Data Validation at Scale: Managing Data Quality in Complex Data Pipelines

Data Analytics & Science
Data Validation at Scale

Managing Data Quality in Complex Data Pipelines

code.talks 2023, Hamburg, Germany
September 14, 2023
Wolle

Material Available at https://wolle.science
Research:
• Stream Processing
• Real-Time Databases
• NoSQL & Cloud Systems
• ...

Practice:
• Web Caching
• Big Data Analytics
• Anger Management
• ...

I Am Wolle
The Importance of Data Validation
Where is data validation integrated into data science pipelines and what is its impact?

Data Quality & Constraints
What dimensions of data quality are there and how can they be ensured?

Scalability-Related Challenges
Why is data validation difficult in data-intensive domains?
Running **Example**: Web Performance Analysis Pipeline

Collection | Ingestion | Analytics | Reporting

- Tracking (RUM)
- Performance Dashboard
- QA Dashboard
- Real-Time Alerting
- Ad-hoc SQL Interface
- Custom Reporting

Felix Gessert. *Batching Was Yesterday: Real-Time Tracking & Analysis For 100+ Million Visitors*, code.talks (2023)

Running Example: Web Performance Analysis Pipeline

Collection → Ingestion → Analytics → Reporting

- Tracking (RUM)
- Flink
- SQL Interface
- Performance Dashboard
- QA Dashboard
- Real-Time Alerting
- Ad-hoc SQL Interface
- Custom Reporting

Running Example: Web Performance Analysis Pipeline

Running **Example**: Web Performance Analysis Pipeline

1. Tracking (RUM)
2. Cloud
3. Database
4. Analysis
5. Insights
Running **Example**: Web Performance Analysis Pipeline

- **Collection**
- **Ingestion**
- **Processing & Storage**
- **Analysis, Reporting & ML**

**Dashboarding**

**Data Warehousing**

**Machine Learning**
• **Goal**: verify that data in the pipeline is in an acceptable state for downstream processing, e.g.
  - External reporting (statistics, visualizations & dashboarding)
  - Internal reporting (debugging, product optimizations)
  - Decision-making (analytics, machine learning)
• **Data validation can be integrated in and between all stages**
There are different **dimensions of data quality**, especially:

- **Completeness**: Do we have all the data we need to assess page load performance?
- **Consistency**: Does data have a valid format and does it comply with business semantics?
- **Accuracy**: Do data items represent their corresponding real-world entities well?
- **Uniqueness**: Are duplicate records known and are all unique attributes actually distinct?
There are different **dimensions of data quality**, especially:

- **Completeness**: Do we have all the data we need to assess page load performance?
- **Consistency**: Does data have a valid format and does it comply with business semantics?
- **Accuracy**: Do data items represent their corresponding real-world entities well?
- **Uniqueness**: Are duplicate records known and are all unique attributes actually distinct?
• Is beacon loss a problem at all?
• **When** is beacon loss a problem?
  → For which beacon types? For which beacon strategies?
• **Where** is beacon loss a problem?
  → For which browsers? For which device types?

Data sometimes just gets lost!
Unload Beacon Reliability by Strategy

available on all platforms

experimental feature
(only available as origin trial in Chrome)

```javascript
addEventListener("unload", (event) => {
    navigator.sendBeacon(url, JSON.stringify(data));
});
```

```javascript
addEventListener("beforeunload", (event) => {
    navigator.sendBeacon(url, JSON.stringify(data));
});
```

```javascript
addEventListener("pagehide", (event) => {
    navigator.sendBeacon(url, JSON.stringify(data));
});
```

```javascript
addEventListener("visibilitychange", (event) => {
    if (document.visibilityState === 'hidden') {
        navigator.sendBeacon(url, JSON.stringify(data));
    }
});
```

```javascript
var beacon = new window.PendingPostBeacon(
    url,
    {
        timeout: 60000,
        backgroundTimeout: 0
    }
);
```

