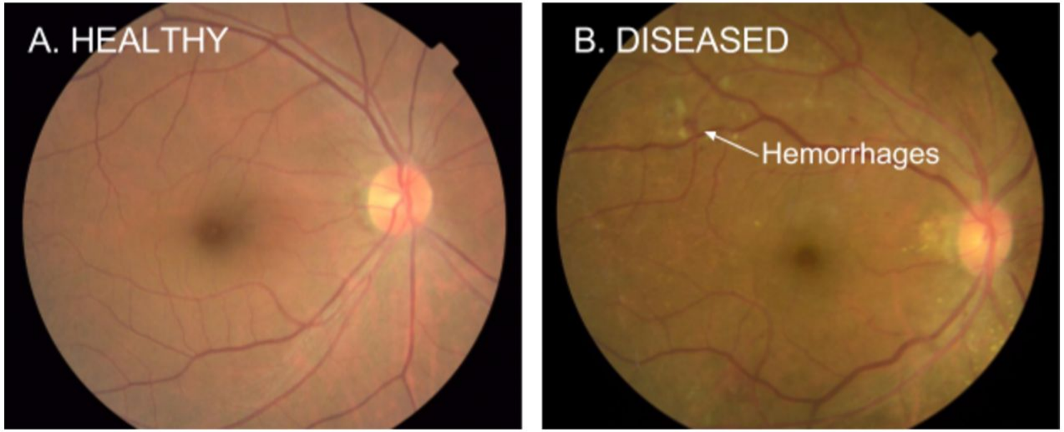
Data Management Challenges in Production Machine Learning

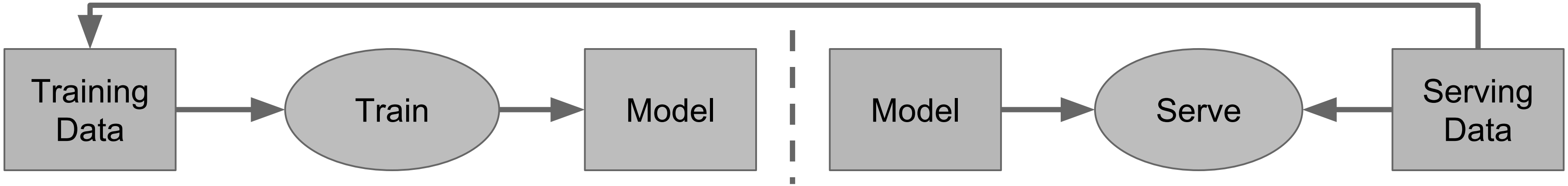
Neoklis Polyzotis, Sudip Roy, Steven Whang, Martin Zinkevich

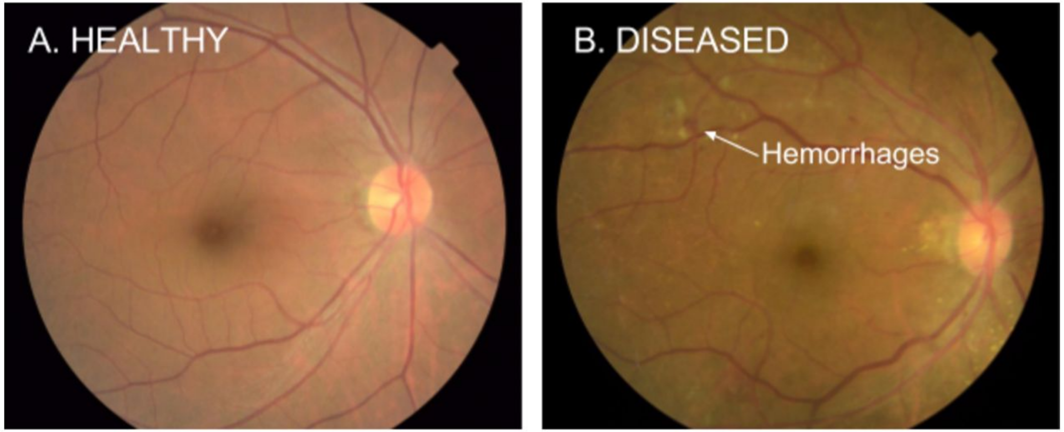
ML in front of consumers



Source: [Deep Learning for Detection of Diabetic Eye Disease, Google Research Blog]

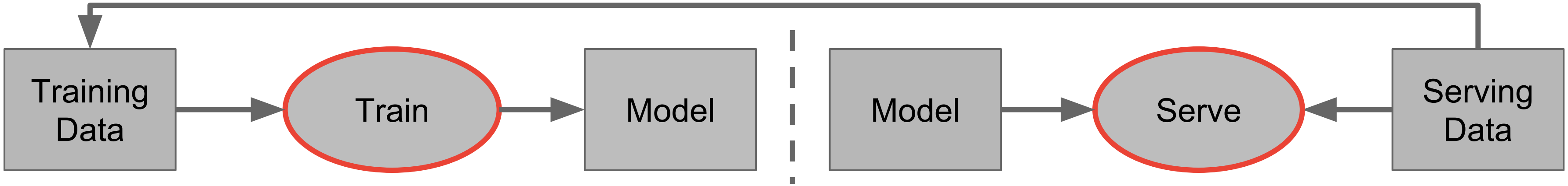
ML behind the scenes





Source: [Deep Learning for Detection of Diabetic Eye Disease, Google Research Blog]

The data flow point-of-view



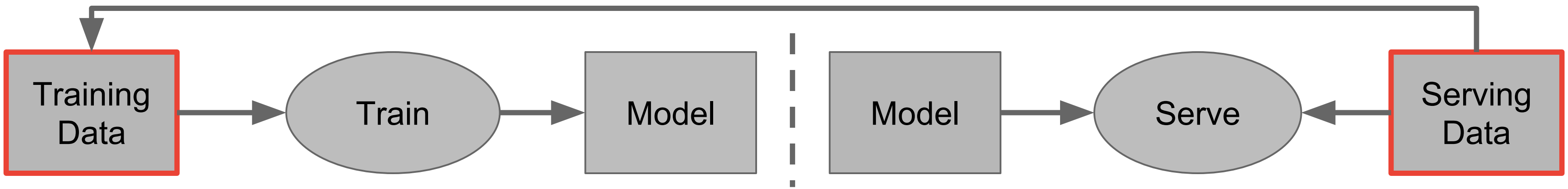
“Train” and “Serve” are data flows.

Optimizing these data flows is an interesting research problem.

- DB technology and principles are relevant in this new context.
- Velox [CBG+ CIDR15], Weld [PTS+ CIDR17], SystemML [BDE+ VLDB16]

This is NOT what this tutorial is about.

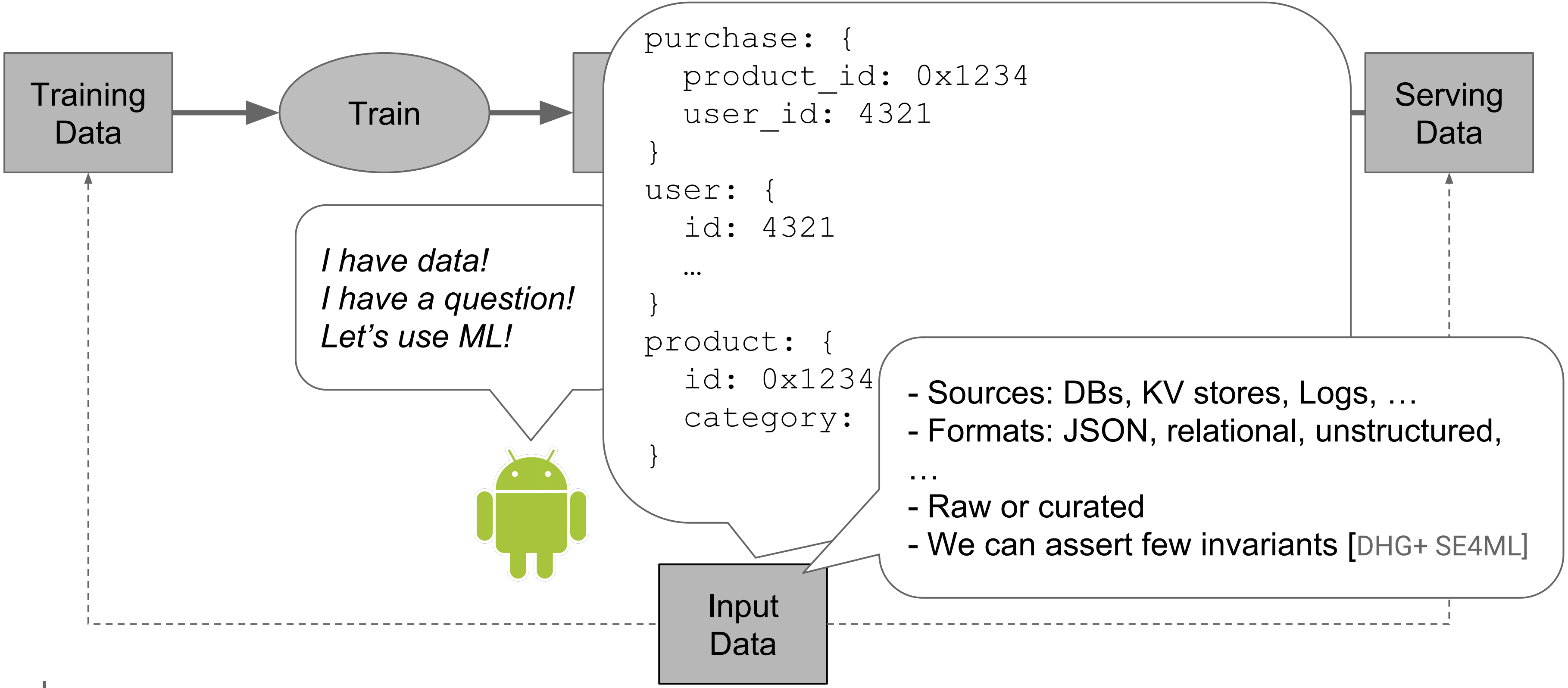
This tutorial: The data ~~flow~~ point-of-view



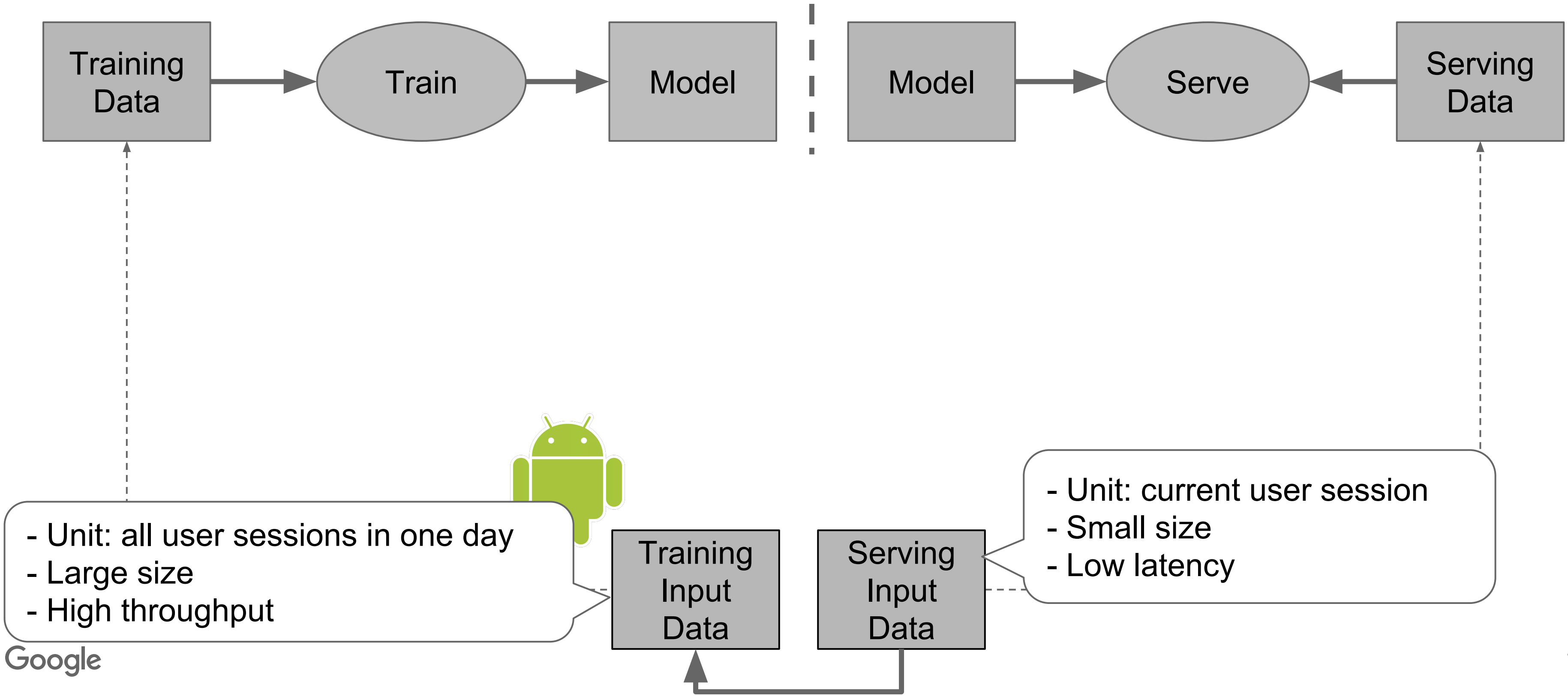
What data-management issues arise when deploying ML in production?

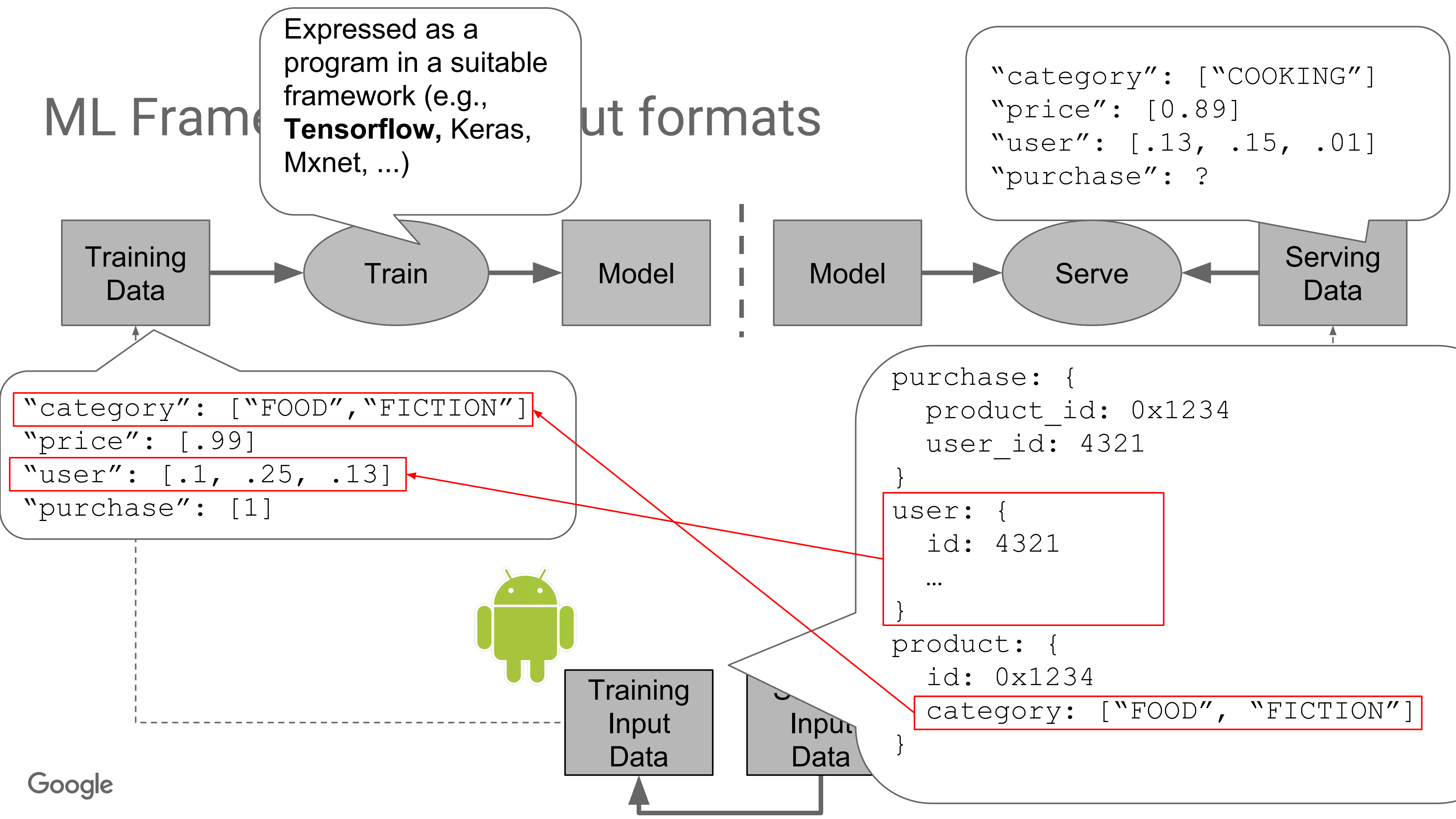
- Having the right data is crucial for model quality.
- Preparing data for an ML pipeline requires effort and care.
- Invalid data can cause outages in production ⇒ data monitoring, validation, and fixing are essential.

