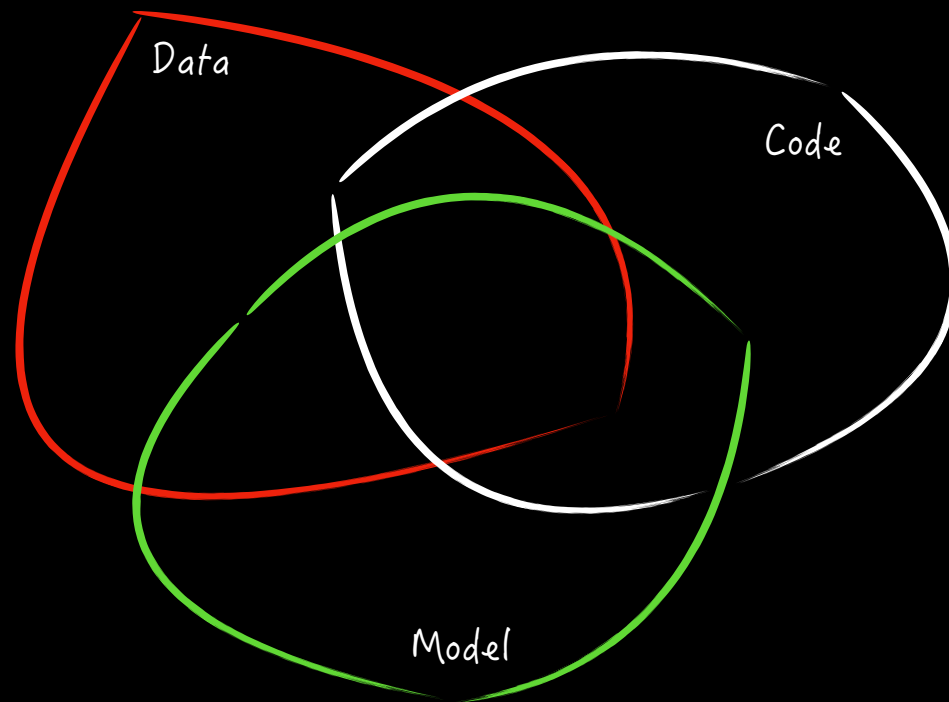


# Data Quality Assessment

Jin Guo    Oct 20th, 2020

# Software Quality Management Process

- Quality planning
- Quality Assurance
- Quality Control
- Process Improvement



# Data Quality

- What makes “good” data?

Contains relevant features

Reflect real interactions

Good coverage

Few or no biases

Large enough

# Dataset Specification and Documentation

- Why is the dataset created? (e.g., is there a specific intended task gap that needed to be filled? Who fund the creation of the dataset?)
- What preprocessing/cleaning is done? (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances)
- If it relates to people, are they told what the dataset would be used for and did they consent? If so, how? Were they provided with any mechanism to revoke their consent in the future or for certain uses?
- Will the dataset be updated? How often, by whom?

# Datasheet

Gebru, Timnit, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. "Datasheets for Datasets."

## Motivation for Dataset Creation

**Why was the dataset created?** (e.g., were there specific tasks in mind, or a specific gap that needed to be filled?)

**What (other) tasks could the dataset be used for?** Are there obvious tasks for which it should *not* be used?

**Has the dataset been used for any tasks already?** If so, where are the results so others can compare (e.g., links to published papers)?

**Who funded the creation of the dataset?** If there is an associated grant, provide the grant number.

**Any other comments?**

## Data Collection Process

**How was the data collected?** (e.g., hardware apparatus/sensor, manual human curation, software program, software interface/API; how were these constructs/measures/methods validated?)

**Who was involved in the data collection process?** (e.g., students, crowdworkers) How were they compensated? (e.g., how much were crowdworkers paid?)

**Over what time-frame was the data collected?** Does the collection time-frame match the creation time-frame?

**How was the data associated with each instance acquired?** Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part of speech tags; model-based guesses for age or language)? If the latter two, were they validated/verified and if so how?