Unload Beacon Reliability by Strategy

```javascript
addEventListener("unload", (event) => {
  navigator.sendBeacon(url, JSON.stringify(data));
});

addEventListener("beforeunload", () => {
  navigator.sendBeacon(url, JSON.stringify(data));
});

addEventListener("pagehide", () => {
  navigator.sendBeacon(url, JSON.stringify(data));
});

if (document.visibilityState === "hidden") {
  navigator.sendBeacon(url, JSON.stringify(data));
}
beacon.setData(JSON.stringify(data));
```

Available on all platforms

---

**Thursday, 14:00**
(Data Analytics & Science)

Sophie Ferrlein. *Data Viz for Engineers: Optimizing Insight & Decision Making Through Visualization*, code.talks (2023)
Unload Beacon **Reliability** by Strategy & Device

**MOBILE**

- **PendingPostBeacon**: 97%
- **visibilitychange**: 90%
- **pagehide**: 81%
- **unload**: 78%
- **beforeunload**: 28%

**DESKTOP**

- **PendingPostBeacon**: 99%
- **visibilitychange**: 91%
- **pagehide**: 88%
- **unload**: 87%
- **beforeunload**: 73%

Based on 52 million page views on a globally operating e-commerce site measured by Speed Kits real user-monitoring | August 2023

Deskstop data may be overrepresented!
## Unload Beacon Reliability by Strategy & Browser

<table>
<thead>
<tr>
<th></th>
<th>Chrome</th>
<th>Safari</th>
<th>Firefox</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pending PostBeacon</td>
<td>98%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>visibility change</td>
<td>92%</td>
<td>89%</td>
<td>95%</td>
</tr>
<tr>
<td>pagehide</td>
<td>88%</td>
<td>80%</td>
<td>91%</td>
</tr>
<tr>
<td>unload</td>
<td>86%</td>
<td>80%</td>
<td>90%</td>
</tr>
<tr>
<td>before unload</td>
<td>83%</td>
<td>3%</td>
<td>91%</td>
</tr>
</tbody>
</table>

Based on 52 million page views on a globally operating e-commerce site measured by Speed Kit's real user monitoring | August 2023

Unload Beacon Reliability: The Ideal Combo Strategy

Can be used with Backward-Forward Cache?

- `visibilitychange + pagehide + beforeunload + unload`
  - 91.7%
  - ✗

- `visibilitychange + pagehide + beforeunload`
  - 91.7%
  - ✗

- `visibilitychange + pagehide`
  - 91.3%
  - ✓

Based on 52 million page views on a globally operating e-commerce site measured by Speed Kits real user-monitoring | August 2023

Dimensions of Data Quality: **Consistency**

- There are different **dimensions of data quality**, especially:
  - **Completeness**: Do we have all the data we need to assess page load performance?
  - **Consistency**: Does data have a valid format and does it comply with business semantics?
  - **Accuracy**: Do data items represent their corresponding real-world entities well?
  - **Uniqueness**: Are duplicate records known and are all unique attributes actually distinct?

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Pageload ID</th>
<th>Browser</th>
<th>LCP (Performance)</th>
<th>Session ID</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:05:04.578</td>
<td>37ab08</td>
<td>Edge</td>
<td>&quot;670ms&quot;</td>
<td>123</td>
<td>null</td>
</tr>
<tr>
<td>13:26:48.139</td>
<td>9cddf7</td>
<td>Firefox</td>
<td>692 654</td>
<td>456</td>
<td>abc.de/red</td>
</tr>
<tr>
<td>13:28:23.857</td>
<td>0b577a</td>
<td>Firefox</td>
<td>0.256</td>
<td>456</td>
<td>abc.de/blue</td>
</tr>
<tr>
<td>13:29:17.468</td>
<td>faf55e</td>
<td>Edge</td>
<td>1.598</td>
<td>456</td>
<td>abc.de/sold</td>
</tr>
<tr>
<td>20:45:38.941</td>
<td>faf55e</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>abc.de/sold</td>
</tr>
</tbody>
</table>

**Note**: Browser values should be unified for all records in the same session!
There are different dimensions of data quality, especially:

- **Completeness**: Do we have all the data we need to assess page load performance?
- **Consistency**: Does data have a valid format and does it comply with business semantics?
- **Accuracy**: Do data items represent their corresponding real-world entities well?
- **Uniqueness**: Are duplicate records known and are all unique attributes actually distinct?