Starting point: Data and a question

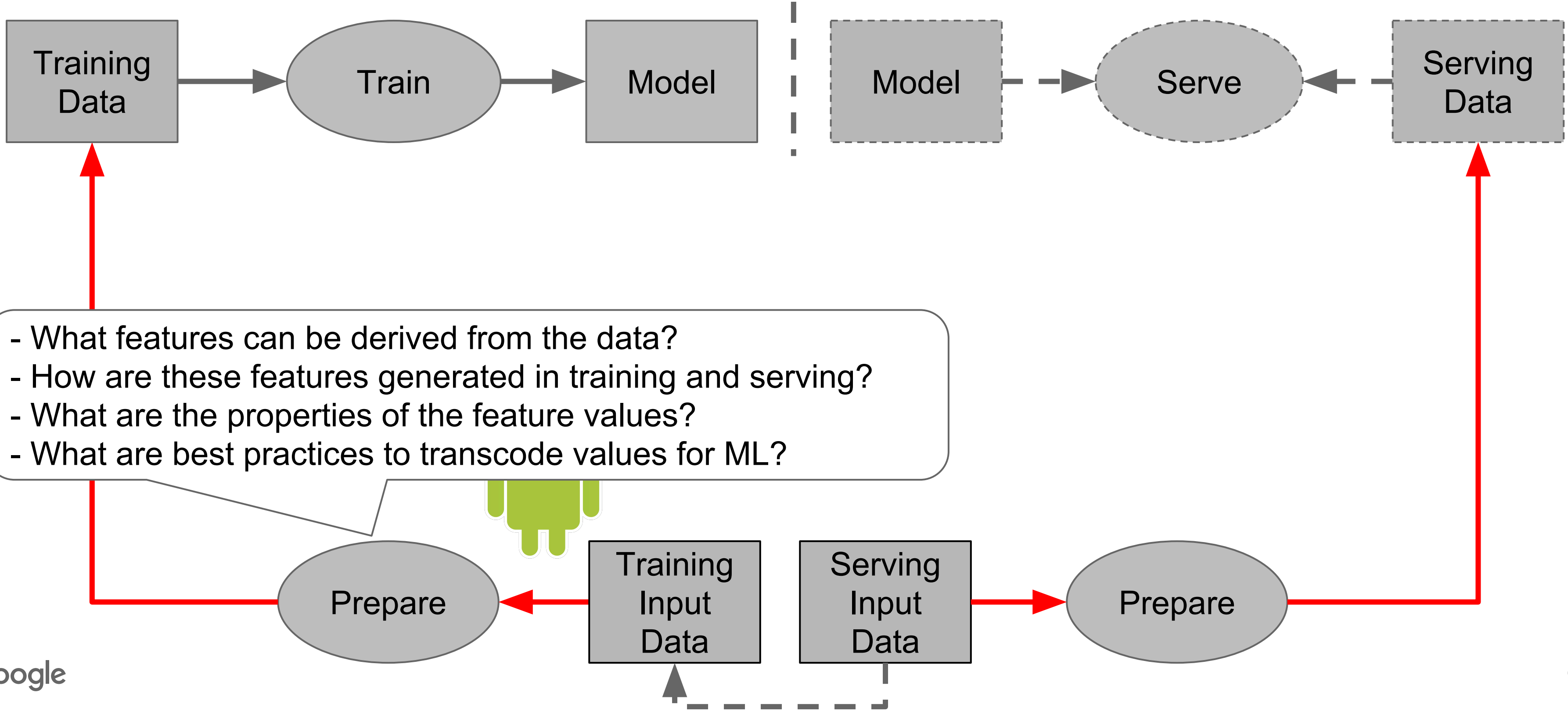


Data-access paths in training/serving

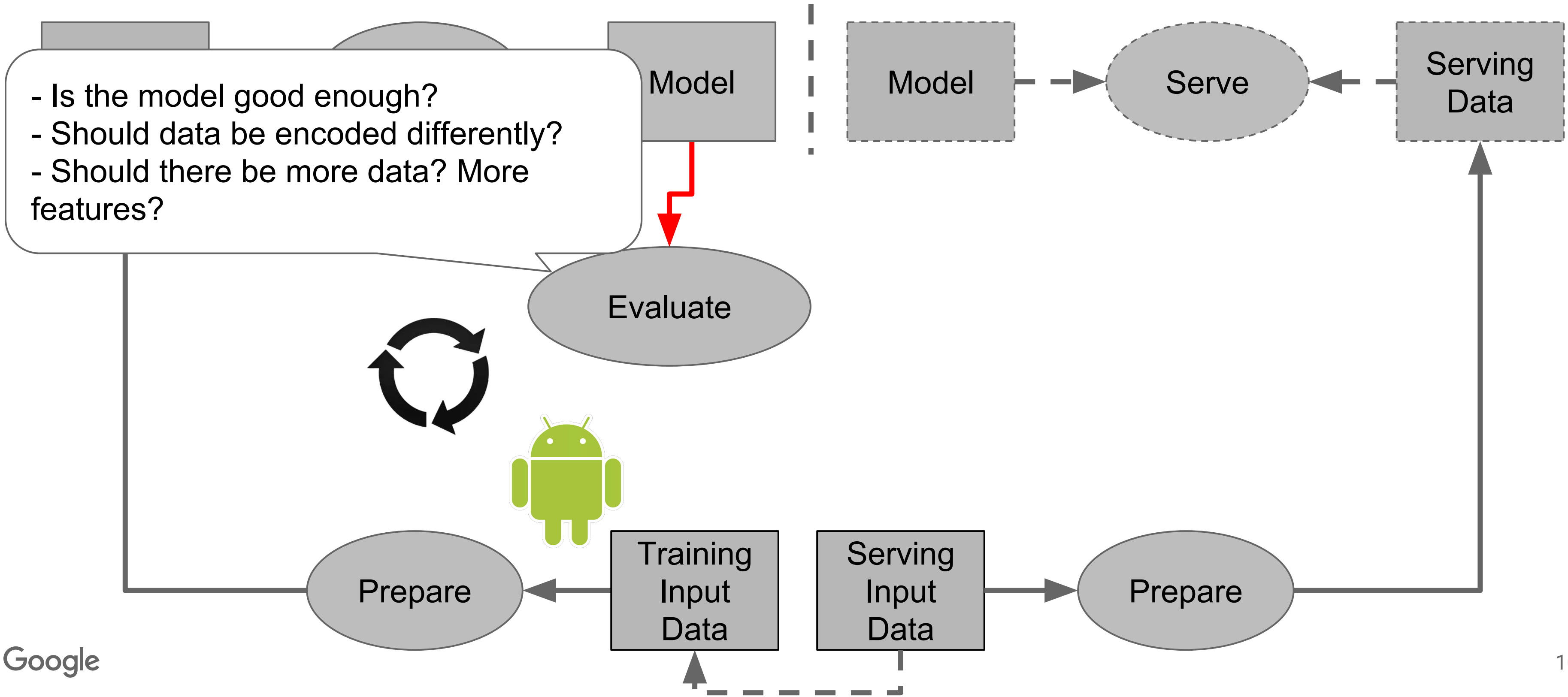



ML Frame
ut formats
Expressed as a program in a suitable framework (e.g., **Tensorflow**, Keras, Mxnet, ...)
Training Data
Train
Model
Model
Serve
Serving Data
"category": ["COOKING"]
"price": [0.89]
"user": [.13, .15, .01]
"purchase": ?
"category": ["FOOD","FICTION"]
"price": [.99]
"user": [.1, .25, .13]
"purchase": [1]
purchase: {
product_id: 0x1234
user_id: 4321
}
user: {
id: 4321
…
}
product: {
id: 0x1234
category: ["FOOD", "FICTION"]
}
Training Input Data
Input Data
Google

Preparing the data

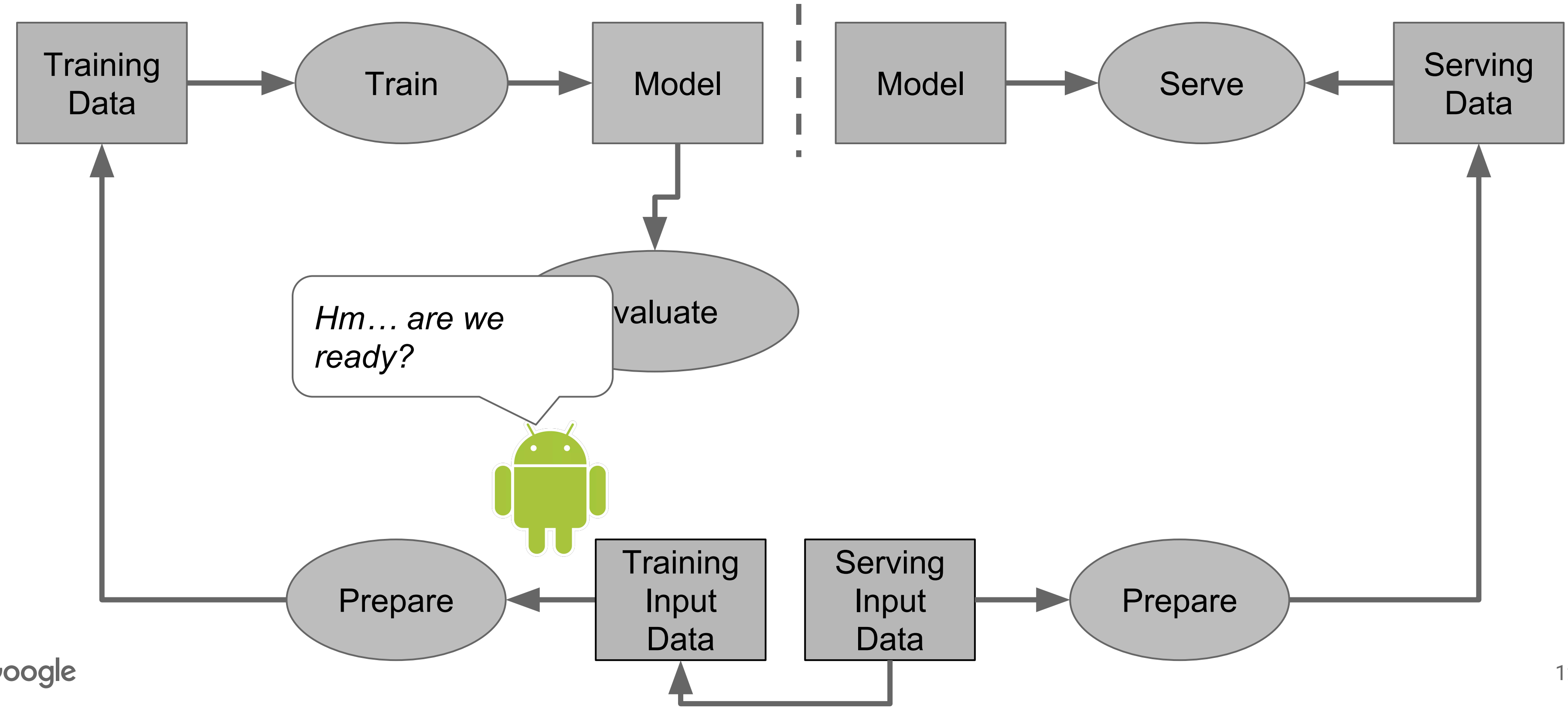


Getting to a good model



Several experiments later...

Ready to launch!



An example of data failure

Refactor backend that generates a feature

- No new features or data, same training and serving logic

An example of data failure

Refactor backend that generates a feature → Prod rollout

- No new features or data, same training and serving logic

An example of data failure

Refactor backend that generates a feature → Prod rollout → Incompatible binaries result in errors ⇒ feature = -1

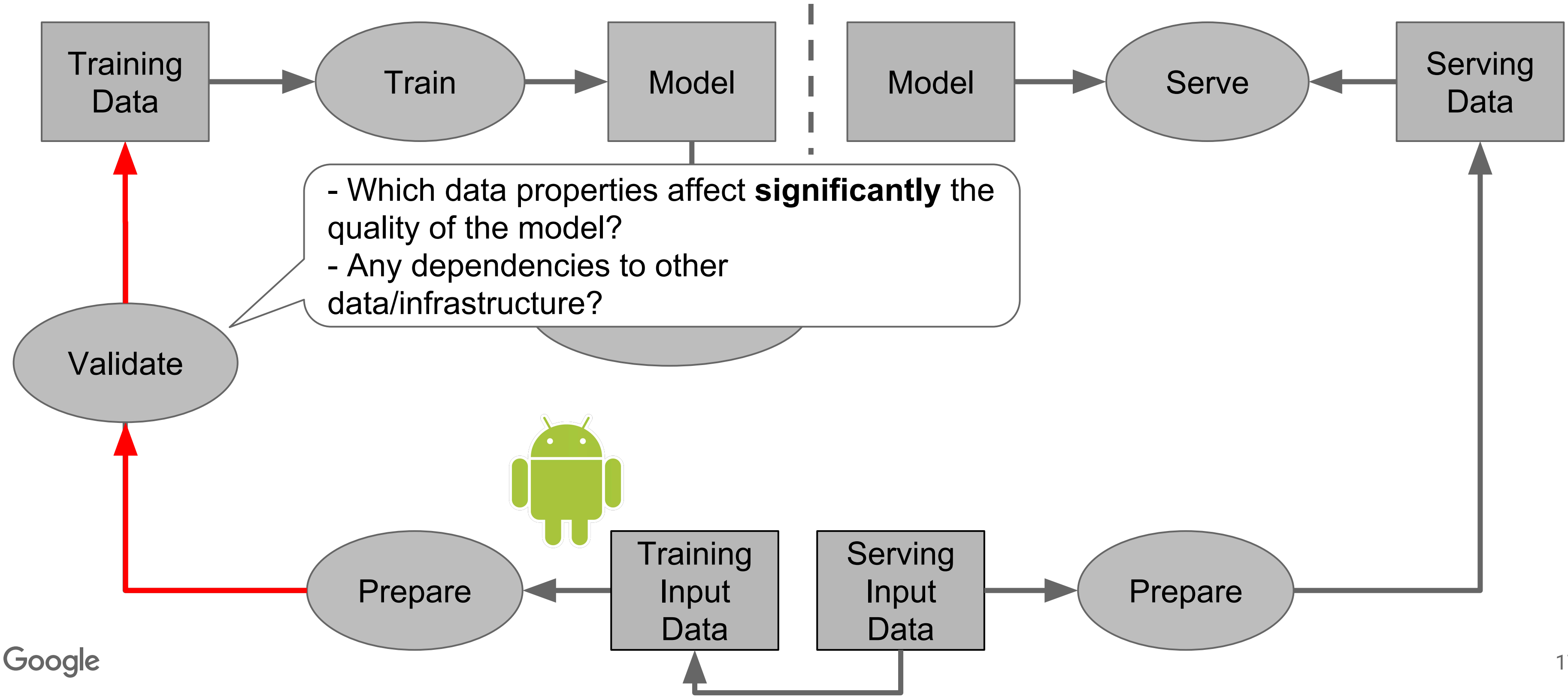
- No new features or data, same training and serving logic

An example of data failure

Refactor backend that generates a feature → Prod rollout → Incompatible binaries result in errors ⇒ feature = -1

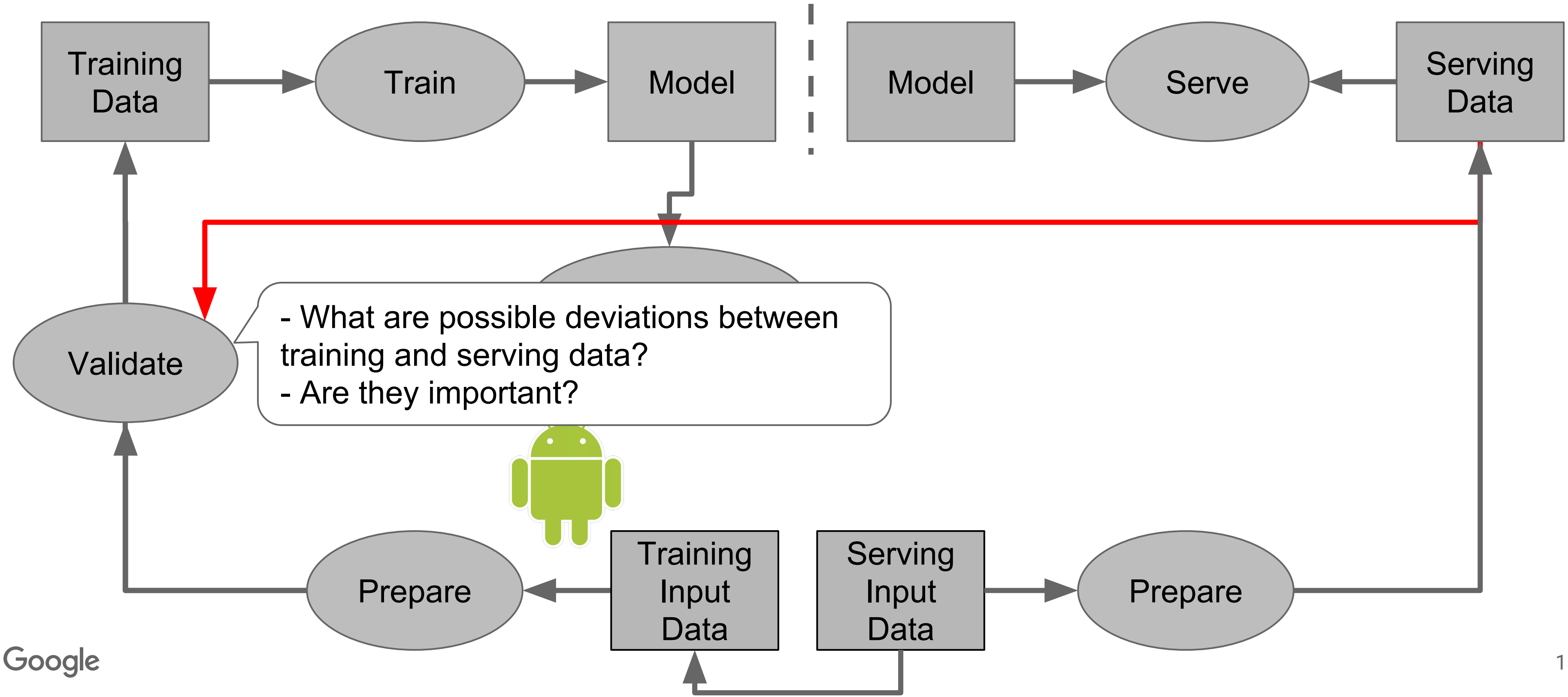
- No new features or data, same training and serving logic
- Model performance goes south
- Issues propagate through the system (bad serving data ⇒ bad training data ⇒ bad models)
- Re-training can be expensive ⇒ Catching errors early is important