**Does the dataset contain all possible instances?** Or is it, for instance, a sample (not necessarily random) from a larger set of instances?

**If the dataset is a sample, then what is the population?** What was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)? Is the sample representative of the larger set (e.g., geographic coverage)? If not, why not (e.g., to cover a more diverse range of instances)? How does this affect possible uses?

**Is there information missing from the dataset and why?** (this does not include intentionally dropped instances; it might include, e.g., redacted text, withheld documents) Is this data missing because it was unavailable?

**Are there any known errors, sources of noise, or redundancies in the data?**

**Any other comments?**

## Dataset Composition

**What are the instances?** (that is, examples; e.g., documents, images, people, countries) Are there multiple types of instances? (e.g., movies, users, ratings; people, interactions between them; nodes, edges)

**Are relationships between instances made explicit in the data** (e.g., social network links, user/movie ratings, etc.)?

**How many instances of each type are there?**

**What data does each instance consist of?** "Raw" data (e.g., unprocessed text or images)? Features/attributes? Is there a label/target associated with instances? If the instances are related to people, are subpopulations identified (e.g., by age, gender, etc.) and what is their distribution?

**Is everything included or does the data rely on external resources?** (e.g., websites, tweets, datasets) If external resources, a) are there guarantees that they will exist, and remain constant, over time; b) is there an official archival version. Are there licenses, fees or rights associated with any of the data?

**Are there recommended data splits or evaluation measures?** (e.g., training, development, testing; accuracy/AUC)

**What experiments were initially run on this dataset?** Have a summary of those results and, if available, provide the link to a paper with more information here.

**Any other comments?**

## Data Preprocessing

**What preprocessing/cleaning was done?** (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values, etc.)

**Was the “raw” data saved in addition to the preprocessed/cleaned data?** (e.g., to support unanticipated future uses)

**Is the preprocessing software available?**

**Does this dataset collection/processing procedure achieve the motivation for creating the dataset stated in the first section of this datasheet?**

**Any other comments?**

## Dataset Distribution

**How is the dataset distributed?** (e.g., website, API, etc.; does the data have a DOI; is it archived redundantly?)

**When will the dataset be released/first distributed?** (Is there a canonical paper/reference for this dataset?)

**What license (if any) is it distributed under?** Are there any copyrights on the data?

**Are there any fees or access/export restrictions?**

**Any other comments?**

## Dataset Maintenance

**Who is supporting/hosting/maintaining the dataset?** How does one contact the owner/curator/manager of the dataset (e.g. email address, or other contact info)?

**Will the dataset be updated?** How often and by whom? How will updates/revisions be documented and communicated (e.g., mailing list, GitHub)? Is there an erratum?

**If the dataset becomes obsolete how will this be communicated?**

**Is there a repository to link to any/all papers/systems that use this dataset?**

**If others want to extend/augment/build on this dataset, is there a mechanism for them to do so?** If so, is there a process for tracking/assessing the quality of those contributions. What is the process for communicating/distributing these contributions to users?

**Any other comments?**

## Legal & Ethical Considerations

**If the dataset relates to people (e.g., their attributes) or was generated by people, were they informed about the data collection?** (e.g., datasets that collect writing, photos, interactions, transactions, etc.)

**If it relates to other ethically protected subjects, have appropriate obligations been met?** (e.g., medical data might include information collected from animals)

**If it relates to people, were there any ethical review applications/reviews/approvals?** (e.g. Institutional Review Board applications)

**If it relates to people, were they told what the dataset would be used for and did they consent? What community norms exist for data collected from human communications?** If consent was obtained, how? Were the people provided with any mechanism to revoke their consent in the future or for certain uses?

**If it relates to people, could this dataset expose people to harm or legal action?** (e.g., financial social or otherwise) What was done to mitigate or reduce the potential for harm?

**If it relates to people, does it unfairly advantage or disadvantage a particular social group?** In what ways? How was this mitigated?

