encrypted data (ciphertext) and the result, when decrypted, is the same as if you had computed on the original plaintext!

Data Privacy Outline

- How do I compare and choose the best privacy-preserving technologies and methods? Are there open-source libraries that can help?
- What do privacy regulations like GDPR and CCPA mean for my data workflows and data science use cases?
- How do I work with governance and infosec teams to implement internal policies appropriate

State of Data Privacy in 2015

- According to a recent survey by Dimensional Research, 93% of businesses are challenged by data privacy.

- It estimated that by 2018, more than 9 billion U.S. dollars will be lost due to payment card fraud, 6.4 billion due to CNP (card not present) transactions.

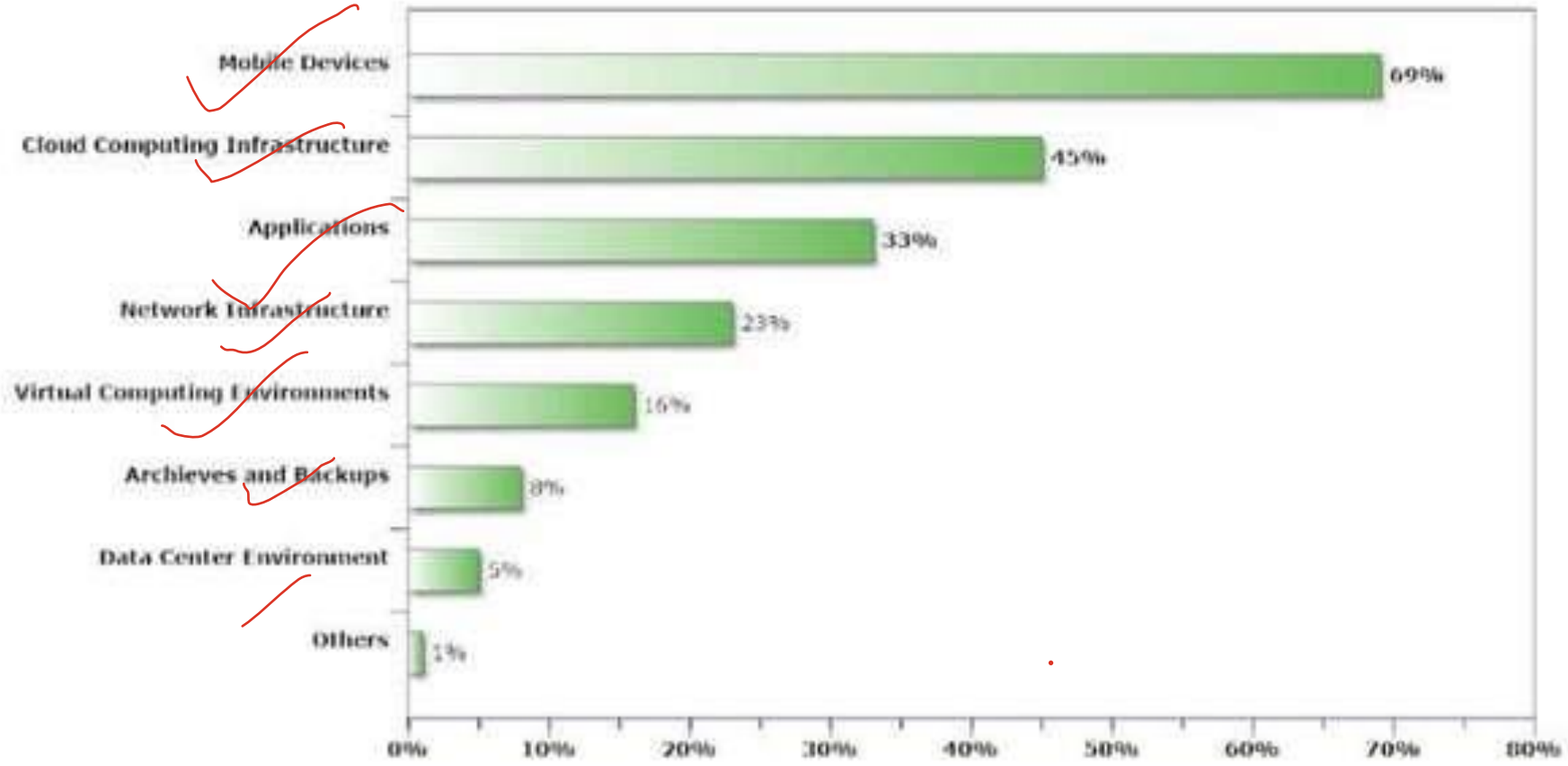


Figure 1l. Percentage data protection risk on regulated data [3]

State of Data Privacy in 2015

- Another increasing worry in the online medium is malicious use of personal information intended to humiliate, harass or in other ways damage someone's reputation.

- Especially among youth, internet bullying is one of the biggest fear parents have when it comes to their children's online safety.

Data Security Vs. Data Privacy

- Data security is commonly referred to as the confidentiality, availability, and integrity of data.
- Data privacy is suitably defined as the appropriate use of data.
- When companies and merchants use data or information that is provided or entrusted to them, the data should be used according to the agreed purposes.
- Companies need to enact a data security policy for the sole purpose of ensuring data privacy or the privacy of their consumers' information.

Data Security Vs. Data Privacy

- Companies must ensure data privacy because the information is an asset to the company.
- A data security policy is simply the means to the desired end, which is data privacy.
- No data security policy can overcome the willing sell or soliciting of the consumer data that was entrusted to an organization.