Life of an ML pipeline: Validating data

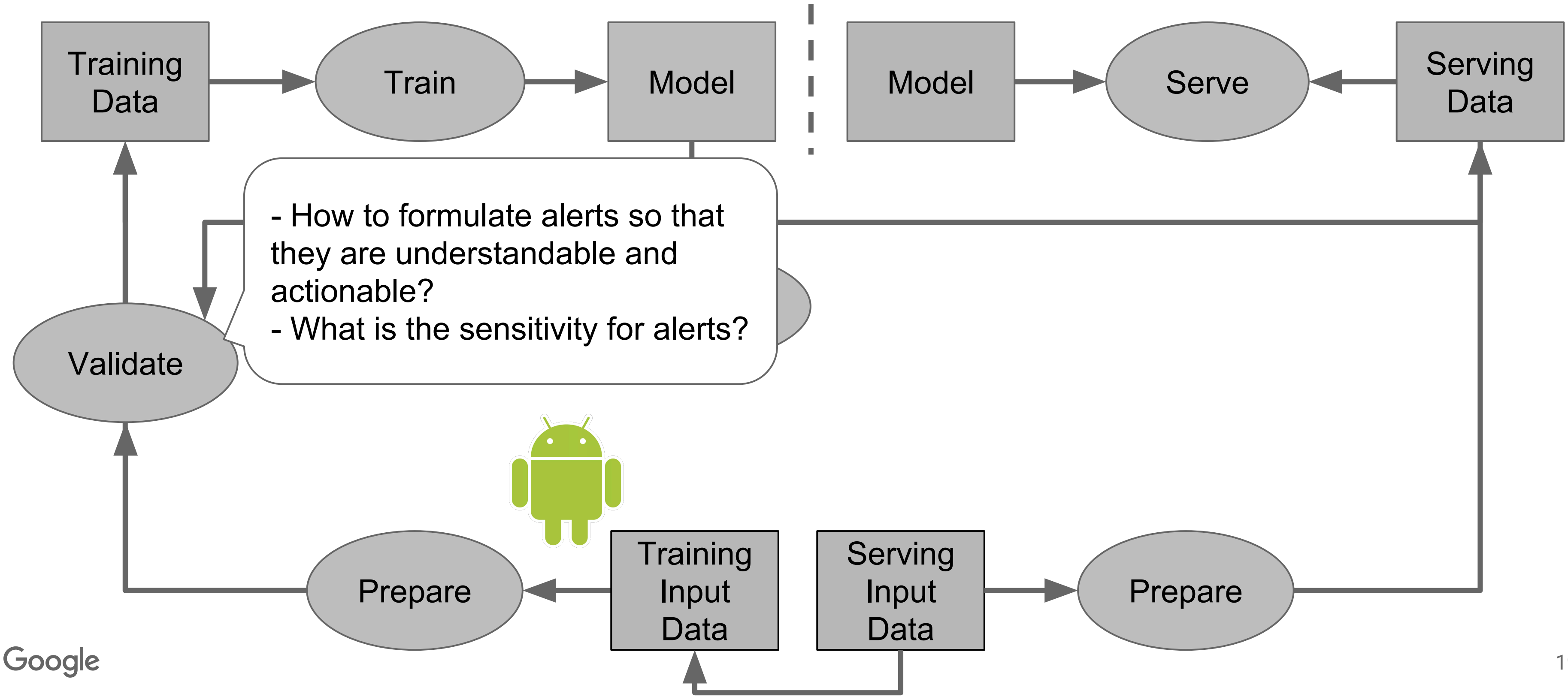


Google

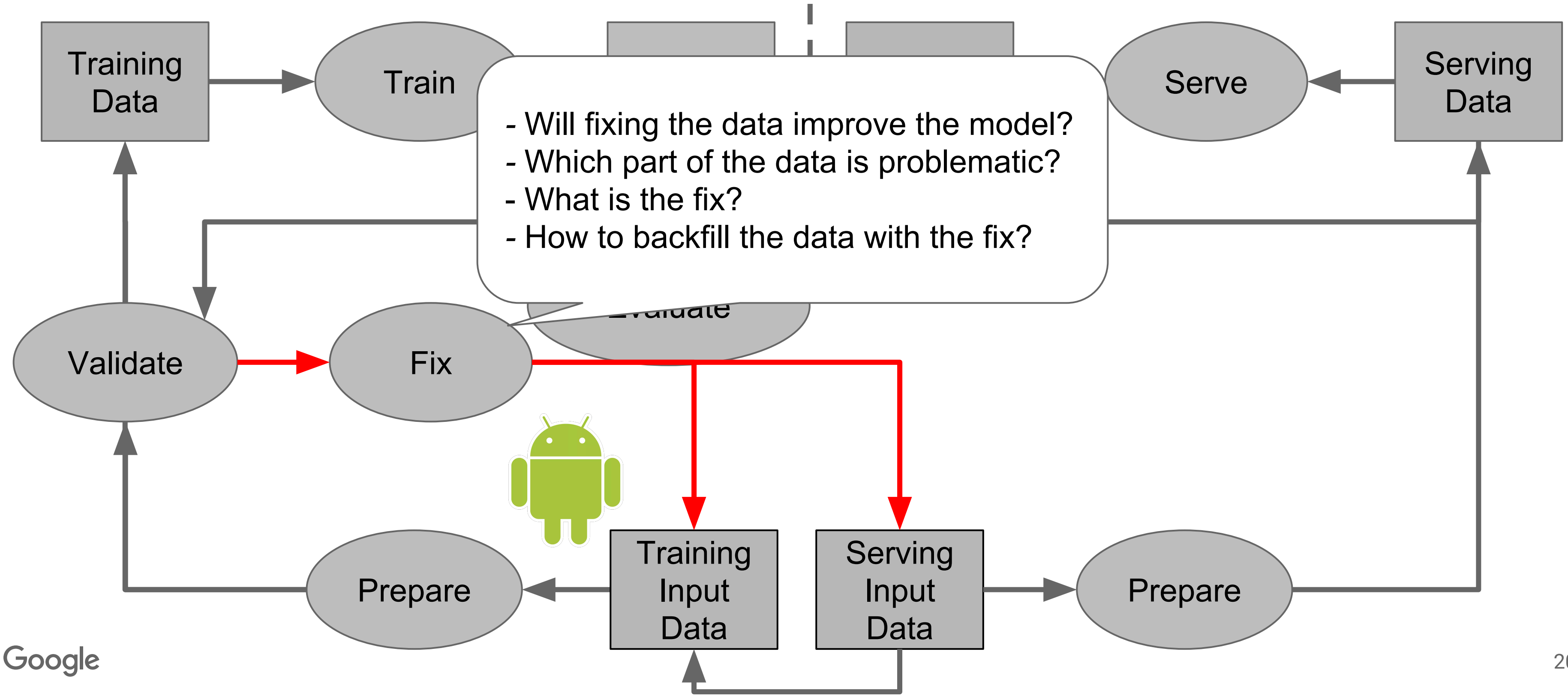
Tracking training/serving skew



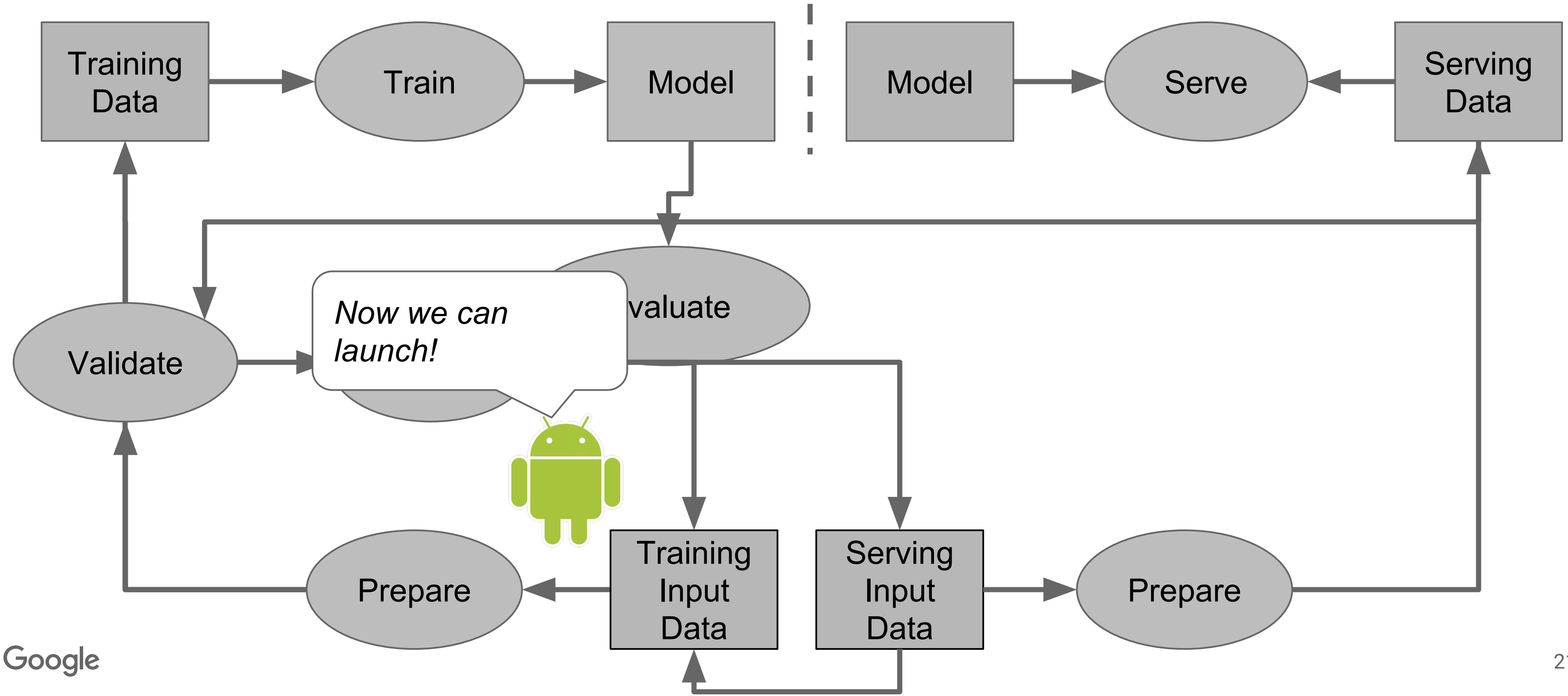
Alerting on data errors



Fixing data

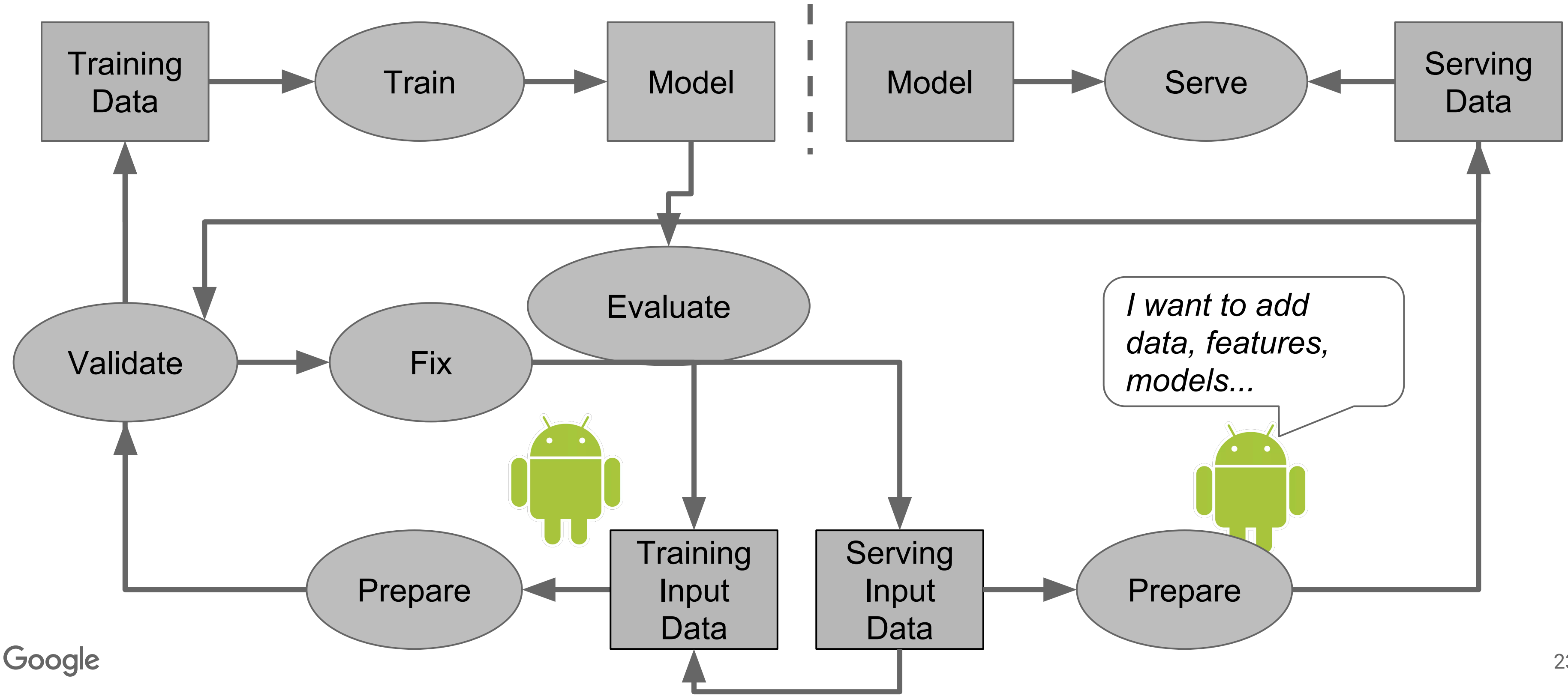

Training Data
Train
Serve
Serving Data
- Will fixing the data improve the model?
- Which part of the data is problematic?
- What is the fix?
- How to backfill the data with the fix?
Validate
Fix
Prepare
Training Input Data
Serving Input Data
Prepare

Everything in place

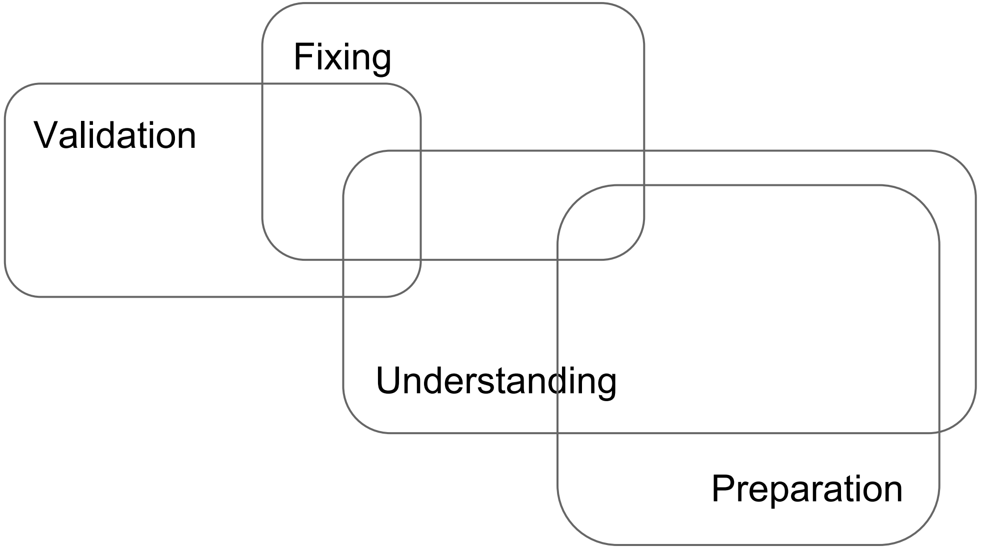


Several weeks (and production fires) later...