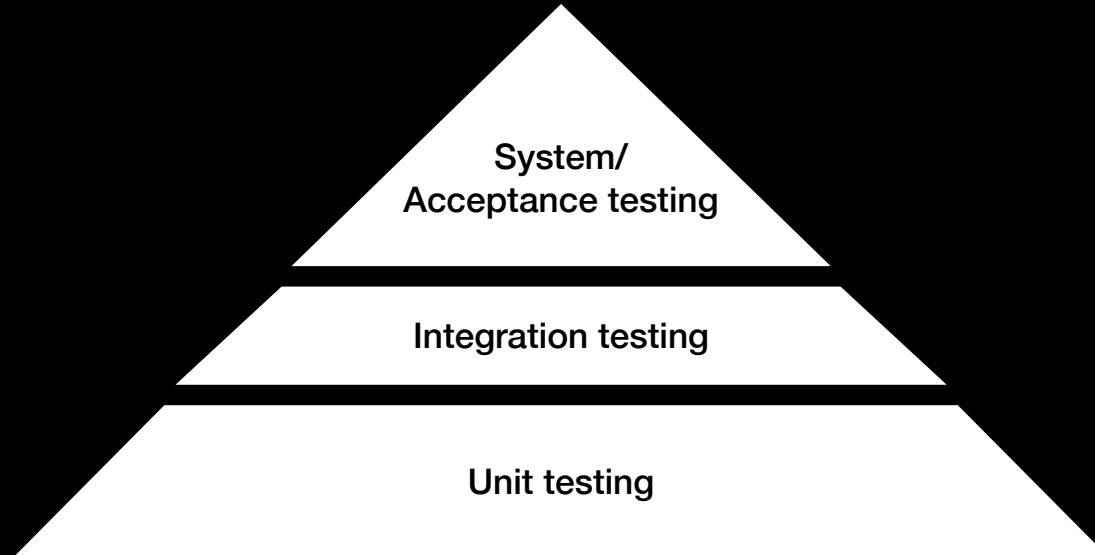
**If it relates to people, were they provided with privacy guarantees?** If so, what guarantees and how are these ensured?

**Does the dataset comply with the EU General Data Protection Regulation (GDPR)?** Does it comply with any other standards, such as the US Equal Employment Opportunity Act?

**Does the dataset contain information that might be considered sensitive or confidential?** (e.g., personally identifying information)

**Does the dataset contain information that might be considered inappropriate or offensive?**

# Testing in Traditional Software Development





# Unit Tests for Data?

- Completeness
- Accuracy
- Uniqueness
- Consistency
- Timeliness

# Data Schema

Scope/Problem		Dirty Data	Reasons/Remarks
<b>Attribute</b>	Illegal values	bdate=30.13.70	values outside of domain range
<b>Record</b>	Violated attribute dependencies	age=22, bdate=12.02.70	age = current year - birth year should hold
<b>Record type</b>	Uniqueness violation	emp <sub>1</sub> =(name="John Smith", SSN="123456"); emp <sub>2</sub> =(name="Peter Miller", SSN="123456")	uniqueness for SSN (social security number) violated
<b>Source</b>	Referential integrity violation	emp=(name="John Smith", deptno=127)	referenced department (127) not defined

Rahm, Erhard, and Hong Hai Do. "Data cleaning: Problems and current approaches."  
*IEEE Data Eng. Bull.* 23, no. 4 (2000): 3-13.