**Note**: Browser values should be unified for all records in the same session! → value may be replaced with majority vote (another reasonable option: replace old values with latest one)
Dimensions of Data Quality: **Consistency**

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Pageload ID</th>
<th>Browser</th>
<th>LCP (Performance)</th>
<th>Session ID</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:05:04.578</td>
<td>37ab08</td>
<td>Edge</td>
<td>&quot;670ms&quot;</td>
<td>123</td>
<td>null</td>
</tr>
<tr>
<td>13:26:48.139</td>
<td>9cddf7</td>
<td>Firefox</td>
<td>692 654</td>
<td>456</td>
<td>abc.de/red</td>
</tr>
<tr>
<td>13:28:23.857</td>
<td>0b577a</td>
<td>Firefox</td>
<td>0.256</td>
<td>456</td>
<td>abc.de/blue</td>
</tr>
<tr>
<td>13:29:17.468</td>
<td>faf55e</td>
<td>Firefox</td>
<td>1.598</td>
<td>456</td>
<td>abc.de/sold</td>
</tr>
<tr>
<td>20:45:38.941</td>
<td>faf55e</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>abc.de/sold</td>
</tr>
</tbody>
</table>

**Note:** All values represent milliseconds, but formats differ depending on browser.

- There are different **dimensions of data quality**, especially:
  - **Completeness:** Do we have all the data we need to assess page load performance?
  - **Consistency:** Does data have a valid format and does it comply with business semantics?
  - **Accuracy:** Do data items represent their corresponding real-world entities well?
  - **Uniqueness:** Are duplicate records known and are all unique attributes actually distinct?
Dimensions of Data Quality: **Consistency**

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Pageload ID</th>
<th>Browser</th>
<th>LCP (Performance)</th>
<th>Session ID</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:05:04.578</td>
<td>37ab08</td>
<td>Edge</td>
<td>670</td>
<td>123</td>
<td>null</td>
</tr>
<tr>
<td>13:26:48.139</td>
<td>9cddf7</td>
<td>Firefox</td>
<td>692 654</td>
<td>456</td>
<td>abc.de/red</td>
</tr>
<tr>
<td>13:28:23.857</td>
<td>0b577a</td>
<td>Firefox</td>
<td>256</td>
<td>456</td>
<td>abc.de/blue</td>
</tr>
<tr>
<td>13:29:17.468</td>
<td>faf55e</td>
<td>Firefox</td>
<td>1598</td>
<td>456</td>
<td>abc.de/sold</td>
</tr>
<tr>
<td>20:45:38.941</td>
<td>faf55e</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>abc.de/sold</td>
</tr>
</tbody>
</table>

**Note:** All values represent milliseconds, but formats differ depending on browser. → values may be converted to integer

- There are different **dimensions of data quality**, especially:
  - **Completeness:** Do we have all the data we need to assess page load performance?
  - **Consistency:** Does data have a valid format and does it comply with business semantics?
  - **Accuracy:** Do data items represent their corresponding real-world entities well?
  - **Uniqueness:** Are duplicate records known and are all unique attributes actually distinct?
Dimensions of Data Quality: **Accuracy**

- **Completeness**: Do we have all the data we need to assess page load performance?
- **Consistency**: Does data have a valid format and does it comply with business semantics?
- **Accuracy**: Do data items represent their corresponding real-world entities well?
- **Uniqueness**: Are duplicate records known and are all unique attributes actually distinct?