Life of an ML pipeline: The cycle starts over



1st dimension: High-level data activities



2nd dimension: Users

ML Expert

Broad knowledge of ML. Knows how to create models and how to use statistics. Advises on dozens of pipelines.

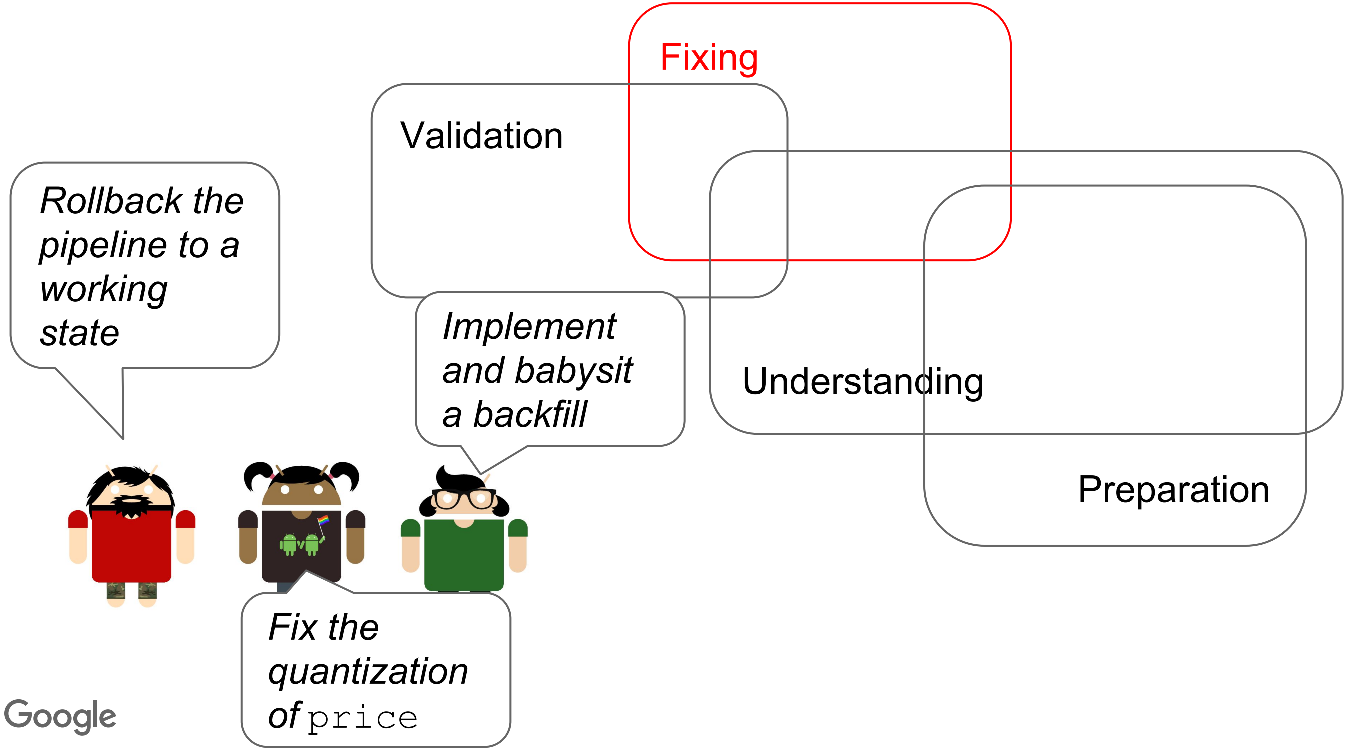
SWE

Understands the problem domain. Most ML experience is with this product.
Coding is world class.

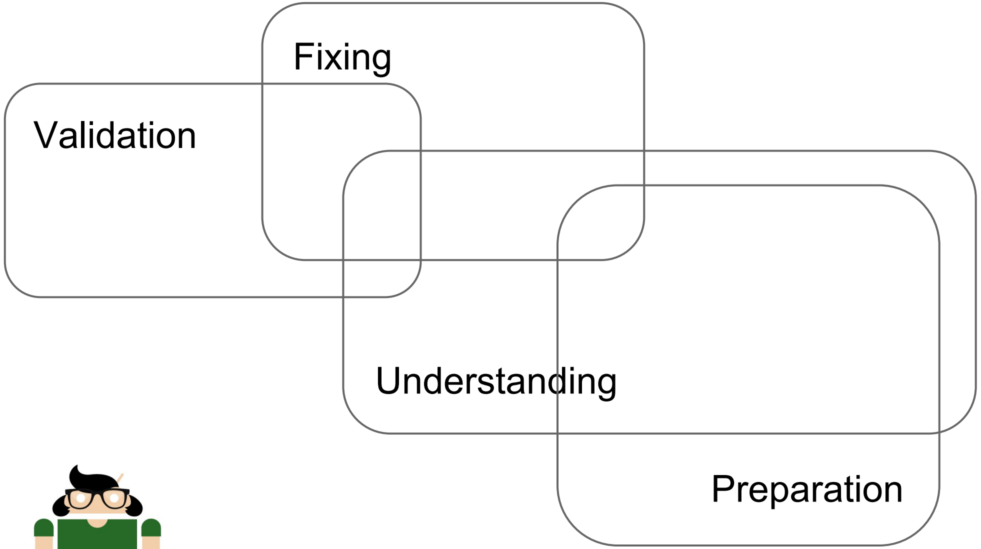
SRE

Problem fixer. On-call for possibly hundreds of pipelines. Can't afford to know the details. Dealing with many issues simultaneously.

2nd dimension: Users



3rd dimension: Time in the pipeline’s lifecycle



Experiment

Launch

Refinement

Maintenance

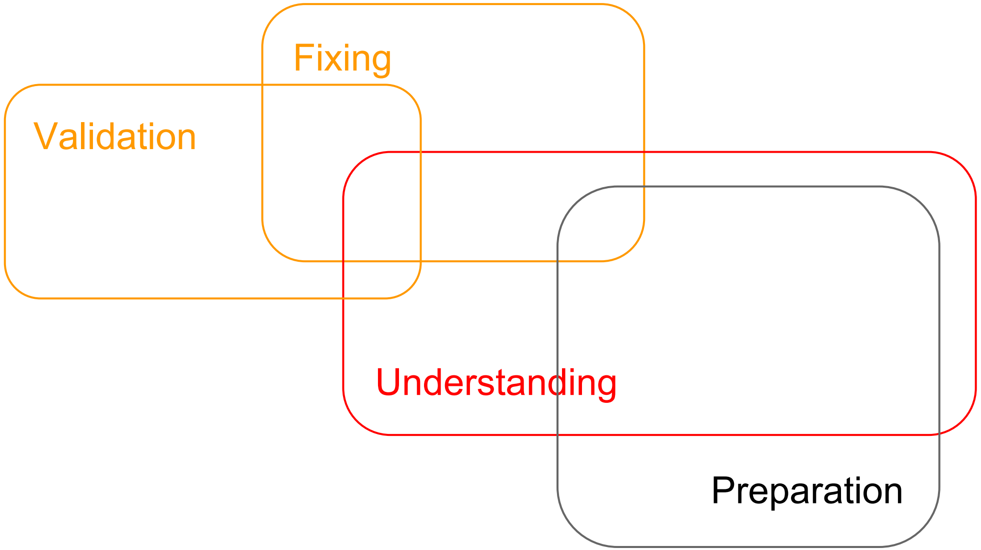
...

Organization of the tutorial

Part 1: Understanding

Part 2: Validation + Fixing

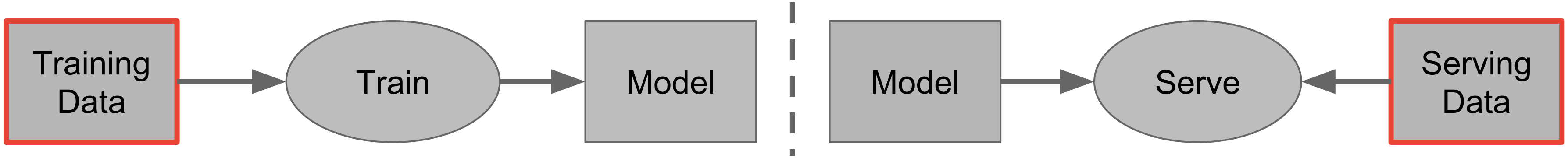
Part 3: Preparation



Driving questions:

- What previous work is relevant?
- What is lacking in terms of ML?
- What are interesting research directions?

Backstory of this tutorial



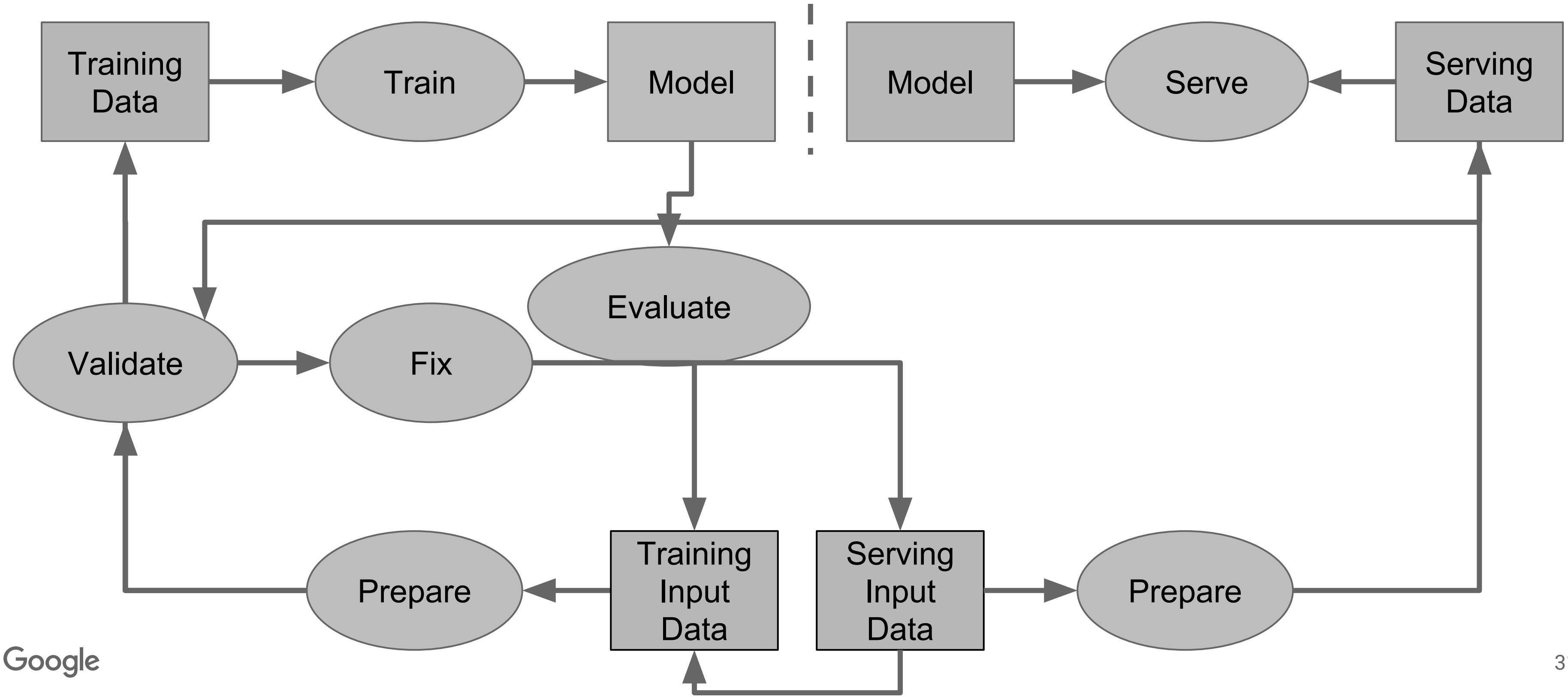
- Influenced by our experience with infra for ML pipelines in production.

"The Anatomy of a Production-Scale Continuously-Training Machine Learning Platform", to appear in KDD'17

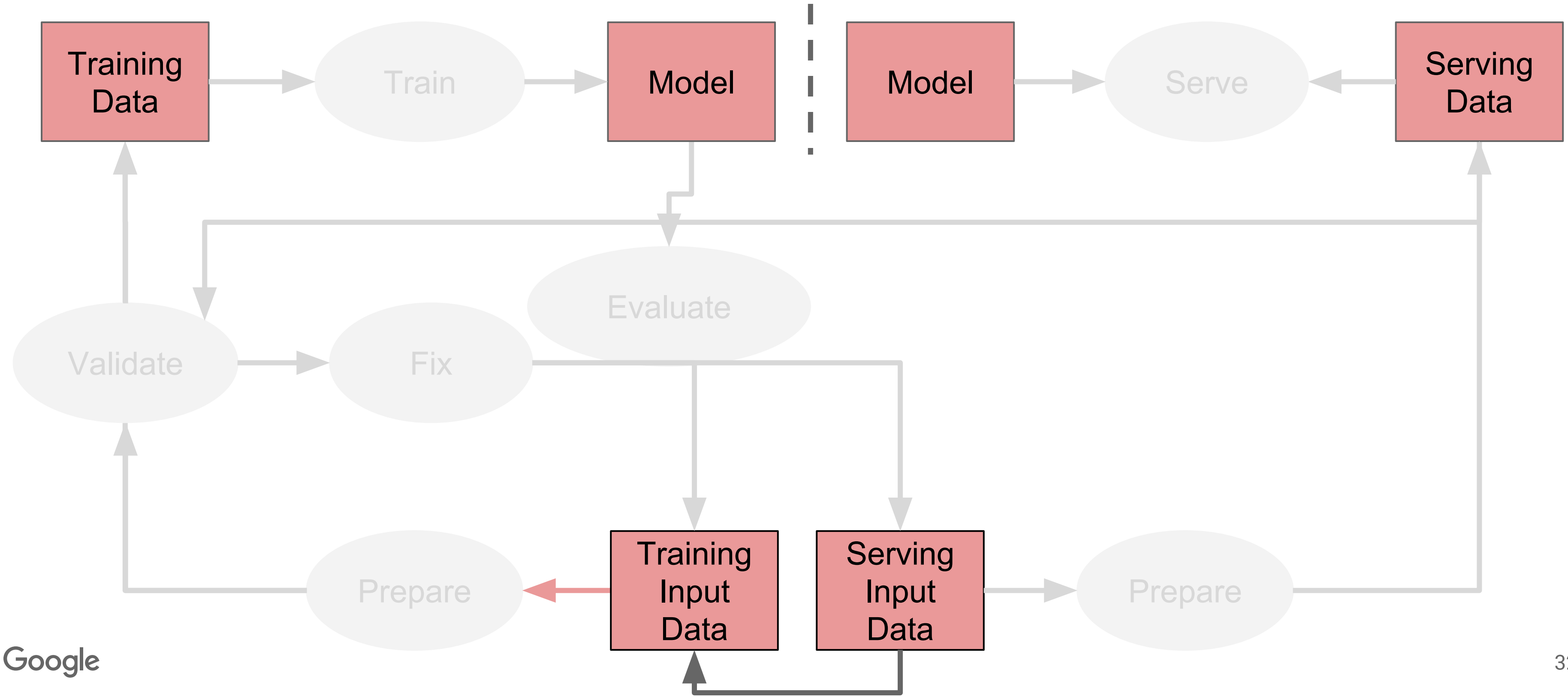
- Presenters: three DB researchers and one ML researcher.
- DB folks have the technical background to deal with data problems but ML folks will provide important context, and vice versa.