# Instance Constraints

Scope/Problem		Dirty Data	Reasons/Remarks
<b>Attribute</b>	Missing values	phone=9999-999999	unavailable values during data entry (dummy values or null)
	Misspellings	city="Liipzig"	usually typos, phonetic errors
	Cryptic values, Abbreviations	experience="B"; occupation="DB Prog."	
	Embedded values	name="J. Smith 12.02.70 New York"	multiple values entered in one attribute (e.g. in a free-form field)
	Misfielded values	city="Germany"	
<b>Record</b>	Violated attribute dependencies	city="Redmond", zip=77777	city and zip code should correspond
<b>Record type</b>	Word transpositions	name <sub>1</sub> ="J. Smith", name <sub>2</sub> ="Miller P."	usually in a free-form field
	Duplicated records	emp <sub>1</sub> =(name="John Smith",...); emp <sub>2</sub> =(name="J. Smith",...)	same employee represented twice due to some data entry errors
	Contradicting records	emp <sub>1</sub> =(name="John Smith", bdate=12.02.70); emp <sub>2</sub> =(name="John Smith", bdate=12.12.70)	the same real world entity is described by different values
<b>Source</b>	Wrong references	emp=(name="John Smith", deptno=17)	referenced department (17) is defined but wrong

# Verify Constraints - Deequ Tool



constraint	semantic
dimension <i>completeness</i> isComplete hasCompleteness	check that there are no missing values in a column custom validation of the fraction of missing values in a column
dimension <i>consistency</i> isUnique hasUniqueness hasDistinctness isInRange hasConsistentType isNonNegative isLessThan satisfies satisfiesIf hasPredictability	check that there are no duplicates in a column custom validation of the unique value ratio in a column custom validation of the unique row ratio in a column validation of the fraction of values that are in a valid range validation of the largest fraction of values that have the same type validation whether all values in a numeric column are non-negative validation whether values in the 1st column are always less than in the 2nd column validation whether all rows match predicate validation whether all rows matching 1st predicate also match 2nd predicate user-defined validation of the predictability of a column

Schelter, Sebastian, Dustin Lange, Philipp Schmidt, Meltem Celikel, Felix Biessmann, and Andreas Grafberger.  
"Automating large-scale data quality verification." *Proceedings of the VLDB Endowment* 11, no. 12 (2018): 1781-1794.

# Verify Constraints - Deequ Tool



## constraint

statistics

hasSize

hasTypeConsistency

hasCountDistinct

hasApproxCountDistinct

hasMin

hasMax

hasMean

hasStandardDeviation

hasApproxQuantile

hasEntropy

hasMutualInformation

hasHistogramValues

hasCorrelation

time

hasNoAnomalies

## semantic

custom validation of the number of records

custom validation of the maximum fraction of values of the same data type

custom validation of the number of distinct non-null values in a column

custom validation of the approx. number of distinct non-null values

custom validation of a column's minimum value

custom validation of a column's maximum value

custom validation of a column's mean value

custom validation of a column's standard deviation

custom validation of a particular quantile of a column (approx.)

custom validation of a column's entropy

custom validation of a column pair's mutual information

custom validation of column histogram

custom validation of a column pair's correlation

validation of anomalies in time series of metric values

Schelter, Sebastian, Dustin Lange, Philipp Schmidt, Meltem Celikel, Felix Biessmann, and Andreas Grafberger.  
"Automating large-scale data quality verification." *Proceedings of the VLDB Endowment* 11, no. 12 (2018): 1781-1794.

# Verify Constraints - Deequ Tool



```
import com.amazon.deequ.{VerificationSuite, VerificationResult}
import com.amazon.deequ.VerificationResult.checkResultsAsDataFrame
import com.amazon.deequ.checks.{Check, CheckLevel}

val verificationResult: VerificationResult = { VerificationSuite()
  // data to run the verification on
  .onData(dataset)
  // define a data quality check
  .addCheck(
    Check(CheckLevel.Error, "Review Check")
      .hasSize(_ >= 3000000) // at least 3 million rows
      .hasMin("star_rating", _ == 1.0) // min is 1.0
      .hasMax("star_rating", _ == 5.0) // max is 5.0
      .isComplete("review_id") // should never be NULL
      .isUnique("review_id") // should not contain duplicates
      .isComplete("marketplace") // should never be NULL
      // contains only the listed values
      .isContainedIn("marketplace", Array("US", "UK", "DE", "JP", "FR"))
      .isNonNegative("year")) // should not contain negative values
  // compute metrics and verify check conditions
  .run()
}

// convert check results to a Spark data frame
val resultDataFrame = checkResultsAsDataFrame(spark, verificationResult)
```