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Pageload ID</th>
<th>Browser</th>
<th>LCP (Performance)</th>
<th>Session ID</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:05:04.578</td>
<td>37ab08</td>
<td>Edge</td>
<td>670</td>
<td>123</td>
<td>null</td>
</tr>
<tr>
<td>13:26:48.139</td>
<td>9cddf7</td>
<td>Firefox</td>
<td><strong>692 654</strong></td>
<td>456</td>
<td>abc.de/red</td>
</tr>
<tr>
<td>13:28:23.857</td>
<td>0b577a</td>
<td>Firefox</td>
<td>256</td>
<td>456</td>
<td>abc.de/blue</td>
</tr>
<tr>
<td>13:29:17.468</td>
<td>faf55e</td>
<td>Firefox</td>
<td><strong>1598</strong></td>
<td>456</td>
<td>abc.de/sold</td>
</tr>
<tr>
<td>20:45:38.941</td>
<td>faf55e</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>abc.de/sold</td>
</tr>
</tbody>
</table>

**Note**: Despite being in the right format, one value does not represent a reasonable timer value.
Dimensions of Data Quality: **Accuracy**

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Pageload ID</th>
<th>Browser</th>
<th>LCP (Performance)</th>
<th>Session ID</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:05:04.578</td>
<td>37ab08</td>
<td>Edge</td>
<td>670</td>
<td>123</td>
<td>null</td>
</tr>
<tr>
<td>13:26:48.139</td>
<td>9cddf7</td>
<td>Firefox</td>
<td>null</td>
<td>456</td>
<td>abc.de/red</td>
</tr>
<tr>
<td>13:28:23.857</td>
<td>0b577a</td>
<td>Firefox</td>
<td>256</td>
<td>456</td>
<td>abc.de/blue</td>
</tr>
<tr>
<td>13:29:17.468</td>
<td>faf55e</td>
<td>Firefox</td>
<td>1598</td>
<td>456</td>
<td>abc.de/sold</td>
</tr>
<tr>
<td>20:45:38.941</td>
<td>faf55e</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>abc.de/sold</td>
</tr>
</tbody>
</table>

**Note:** Despite being in the right format, one value does not represent a reasonable timer value. → broken value may be removed

- There are different **dimensions of data quality**, especially:
  - **Completeness:** Do we have all the data we need to assess page load performance?
  - **Consistency:** Does data have a valid format and does it comply with business semantics?
  - **Accuracy:** Do data items represent their corresponding real-world entities well?
  - **Uniqueness:** Are duplicate records known and are all unique attributes actually distinct?
There are different dimensions of data quality, especially:

- **Completeness**: Do we have all the data we need to assess page load performance?
- **Consistency**: Does data have a valid format and does it comply with business semantics?
- **Accuracy**: Do data items represent their corresponding real-world entities well?
- **Uniqueness**: Are duplicate records known and are all unique attributes actually distinct?

Note: The ID field should be unique, but two different records share the same value!
There are different dimensions of data quality, especially:

- **Completeness**: Do we have all the data we need to assess page load performance?
- **Consistency**: Does data have a valid format and does it comply with business semantics?
- **Accuracy**: Do data items represent their corresponding real-world entities well?
- **Uniqueness**: Are duplicate records known and are all unique attributes actually distinct?
There are different dimensions of data quality, especially:

- **Completeness**: Do we have all the data we need to assess page load performance?
- **Consistency**: Does data have a valid format and does it comply with business semantics?
- **Accuracy**: Do data items represent their corresponding real-world entities well?
- **Uniqueness**: Are duplicate records known and are all unique attributes actually distinct?

**Note**: The ID field should be unique, but two different records share the same value! → merge duplicates into a single record.
• **Constraints** are rules, conditions, or limits that data must adhere to

• **Type Checks** represent expectations on the data format, e.g.
  - *Value range* for numerical data (e.g. \([0, \text{MAX\_INTEGER})\) for load timers)
  - *Format or pattern* for string-valued data (e.g. ISO 8601 for timestamps)
  - *Structure* for complex attributes (e.g. required keys for JSON objects)

• **Complex conditions** can further describe complex semantics such as
  - Cross-field or cross-record relationships (e.g. same browser within sessions)
  - Referential integrity between records in different collections
  - Custom constraints for domain semantics
import pandas as pd
import pandera as pa
from pandera import Column, DataFrameSchema, Check