Data Understanding

Data understanding in ML pipeline

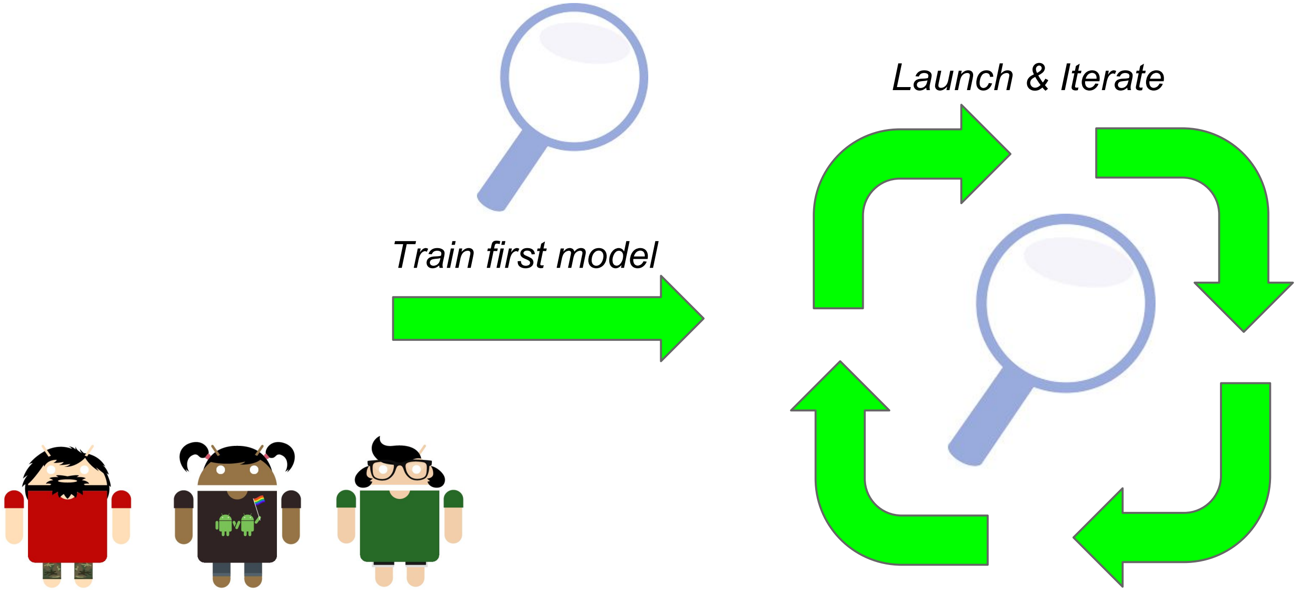


Data understanding in ML pipeline



Data understanding in ML pipeline

- Sanity checks before training the first model
- Other analyses during launch and iterate cycle



Sanity checks on **expected** shape before training first model

- Check a feature's min, max, and most common value
  - Ex: Latitude values must be within the range [-90, 90] or [-π/2, π/2]
- The histograms of continuous or categorical values are as expected
  - Ex: There are similar numbers of positive and negative labels
- Whether a feature is present in enough examples
  - Ex: Country code must be in at least 70% of the examples
- Whether a feature has the right number of values (i.e., cardinality)
  - Ex: There cannot be more than one age of a person

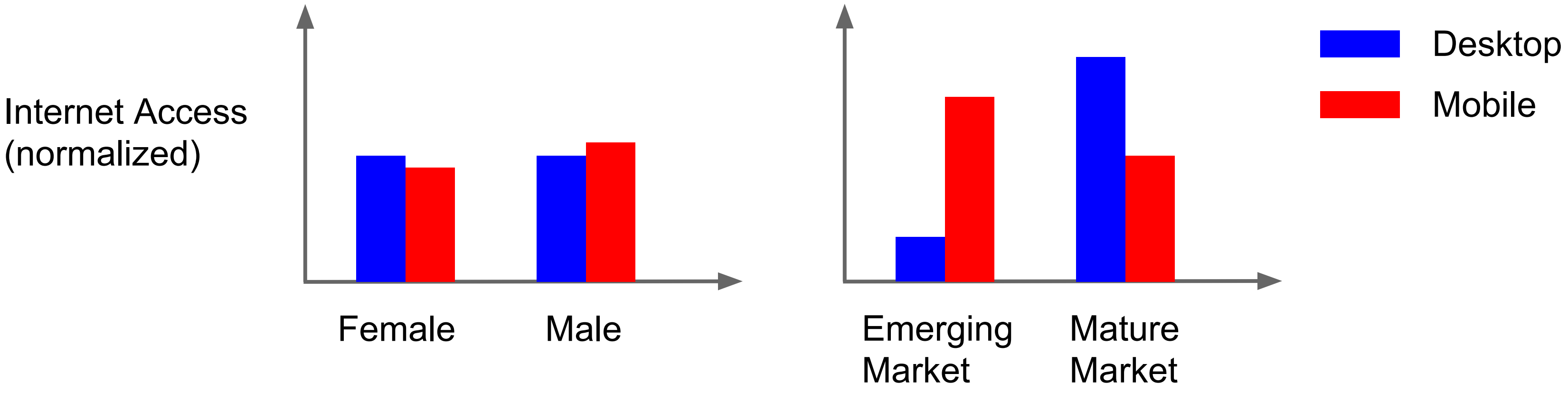
How do we know what to expect of the data?

- If we know exactly what we need, then just use SQL for checks
- However, features may not have clear ownership, which makes it hard to keep track of what to expect
- Visualization tools can help us understand of data shape by discovering surprising properties of data (and thus develop better sanity checks)
    - Visualization recommendations
        - SeeDB [VRS+ VLDB15]
        - ZenVisage [SKL+ VLDB16]
    - False discovery control with multi-hypothesis testing
        - QUDE [BSK+ CIDR17, ZSZ+ SIGMOD17]

SeeDB: Data-driven visualization

[VRM+ PVLDB15]

- Recommends "interesting" visualizations using a deviation-based metric
  - Provides insights to users on what to expect of the training data and subsequent ones
  - Zenvisage: Follow-up work on interactive visual analytics using ZQL [SKL+ PVLDB 16]
- Research question: what is the confidence of these visualizations?

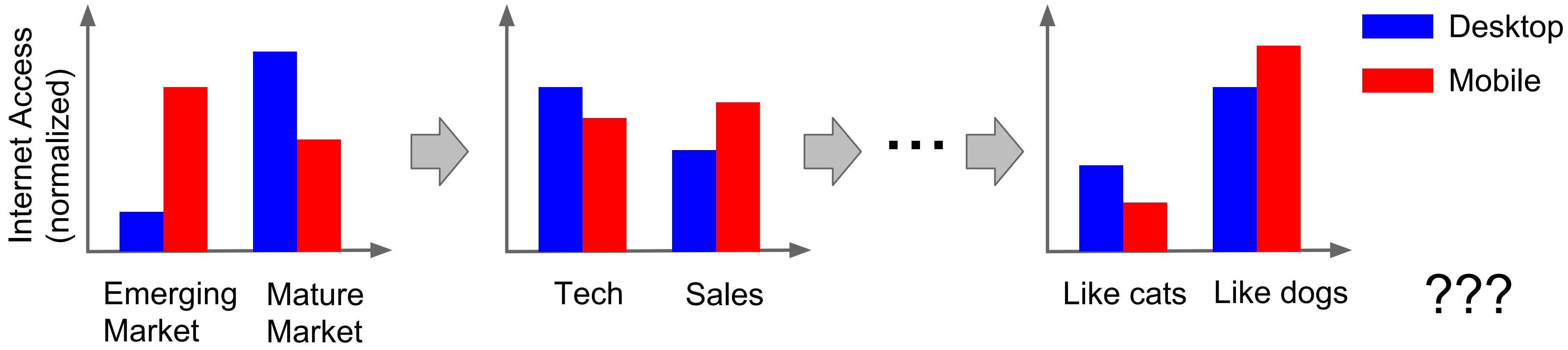


<Uninteresting Visualization> *<Interesting Visualization>*

QUDE: Controlling false discoveries

[BSK+ CIDR17, ZSZ+ SIGMOD17]

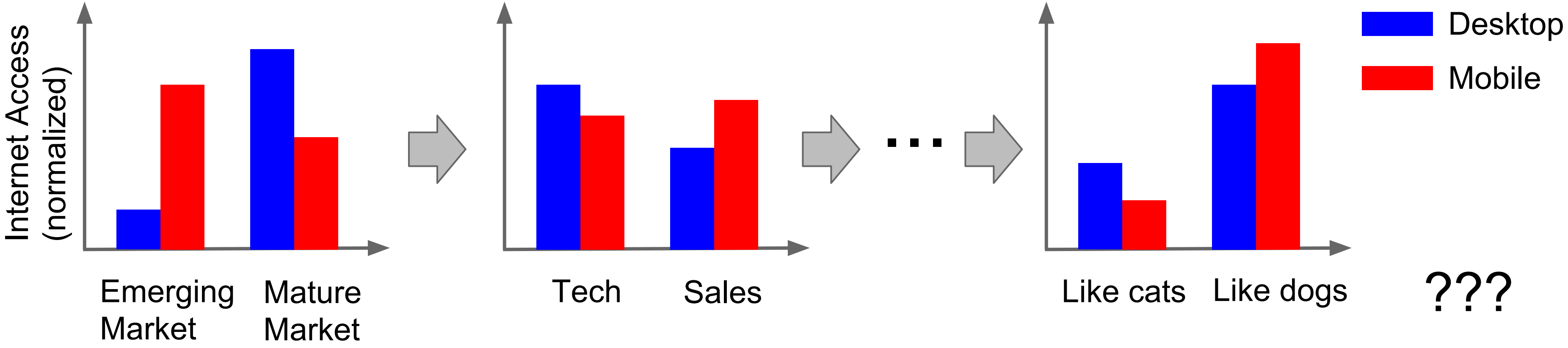
- Provides automatic control of false discoveries (multiple hypothesis testing error) for visual, interactive data exploration
  - Traditional methods for controlling FWER (Bonferroni correction) or FDR (Benjamini-Hochberg procedure) assume “static” hypotheses and do not work for interactive data exploration
  - Proposes α-investing with control mFDR



QUDE: Controlling false discoveries

[BSK+ CIDR17, ZSZ+ SIGMOD17]

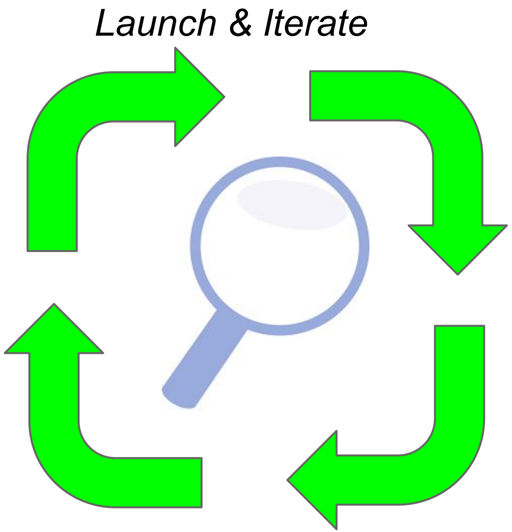
- Provides automatic control of false discoveries (multiple hypothesis testing error) for visual, interactive data exploration
  - Traditional methods for controlling FWER (Bonferroni correction) or FDR (Benjamini-Hochberg procedure) assume “static” hypotheses and do not work for interactive data exploration
  - Proposes α-investing with control mFDR



<Significant> *<Not significant>* ***<Sorry, out of α-budget!>***

Data understanding during launch and iterate

- Feature-based analysis
- Data lifecycle analysis
- Open questions



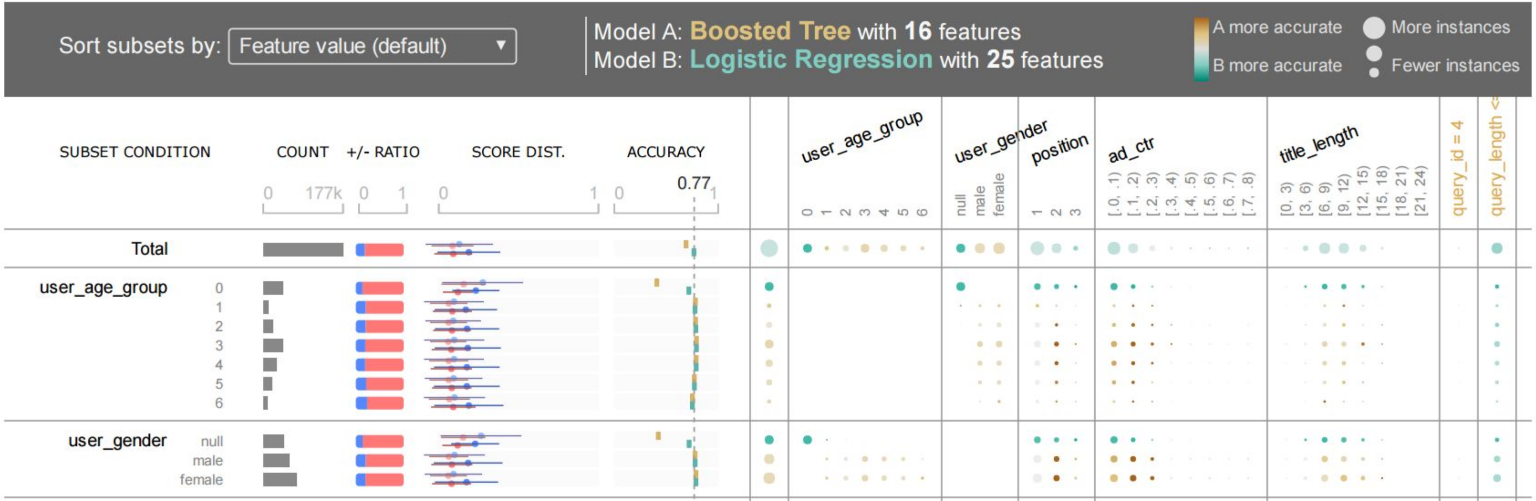
Feature-based analysis

- Types of ML analyses
  - Given a model, identify training data slices (based on features) that lead to high/low model quality
    - E.g., App recommendation model performs poorly for people in CJK countries
  - Given serving logs, detect any training-serving skew on certain slices
    - E.g., The gender ratio between the training data and serving logs is significantly different for people in the age range [20, 40].
- Data cube analysis is effective for analyzing “slices” of data, which are defined with features or feature crosses
  - MLCube [KFC HILDA16]
  - Intelligent roll-up [SS VLDB01]
  - Smart drill-down [JGP ICDE16]

Visual exploration of ML results using data cube analysis

[KFC HILDA16]

- Enables users to define slices using feature conditions and computes aggregate statistics and evaluation metrics over the slices
  - Helps understand and debug a single model or compare two models
- Research question: how to automatically prioritize user attention and identify what are the "important slices"?

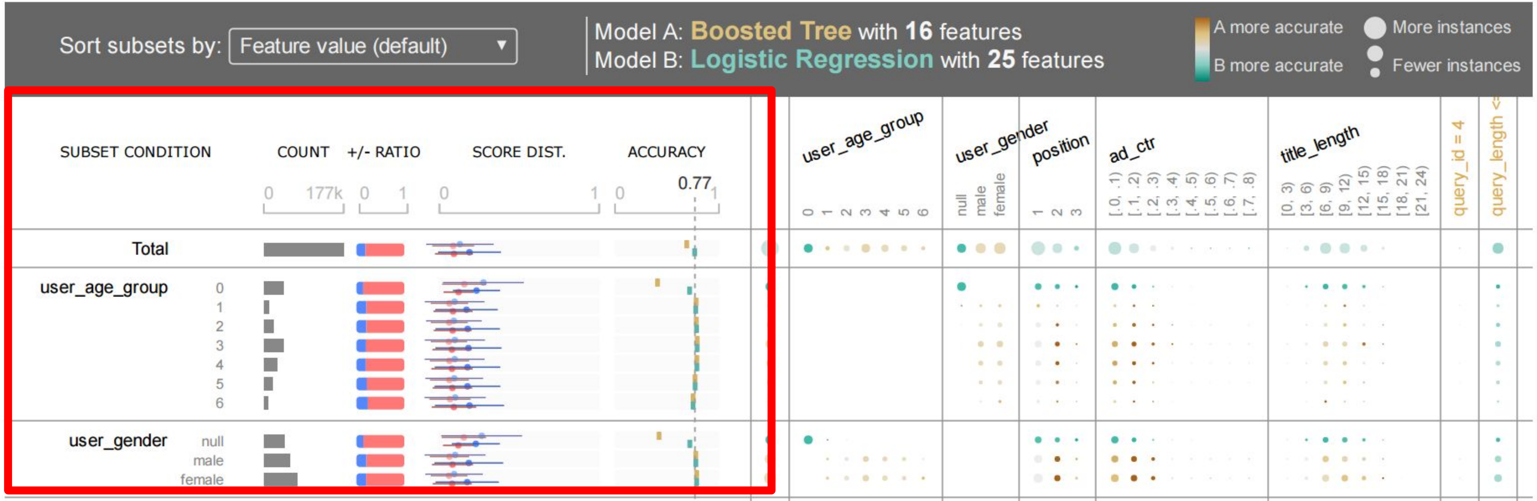


Visual exploration of ML results using data cube analysis

[KFC HILDA16]

- Enables users to define slices using feature conditions and computes aggregate statistics and evaluation metrics over the slices
  - Helps understand and debug a single model or compare two models
- Research question: how to automatically prioritize user attention and identify what are the “important slices”?