constraint	constraint_status	constraint_message
SizeConstraint(Size(None))	Success	
MinimumConstraint(Minimum(star_rating,None))	Success	
MaximumConstraint(Maximum(star_rating,None))	Success	
CompletenessConstraint(Completeness(review_id,None))	Success	
UniquenessConstraint(Uniqueness(List(review_id)))	Failure	Value: 0.9926566948782706 does not meet the constraint requirement!
CompletenessConstraint(Completeness(marketplace,None))	Success	
ComplianceConstraint(Compliance(marketplace contained in US,UK,DE,JP,FR,marketplace IS NULL OR marketplace IN ('US','UK','DE','JP','FR'),None))	Success	
ComplianceConstraint(Compliance(year is non-negative,COALESCE(year, 0.0) >= 0,None))	Success	

<https://aws.amazon.com/blogs/big-data/test-data-quality-at-scale-with-deequ/>

# Verify Constraints - Great Expectation



- `expect_column_values_to_not_be_null`
- `expect_column_values_to_match_regex`
- `expect_column_values_to_be_unique`
- `expect_column_values_to_match_strftime_format`
- `expect_table_row_count_to_be_between`
- `expect_column_median_to_be_between`

```
[49]: my_df.expect_column_values_to_be_in_set("PClass", ["1st", "2nd", "3rd"], mostly=.99)

[49]: {
  "success": true,
  "meta": {},
  "result": {
    "element_count": 1313,
    "missing_count": 0,
    "missing_percent": 0.0,
    "unexpected_count": 1,
    "unexpected_percent": 0.07616146230007616,
    "unexpected_percent_nonmissing": 0.07616146230007616,
    "partial_unexpected_list": [
      "*"
    ]
  },
  "exception_info": null
}
```

```
{
  "data_asset_type": "Dataset",
  "expectation_suite_name": "taxi.demo",
  "expectations": [


    ...

    {
      "expectation_type": "expect_column_values_to_not_be_null",
      "kwargs": {
        "column": "passenger_count"
      },
      "meta": {
        "BasicSuiteBuilderProfiler": {
          "confidence": "very low"
        }
      }
    },
    {
      "expectation_type": "expect_column_distinct_values_to_be_in_set",
      "kwargs": {
        "column": "passenger_count",
        "value_set": [
          1.0,
          2.0,
          3.0,
          4.0,
          5.0,
          6.0
        ]
      },
      "meta": {
        "BasicSuiteBuilderProfiler": {
          "confidence": "very low"
        }
      }
    },
    ...
  ]
}
```

**Expectations Suite**

# Verify Constraints - Great Expectation



 great\_expectations [Home](#) / [taxi.demo](#) / 20200819T024609.241003Z / 2020-08-19T02:46:09.241003+00:00 / cbb8bd044ccaa28d4db5e3d59c0be748

## Expectation Validation Result

Evaluates whether a batch of data matches expectations.

**Actions**

Validation Filter:  
[Show All](#) [Failed Only](#)  
[How to Edit This Suite](#)  
[Show Walkthrough](#)

**Table of Contents**

[Overview](#)  
[Table-Level Expectations](#)  
[passenger\\_count](#)

**Overview**  
Expectation Suite: [taxi.demo](#)  
Status: ✔ Succeeded

**Statistics**

Evaluated Expectations	6
Successful Expectations	6
Unsuccessful Expectations	0
Success Percent	100%

[Show more info...](#)