# Define the schema
schema = DataFrameSchema(
    {
        "Timestamp": Column(pa.DateTime),
        "Pageload ID": Column(pa.String,
            Check(lambda x: x.str.len() == 8)),
        "Browser": Column(pa.String,
            Check(lambda x: x.isin(["Chrome", "Edge", "Firefox"])),
        "LCP": Column(pa.Int,
            Check(lambda x: (x >= 0) & (x <= 600000)),
        "Session ID": Column(pa.Int),
    }
)
# valid data item

good = {
    "Timestamp": [pd.Timestamp("2023-05-10 13:26:48.139")],
    "PageLoad ID": ["9cddf7"],
    "Browser": ["Firefox"],
    "LCP": [256],
    "Session ID": [456],
}

# invalid data item

bad = {
    "Timestamp": [pd.Timestamp("2023-05-10 09:05:04.578")],
    "PageLoad ID": ["37ab08"],
    "Browser": ["Edge"],
    "LCP": [692654],  # timer value out of bounds
    "Session ID": [123],
}
for record in [good, bad]:
    record_id = record['Pageload ID'][0]
    try:
        validated = schema(pd.DataFrame(record))
        print(f"\nValidation passed for record {record_id}!"")
    except pa.errors.SchemaError as e:
        print(f"\nValidation FAILED for record {record_id}:")
        print(e)
One Month in Data Errors at Baqend: April 2023

**Fundamental Challenge:** Scalability

- **Data Errors by Type**
  - Aggregation (66.80%)
  - Attribute (32.83%)
  - MissingValue (63.91%)
  - Internal (0.16%)

- **Data Errors by Cause**
  - MissingValue (63.91%)
  - ValueOutOfRange (8.60%)
  - BlankValue (11.92%)
  - AmbiguousValue (7.98%)
  - StringLengthExceeded (4.30%)

436M errors
• Manual constraint definition is often infeasible, because of ...
  o ... inherent data complexity (often hundreds of attributes)
  o ... aggregation, derived storage, and evolving schemas
  o ... a plethora of other data stores to integrate!

→ Automation is necessary!
Computing validation metrics from scratch periodically can be infeasible, because of …

- … strict timing requirements
- … efficiency or cost reasons
- … data privacy reasons

→ Incremental computation can be the only option!

Details in Felix’ talk!
Specifying generalized constraints can be difficult in large deployments, because of:

- ... temporal fluctuations (e.g. throughout the day, on black Friday, or during holydays)
- ... multi-tenancy (e.g. different data patterns by customer timezone or domain)

Elasticity & Multi-Tenancy requirements can be challenging!
Deployments can break things (and cause sudden load spikes for validation)

- **Availability**
  Can this take down your data pipeline?
- **Attribution**
  Who or what is causing the problem?
- **Responsibility**
  Can you fix the problem?
So How Do You Handle All This?

• Advanced Techniques
  o Inferring constraints
  o Adapting to schema changes
  o Incremental computation of complex measures

• Tooling & Frameworks
  o Validation libraries such as Great Expectations, Pandera, TFDV, or Deequ
  o Preprocessing and validation with Apache Spark and Apache Flink

• Further Challenges
  o Handling distribution (validation per partitioning, avoiding skew, ...)
  o Efficiency and performance (load distribution, approximation, ...)
  o Operational challenges (anomaly detection, fixing, load shedding, ...)
• **Data Quality** can be measured along dimensions such as completeness, consistency, accuracy, and uniqueness

• **Constraints** specify expectations about the data and can be used to enforce them

• **Data Validation** is the process of ensuring high data quality for processes like analysis, modeling, and decision-making

• Data Validation **Challenges at Scale** include
  
  o **Complexity**: schemas are often too complex to define constraints manually
  
  o **Volatility**: data varies throughout the day, by season, or by customer
  
  o **Continuity**: incremental processing is required when computation from scratch is infeasible
Thanks! Questions?

Material Available at https://wolle.science