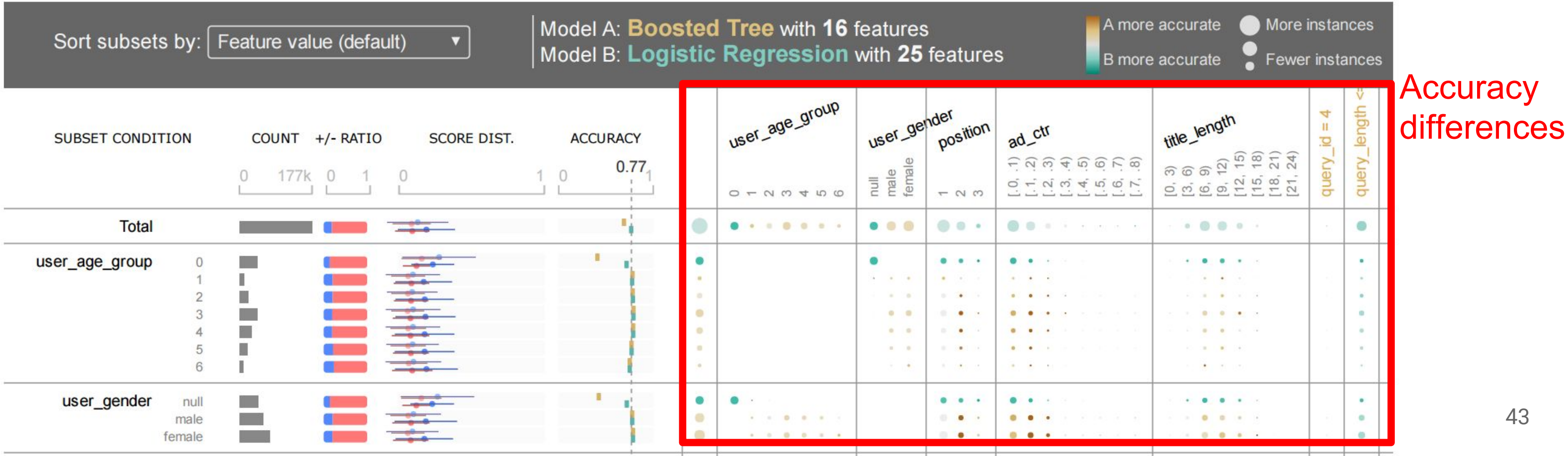
Summary stats



Visual exploration of ML results using data cube analysis

[KFC HILDA16]

- Enables users to define slices using feature conditions and computes aggregate statistics and evaluation metrics over the slices
  - Helps understand and debug a single model or compare two models
- Research question: how to automatically prioritize user attention and identify what are the "important slices"?



Intelligent rollups in multidimensional OLAP data

[SS VLDB01]

- Automatically generalizes from a specific problem case in detailed data and return the broadest context in which the problem occurs
  - Can be used to find problematic slices in training data that positively/negatively affect model metric (e.g., loss, AUC, calibration)
  - More recent work, but using drill downs [JGP ICDE16]
- Research question: training data is mostly flat and noisy with no hierarchy, so we cannot always rely on clean hierarchies

Location	Gender	Age	Nationality
Chicago	Female	[30, 40]	Greek

Month	Jan	Feb	Mar	Apr
Loss	0.11	0.09	0.1	0.5

Intelligent rollups in multidimensional OLAP data

[SS VLDB01]

- Automatically generalizes from a specific problem case in detailed data and return the broadest context in which the problem occurs
    - Can be used to find problematic slices in training data that positively/negatively affect model metric (e.g., loss, AUC, calibration)
    - More recent work, but using drill downs [JGP ICDE16]
- Research question: training data is mostly flat and noisy with no hierarchy, so we cannot always rely on clean hierarchies

Generalizations

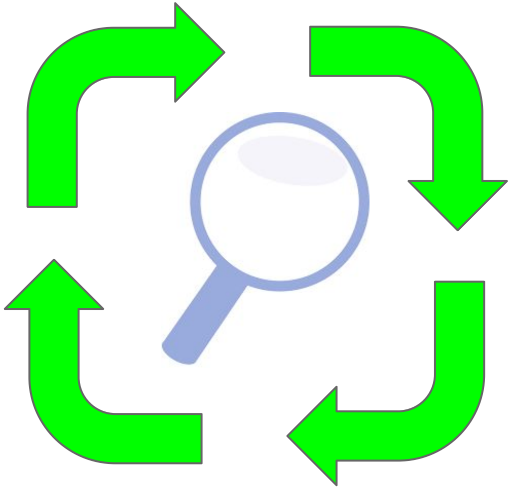
Exceptions

	Location	Gender	Age	Nationality
G1	US	*	*	Greek
E1.1	Seattle	Male	*	Greek
G2	Chicago	Female	*	*

Data understanding during launch and iterate

- Feature-based analysis
- **Data lifecycle analysis**
- Open questions

Launch & Iterate



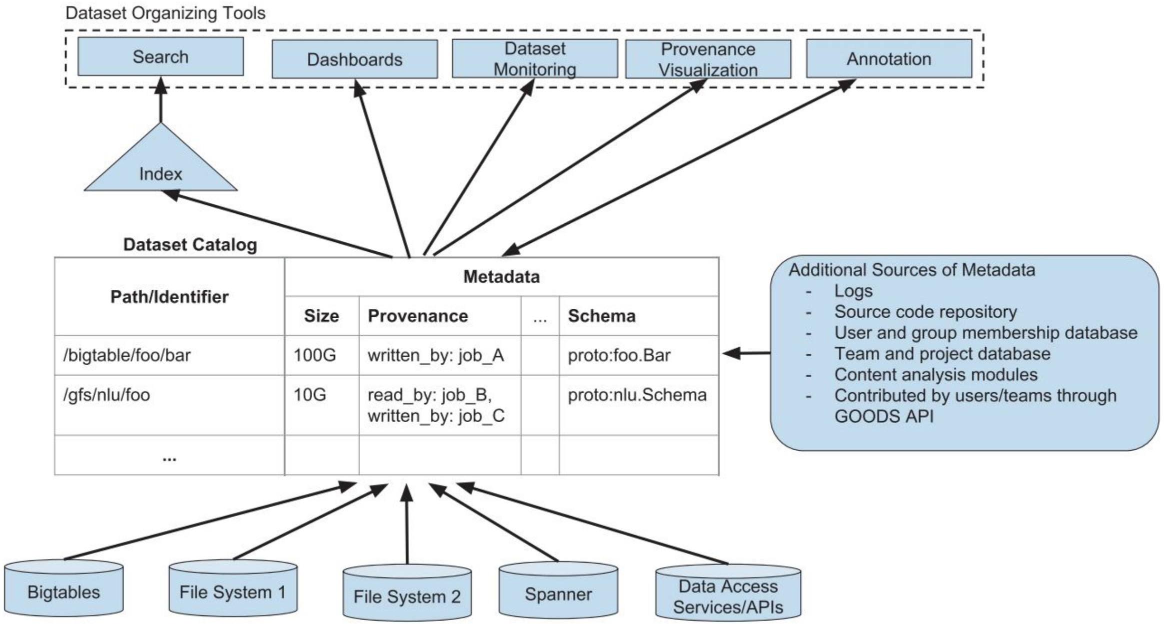
Data lifecycle analysis

- Types of ML analyses
  - Identify dependencies of features
    - E.g., how were the labels generated? Do they “leak” into any other feature?
  - Identify sources of data errors
    - E.g., some examples were dropped because a data source was unavailable
- Provenance and metadata analysis tools are effective
  - Coarse-grained
    - GOODS [HKN+ SIGMOD16]
  - Fine-grained
    - ProvDB [MAD ArXiv16]
    - ModelHub [MLD+ ICDE17]
    - Ground [HSG+ CIDR17]

Google Data Search (GOODS)

[HKN+ SIGMOD16]

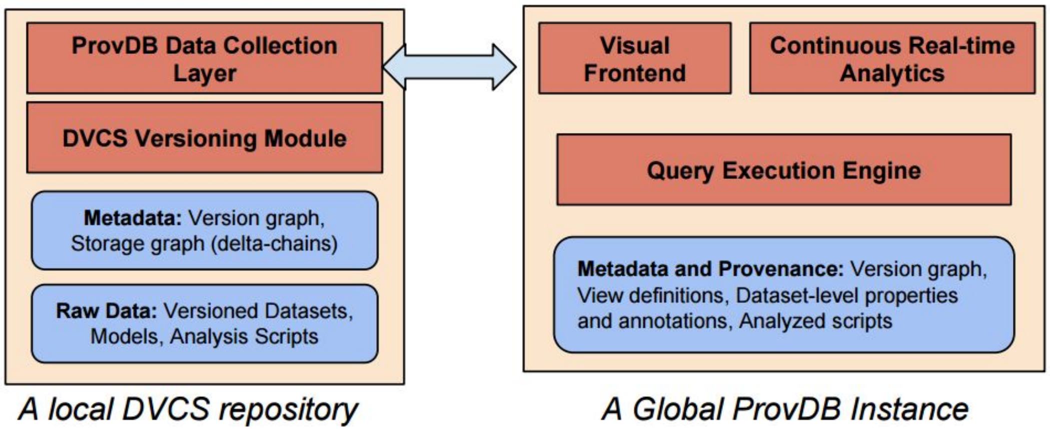
- A system to help users discover, understand, share, and track datasets post-hoc.
- Research question: how to track fine-grained provenance of features?



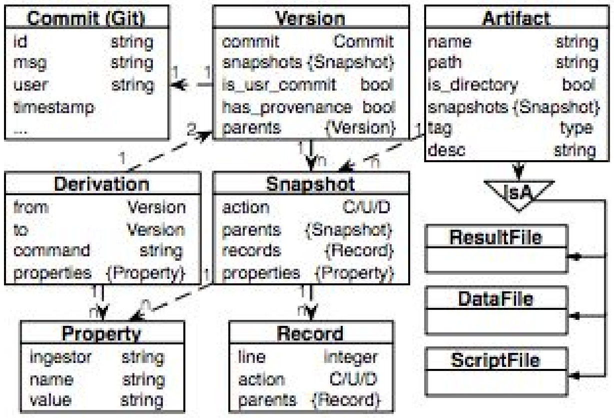
ProvDB: A system for lifecycle management

[MAD ArXiv16]

- A unified provenance and metadata management system to support lifecycles of complex collaborative data science workflows
  - ModelHub: lifecycle management for deep neural networks [MLD+ ICDE2017]
  - Ground: similar goal, but with a simple, flexible metamodel that is model agnosic [HSG+ CIDR17]
- Research question: how to minimize the maintenance overhead?



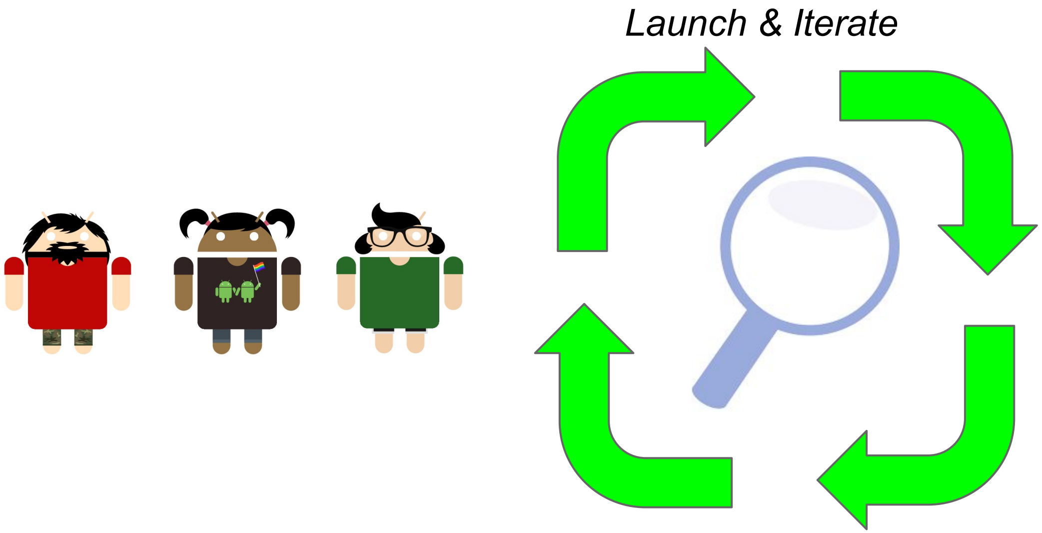
<Architecture>



<Data Model>

Data understanding during launch and iterate

- Feature-based analysis
- Data lifecycle analysis
- **Open questions**



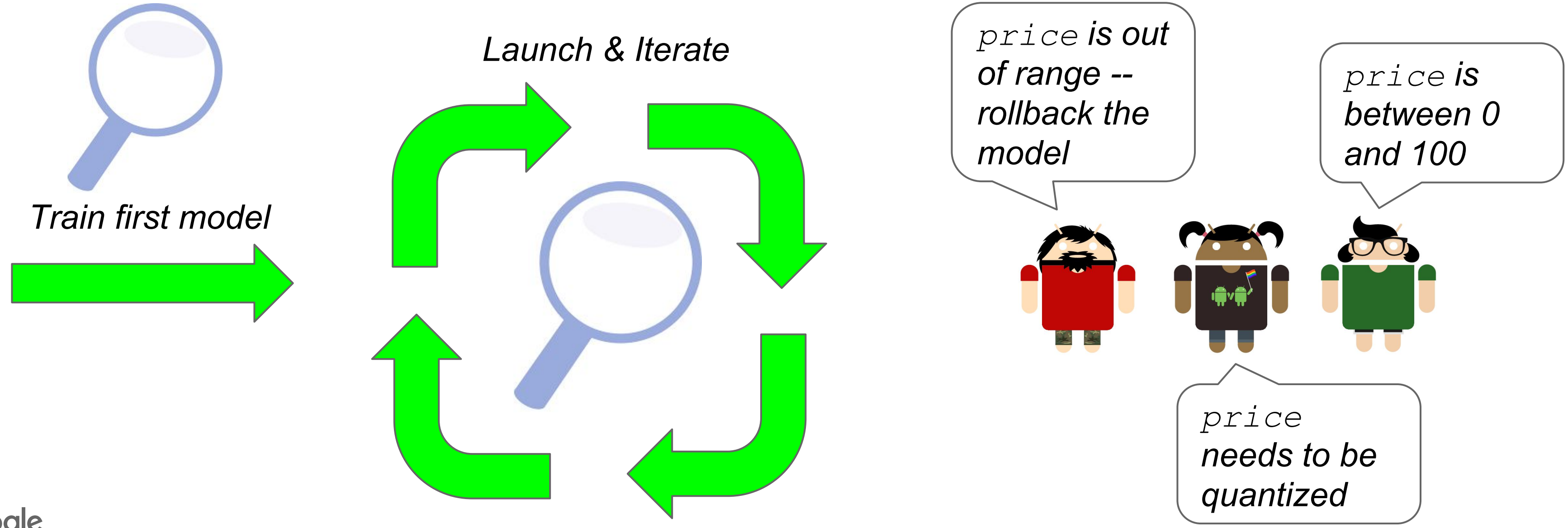
Open questions for ML analysis

- Determine if the model is “fair” [RR KER13]
  - E.g., is a model prejudiced against certain classes of data?
  - Model is only as good as its training data, so need to understand if the data reflects reality
- Identify new kinds of “spam” [GSS ArXiv15]
  - E.g., are users abusing the system in an adversarial way
  - Need to apply adversarial testing on the training data

While SQL [MGL+ PVLDB10, AMP+ Eurosys13] is an “escape hatch” for analysis, can we do better?

Data understanding summary

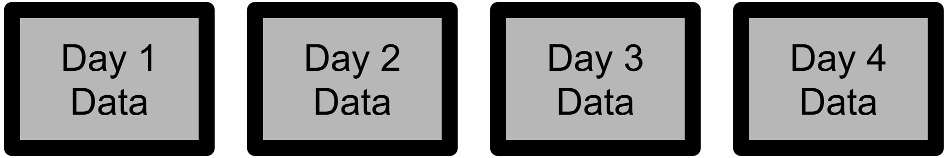
- Need data understanding for sanity checks and launch and iterate
- Existing tools (visualization, data cube analysis, provenance and metadata, and SQL) are helpful, but many ML challenges remain



Data Validation

What if...

- `country` goes from capitalized to lower case?
- Document `age` goes from days old to hours old?
- `document_title` simply disappears?



[FH 76,ACD+16]

What if `country` goes from capitalized to lower case?

Monday	Tuesday	Wednesday	Thursday
US	US	US	us
IN	IN	IN	in
BR	BR	BR	br
CN	CN	CN	cn
	us	us	US
	in	in	IN
	br	br	BR
	cn	cn	CN

Now rare

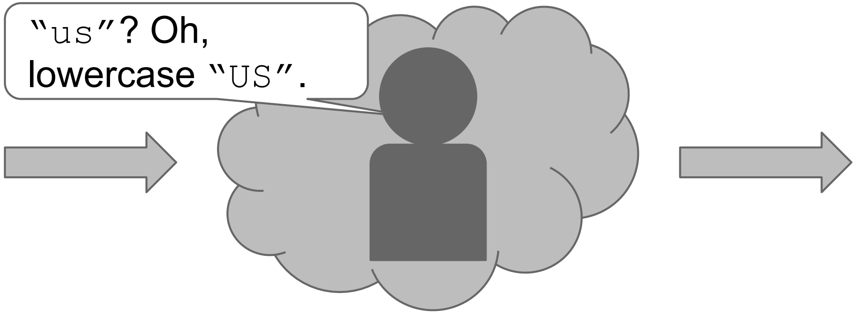
Two different countries

Unknown countries

Rare feature values are hard to learn from.