**Table-Level Expectations**

Search

Status	Expectation	Observed Value
✔	Must have between <code>8000</code> and <code>11000</code> rows.	10000
✔	Must have exactly <code>18</code> columns.	18
✔	Must have these columns in this order: <code>vendor_id</code> , <code>pickup_datetime</code> , <code>dropoff_datetime</code> , <code>passenger_count</code> , <code>trip_distance</code> , <code>rate_code_id</code> , <code>store_and_fwd_flag</code> , <code>pickup_location_id</code> , <code>dropoff_location_id</code> , <code>payment_type</code> , <code>fare_amount</code> , <code>extra</code> , <code>mta_tax</code> , <code>tip_amount</code> , <code>tolls_amount</code> , <code>improvement_surcharge</code> , <code>total_amount</code> , <code>congestion_surcharge</code>	<code>['vendor_id', 'pickup_datetime', 'dropoff_datetime', 'passenger_count', 'trip_distance', 'rate_code_id', 'store_and_fwd_flag', 'pickup_location_id', 'dropoff_location_id', 'payment_type', 'fare_amount', 'extra', 'mta_tax', 'tip_amount', 'tolls_amount', 'improvement_surcharge', 'total_amount', 'congestion_surcharge']</code>

passenger\_count



# Data Profiling

## Adult Data Set

Predict whether income exceeds \$50K/yr based on census data. Also known as "Census Income" dataset.

Data Set Characteristics:	Multivariate	Number of Instances:	48842	Area:	Social
Attribute Characteristics:	Categorical, Integer	Number of Attributes:	14	Date Donated	1996-05-01
Associated Tasks:	Classification	Missing Values?	Yes	Number of Web Hits:	1953013

### Source:

Donor:  
Ronny Kohavi and Barry Becker  
Data Mining and Visualization  
Silicon Graphics.  
e-mail: ronnyk '@' live.com for questions.

### Data Set Information:

Extraction was done by Barry Becker from the 1994 Census database. A set of reasonably clean records was extracted using the following conditions: ((AGE>16) && (AGI>100) && (AFNLWGT>1)&& (HRSWK>0))

Prediction task is to determine whether a person makes over 50K a year.



### Listing of attributes:

>50K, <=50K.

age: continuous.

workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.

fnlwgt: continuous.

education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.

education-num: continuous.

marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.

# Data Profiling



## Overview

Overview

Dataset

Variables

Warnings **6**

Reproduction

### Warnings

Dataset has 25 (0.1%) duplicate rows

Duplicates

**workclass** has 1836 (5.6%) missing values

Missing

**occupation** has 1843 (5.7%) missing values

Missing

**native-country** has 583 (1.8%) missing values

Missing

**capital-gain** has 29849 (91.7%) zeros

Zeros

**capital-loss** has 31042 (95.3%) zeros

Zeros

[https://pandas-profiling.github.io/pandas-profiling/examples/master/census/census\\_report.html](https://pandas-profiling.github.io/pandas-profiling/examples/master/census/census_report.html)

# Data Profiling



occupation  
Categorical

MISSING

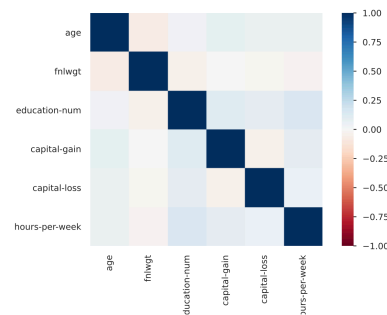
Distinct	14
Distinct (%)	< 0.1%
Missing	1843
Missing (%)	5.7%
Memory size	254.5 KiB

Prof-specialty	4140
Craft-repair	4099
Exec-managerial	4066
Adm-clerical	3770
Sales	3650
Other values (9)	10993

## Correlations

Pearson's r Spearman's  $\rho$  Kendall's  $\tau$  Phik ( $\phi_k$ ) Cramér's V ( $\phi_c$ )

Toggle correlation descriptions



[https://pandas-profiling.github.io/pandas-profiling/examples/master/census/census\\_report.html](https://pandas-profiling.github.io/pandas-profiling/examples/master/census/census_report.html)

# Data Integration from Multiple Sources

**Customer** (source 1)

<i>CID</i>	<i>Name</i>	<i>Street</i>	<i>City</i>	<i>Sex</i>
11	Kristen Smith	2 Hurley Pl	South Fork, MN 48503	0
24	Christian Smith	Hurley St 2	S Fork MN	1