Models and Data

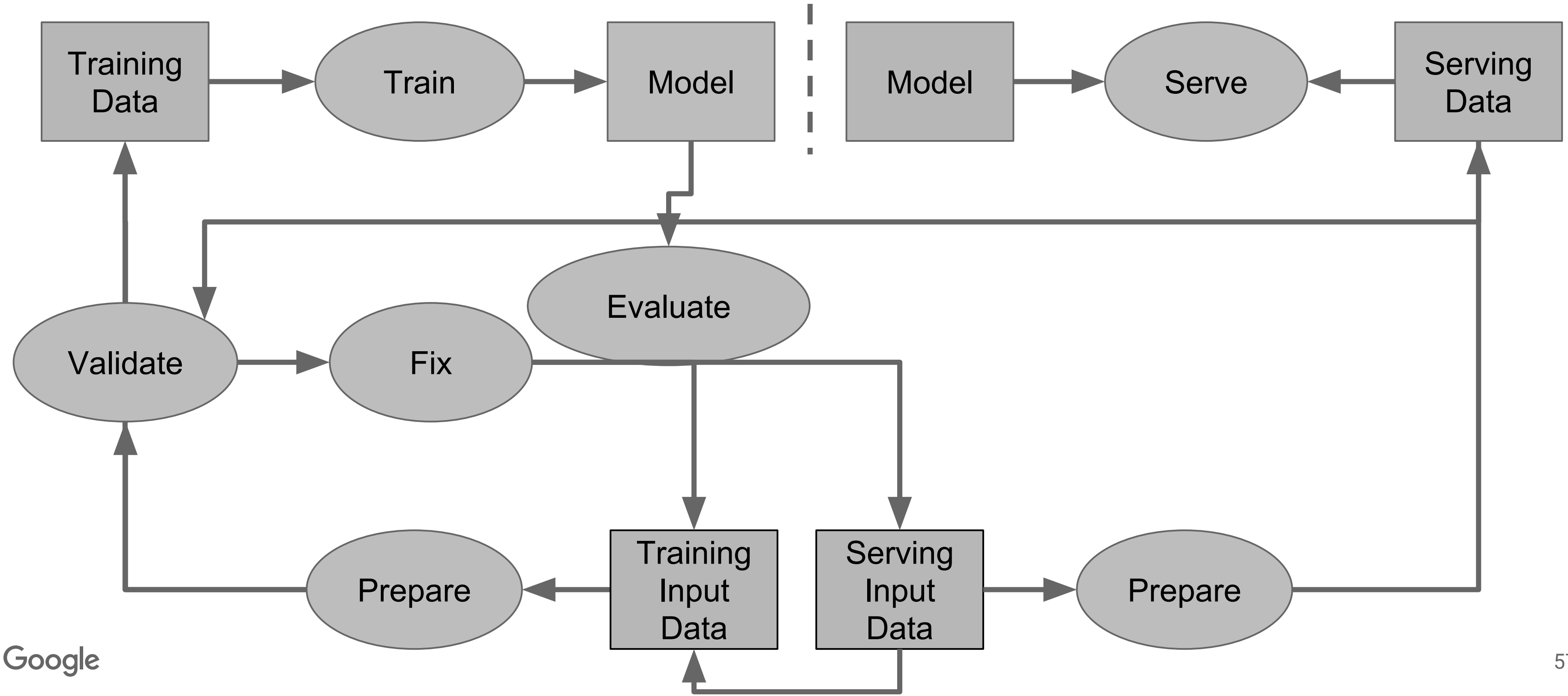
`country="us"`



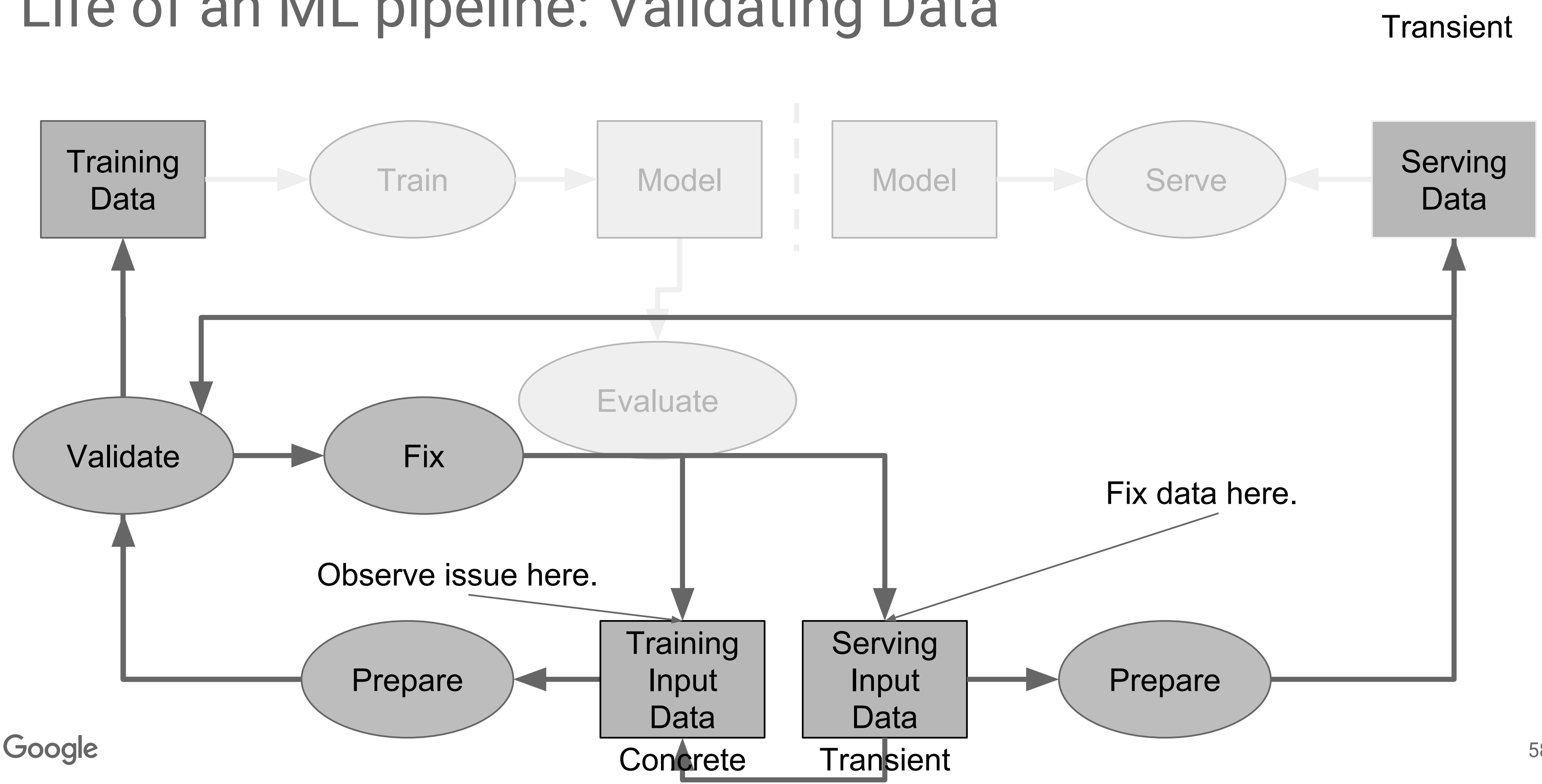
`Pr[Click|country="us"]=0.5`

Models don't answer unasked questions.

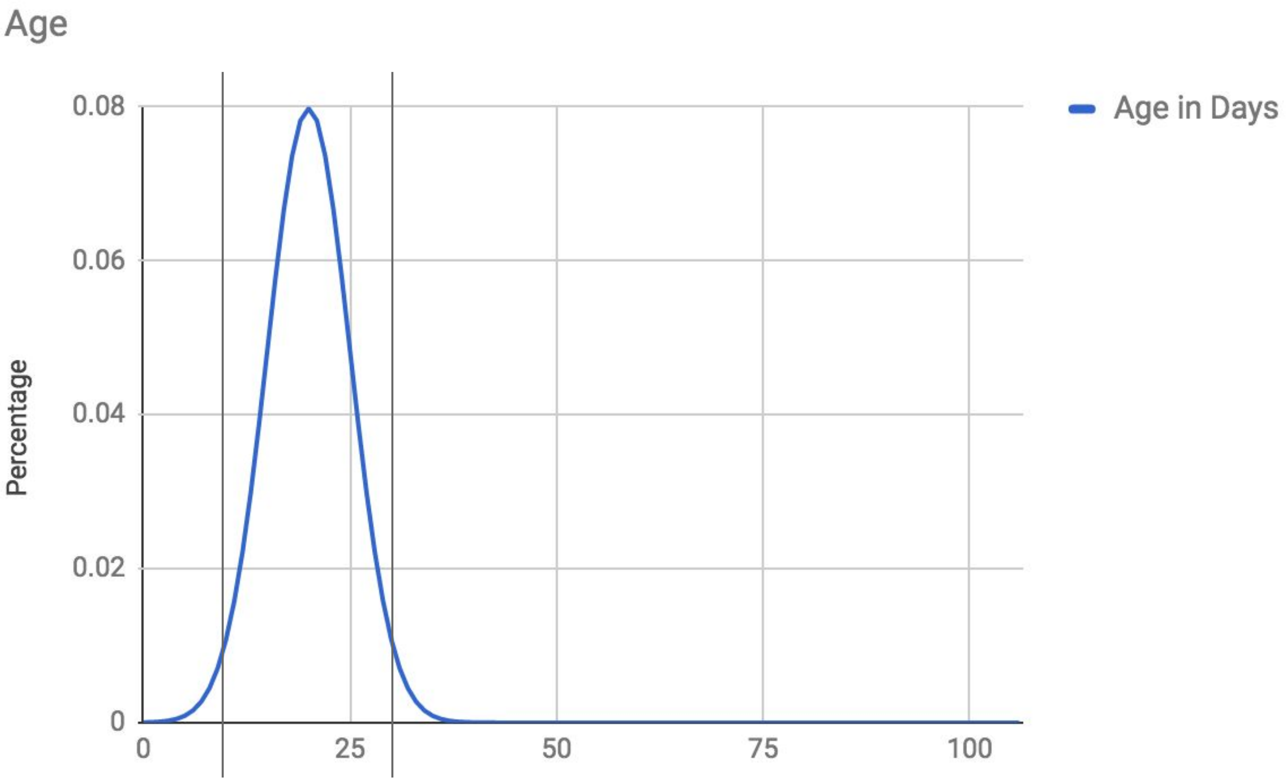
Life of an ML pipeline: Validating Data



Life of an ML pipeline: Validating Data

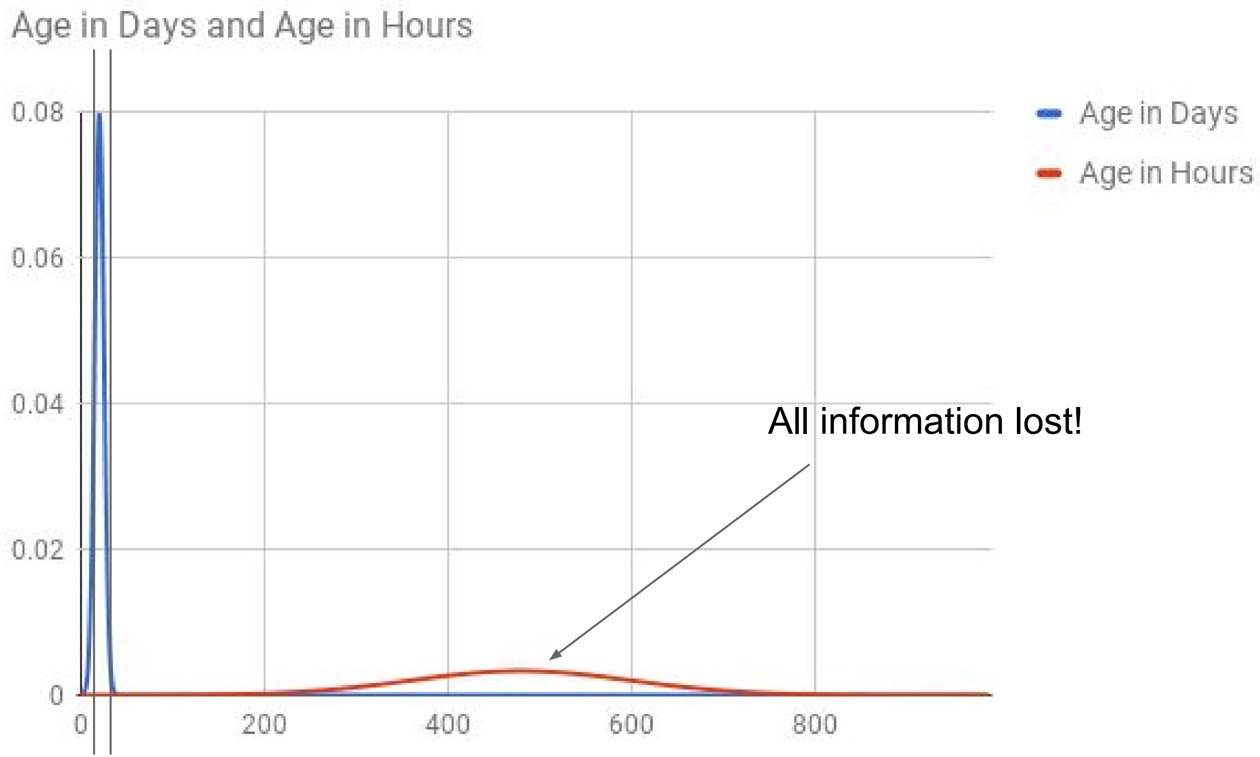


Age of Document



Google

Age of Document

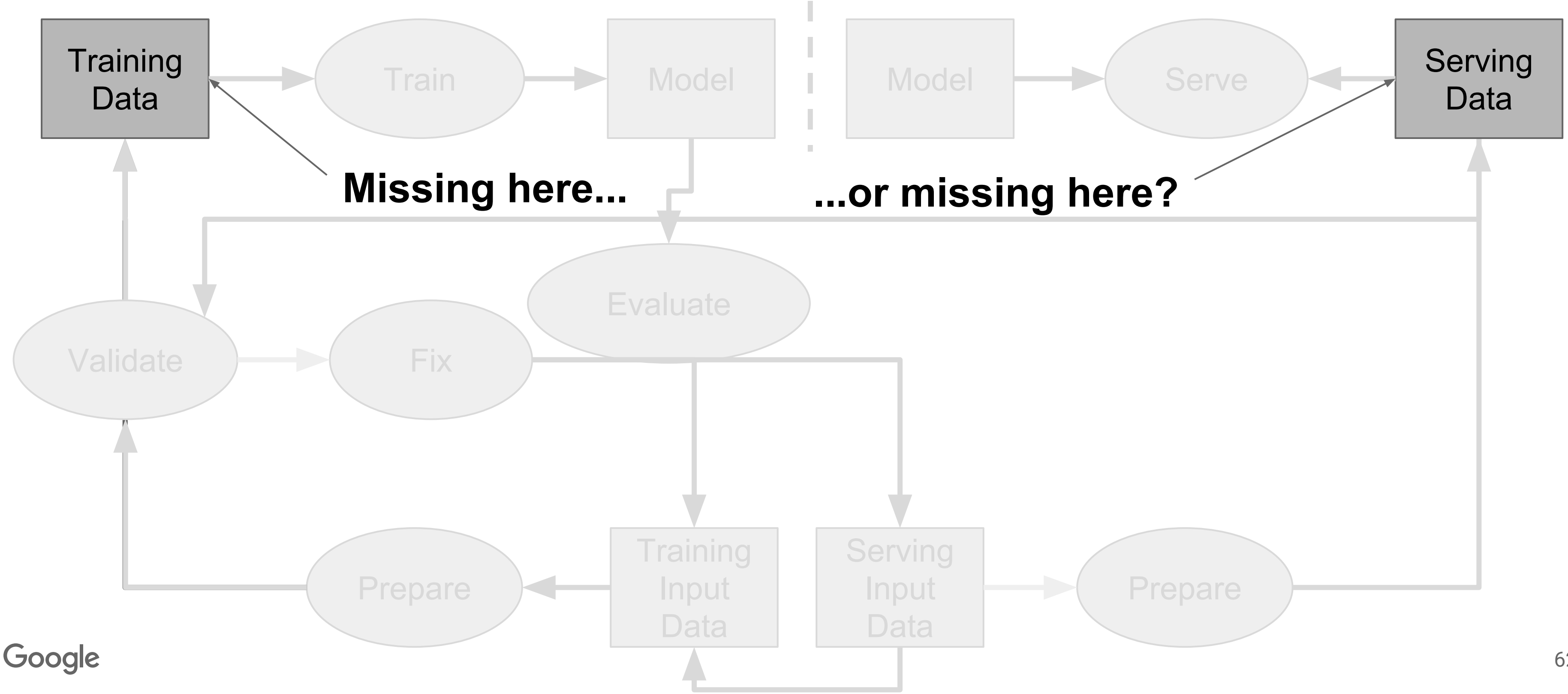


Repair `age`?

Patchy repair: fix winsorization of “`age`”, and throw out all data before shift was made.

Proper repair: throw out “`age`”, and replace it with “`age_in_hours`”

“document_title” Missing



How Do We Deal With These Problems?

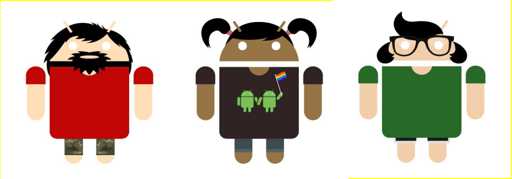
- Automatically insert corrections at serving time (e.g. capitalize all countries)
- Create a new, clean field (e.g. `age_in_hours`)
- Find where a field disappeared (e.g. provenance or root cause analysis on field "`document_title`") (see also Inspector Gadget [OR PVLDB11], Data X-Ray [WDM SIGMOD15], MacroBase [BGM+ SIGMOD17])

We need to detect problems, and in a lot of cases, we need to notify users to solve these problems.