**Client** (source 2)

<i>Cno</i>	<i>LastName</i>	<i>FirstName</i>	<i>Gender</i>	<i>Address</i>	<i>Phone/Fax</i>
24	Smith	Christoph	M	23 Harley St, Chicago IL, 60633-2394	333-222-6542 / 333-222-6599
493	Smith	Kris L.	F	2 Hurley Place, South Fork MN, 48503-5998	444-555-6666



OpenRefine



**Customers** (integrated target with cleaned data)

<i>No</i>	<i>LName</i>	<i>FName</i>	<i>Gender</i>	<i>Street</i>	<i>City</i>	<i>State</i>	<i>ZIP</i>	<i>Phone</i>	<i>Fax</i>	<i>CID</i>	<i>Cno</i>
1	Smith	Kristen L.	F	2 Hurley Place	South Fork	MN	48503-5998	444-555-6666		11	493
2	Smith	Christian	M	2 Hurley Place	South Fork	MN	48503-5998			24	
3	Smith	Christoph	M	23 Harley Street	Chicago	IL	60633-2394	333-222-6542	333-222-6599		24

Rahm, Erhard, and Hong Hai Do. "Data cleaning: Problems and current approaches."  
IEEE Data Eng. Bull. 23, no. 4 (2000): 3-13.

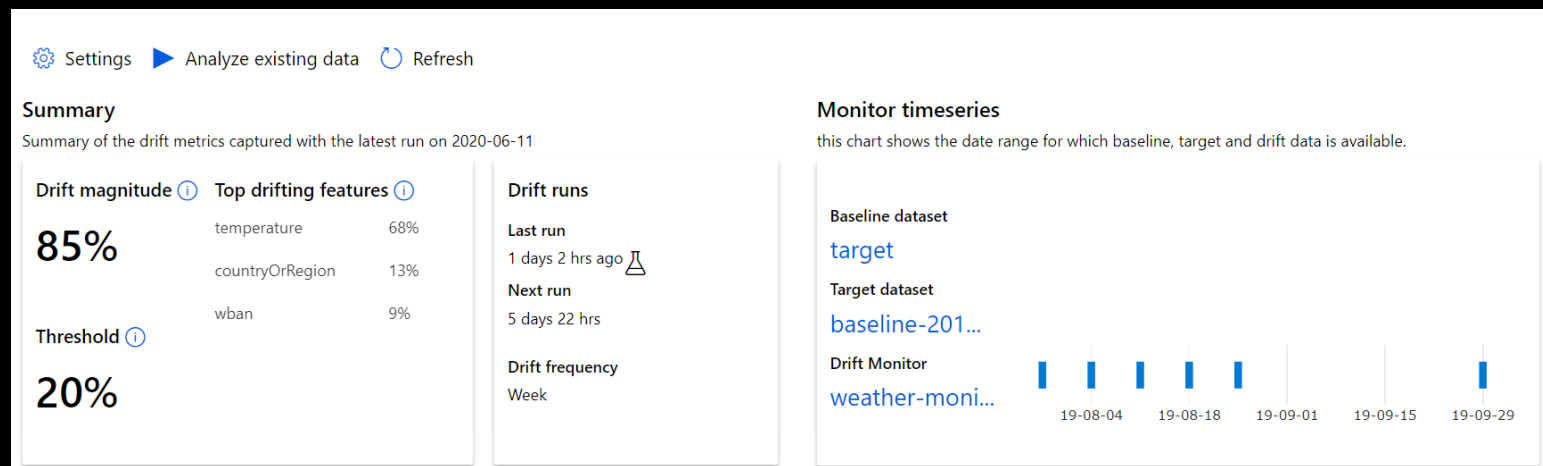
# Data Drifting

- The change in model input data that leads to model performance degradation.
- Causes
  - Upstream process changes, such as a sensor being replaced that changes the units of measurement from inches to centimeters.
  - Data quality issues, such as a broken sensor always reading 0.
  - Natural drift in the data, such as mean temperature changing with the seasons.
  - Change in relation between features, or covariate shift.

# Data Drifting

- Detect and alert to data drift on new data in a dataset.
- Analyze historical data for drift.
- Profile new data over time.
- Monitor model performance

# Data Drifting



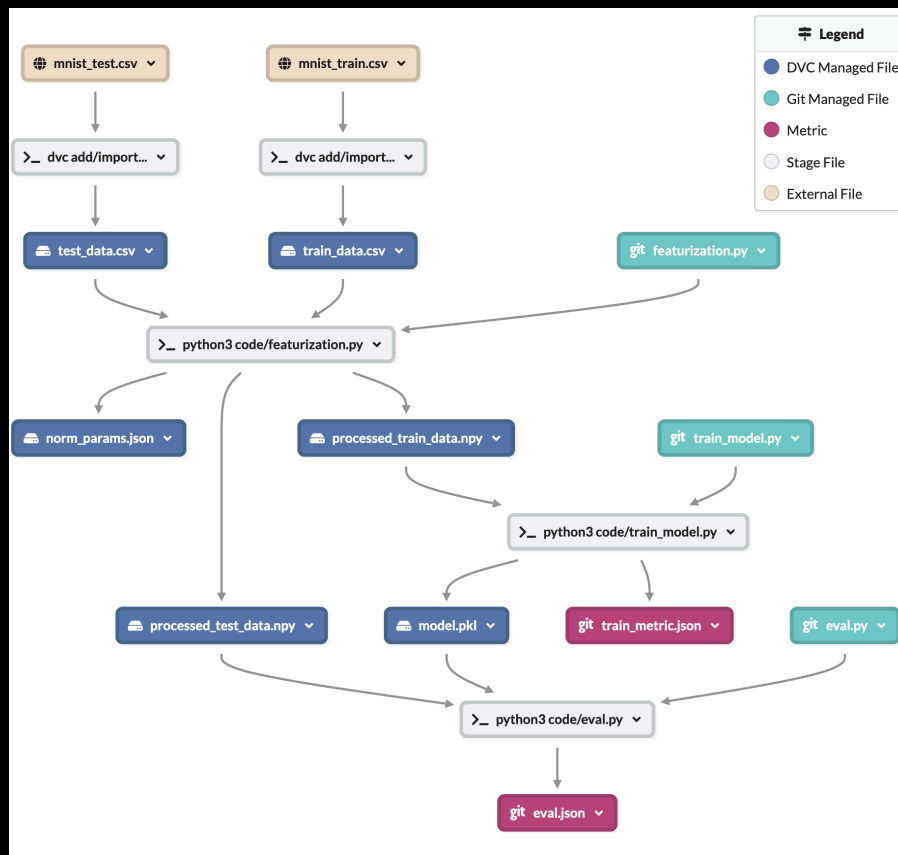
*A percentage of drift between the baseline and target dataset over time. Ranging from 0 to 100, 0 indicates identical datasets and 100 indicates the Azure Machine Learning data drift model can completely tell the two datasets apart. Noise in the precise percentage measured is expected due to machine learning techniques being used to generate this magnitude.*

# Data Drifting

- **Population Stability Index**, popular in financial services, is a measure of population stability between two population samples.
- **Kullback–Leibler (or KL) divergence** is a measure of how one probability distribution is different from a second, reference probability distribution.
- **Jensen-Shannon (or JS) divergence** is a method of measuring the similarity between two probability distributions. It is based on the KL divergence, with some notable differences, including that it is symmetric and it always has a finite value.
- **Kolmogorov-Smirnov test (or KS test)** is a nonparametric test of the equality of continuous (or discontinuous), one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample K–S test), or to compare two samples (two-sample K–S test).



# Data/Experiment Version Control



<https://dagshub.com/leo/DAGsHub-Tutorial-MNIST>