Current Best Practice: Alert + Playbook

- “New values for the field `country’ have appeared. Check that the new values are valid, and where they came from.”
- “The field `age’ is being cropped in 99.99% of the examples. Has the scale of the field changed?”
- “The field `document_title’ is missing from all examples. Earlier, it was pulled in from the table XYZ. Has it been removed from that table?”

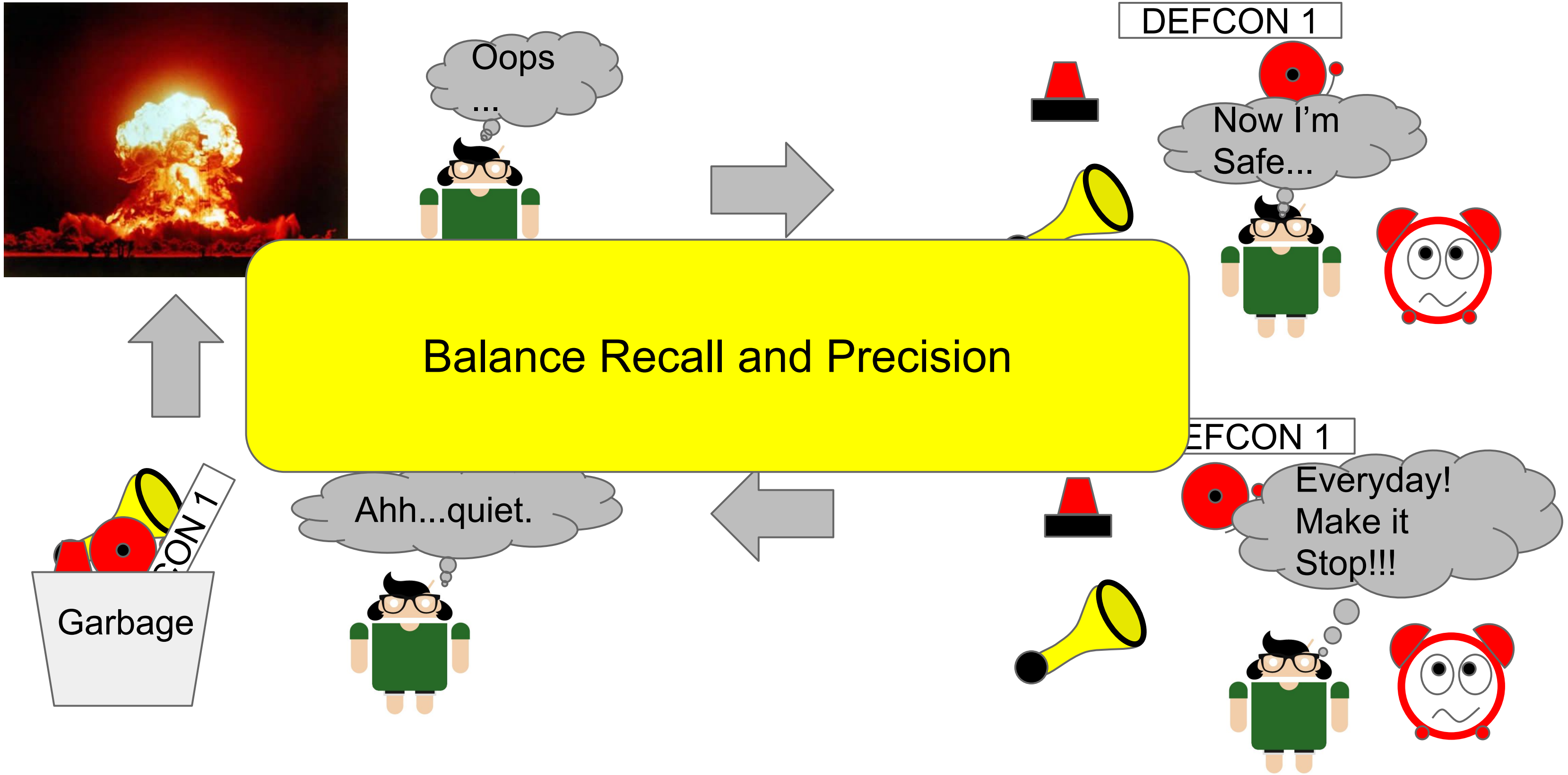
Playbooks are for



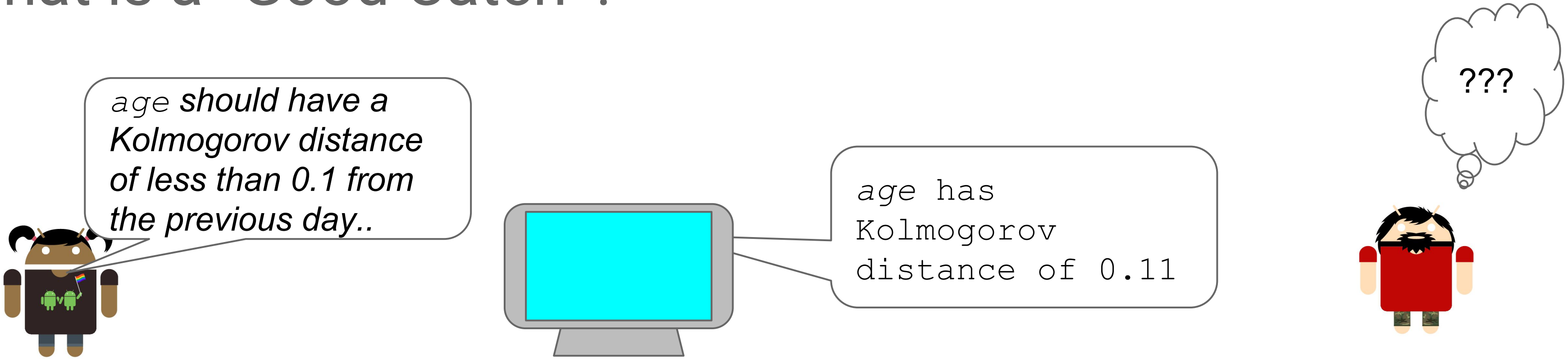
Outline-Data Validation

- Why Data Validation?
  - Models cannot answer questions they are not asked.
  - Automated fixes would be great, but are hard.
  - Current Best Practice: Alert + Playbook
- What about People?
- What Alerts?

A Common Scenario



What is a “Good Catch”?



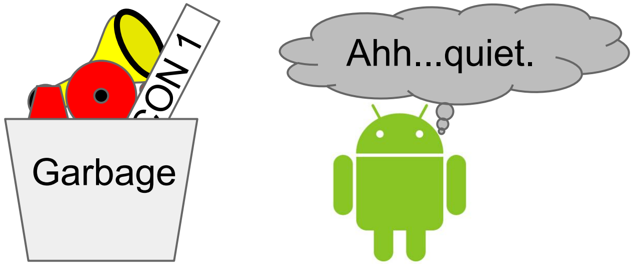
The question is not whether something is “wrong”.
The question is whether it gets fixed.

Question Everything

Monday
US
IN
BR
CN

Tuesday
US
IN
BR
CN
SS

[CM11,BIG+ICDE13]

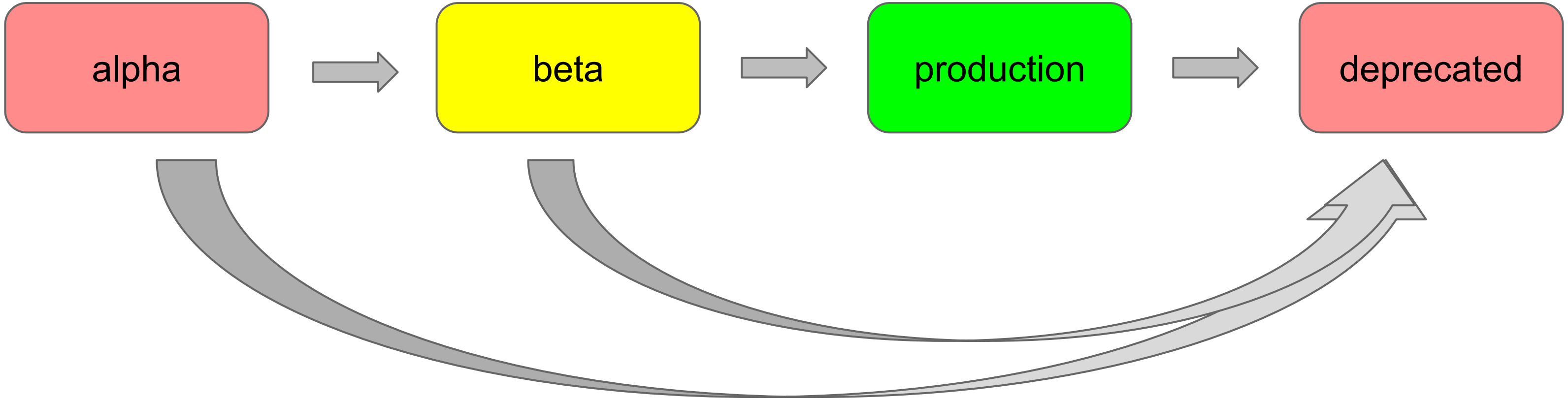


Question the constraint AND the data.

Combining Alerts

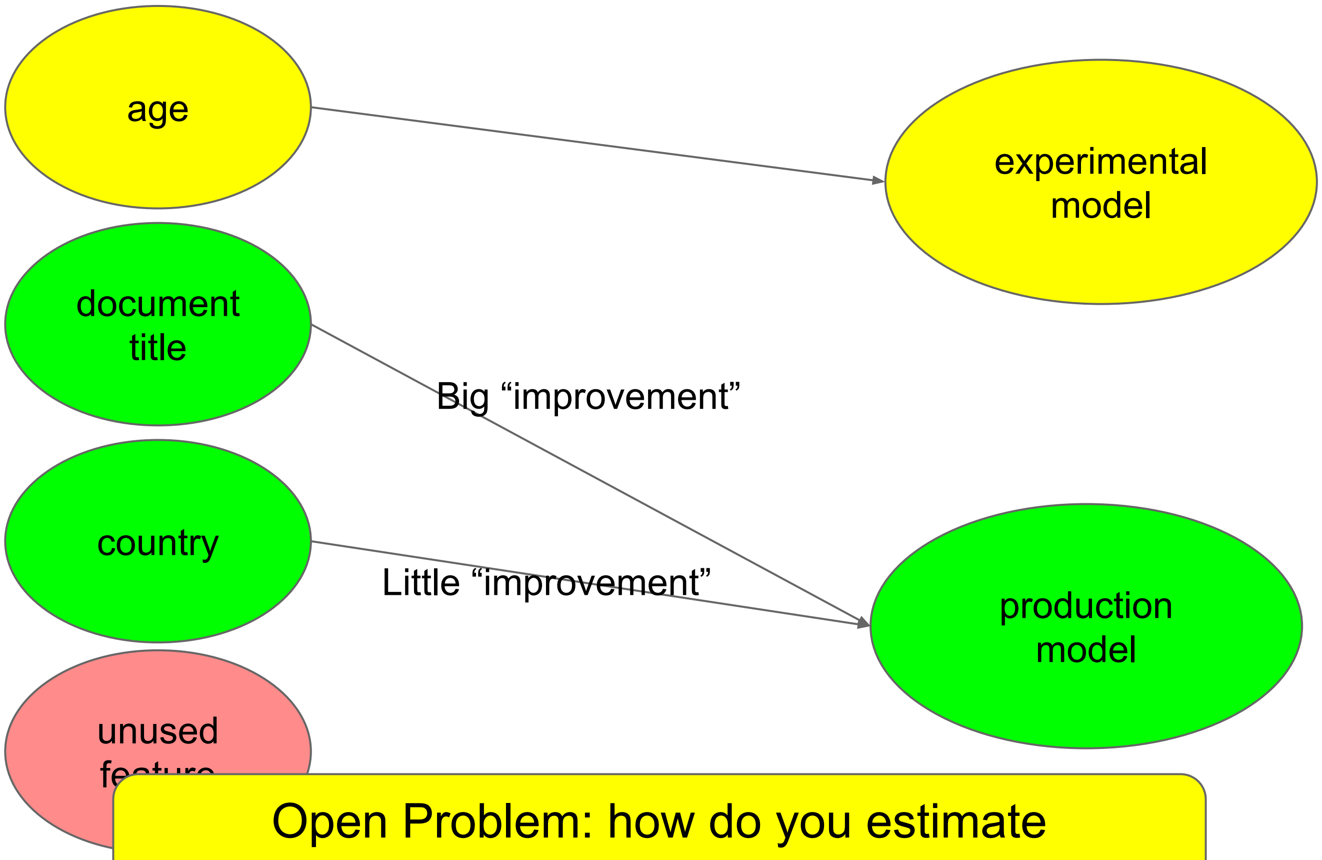
- When there are multiple alerts, what do you do first? How do you decide if they are related, and if so what the root cause is?
- Combining repairs
  - Open area of research [ACD+PVLDB16]
  - Cost-Based Models [BFF+SIGMOD05]
  - Conflict Hypergraph [KL ICDT09,CIP ICDE13]

Lifecycle of Fields



Focus on alerts for data that is used.

Impact



Open Problem: how do you estimate improvement without making a correction?

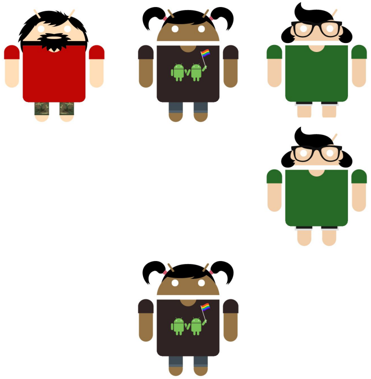
Combining Alerts

MORE ACTIONABLE

The field “`document_title`” is missing.

The field “`country`” has new values.

The distribution of values for the field “`age`” changed.



LESS ACTIONABLE

Rank alerts from most actionable to least actionable.

Outline-Data Validation

- Why Data Validation?
  - Models cannot answer questions they are not asked.
  - Automated fixes would be great, but are hard.
  - Current Best Practice: Alert + Playbook
- What about People?
  - Balance recall and precision.
  - A good catch is one that leads to a fix.
  - Understand how fields are being used.
  - Prioritize alerts by impact/actionability.
- What Alerts?

Continuous Data Cleaning

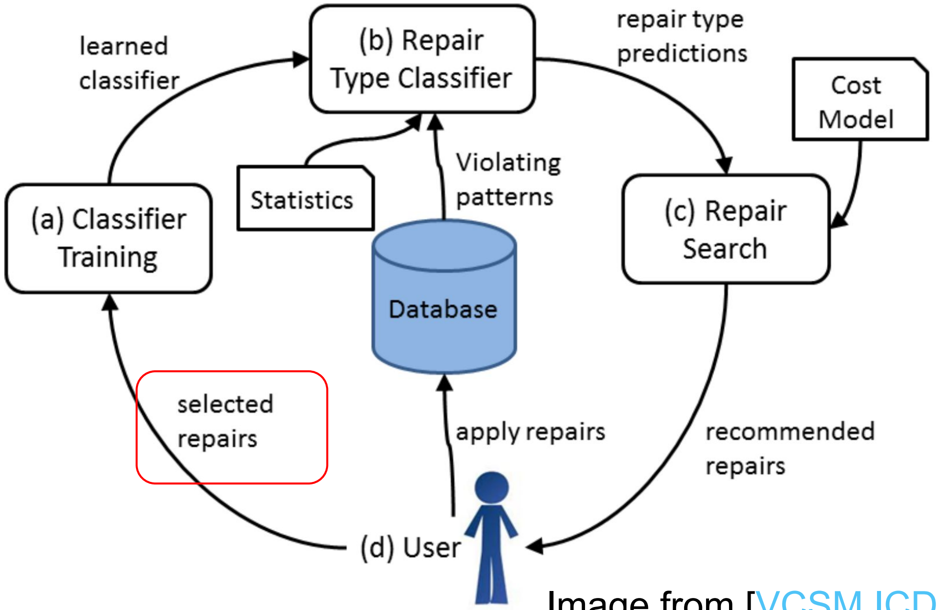
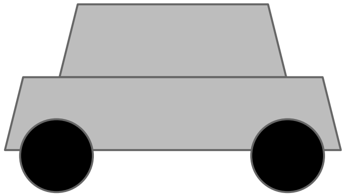
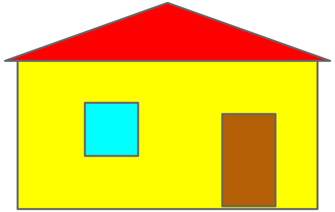
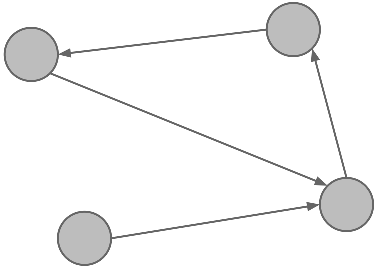


Image from [VCSM ICDE14]

Generic Alerts are Hard To Design





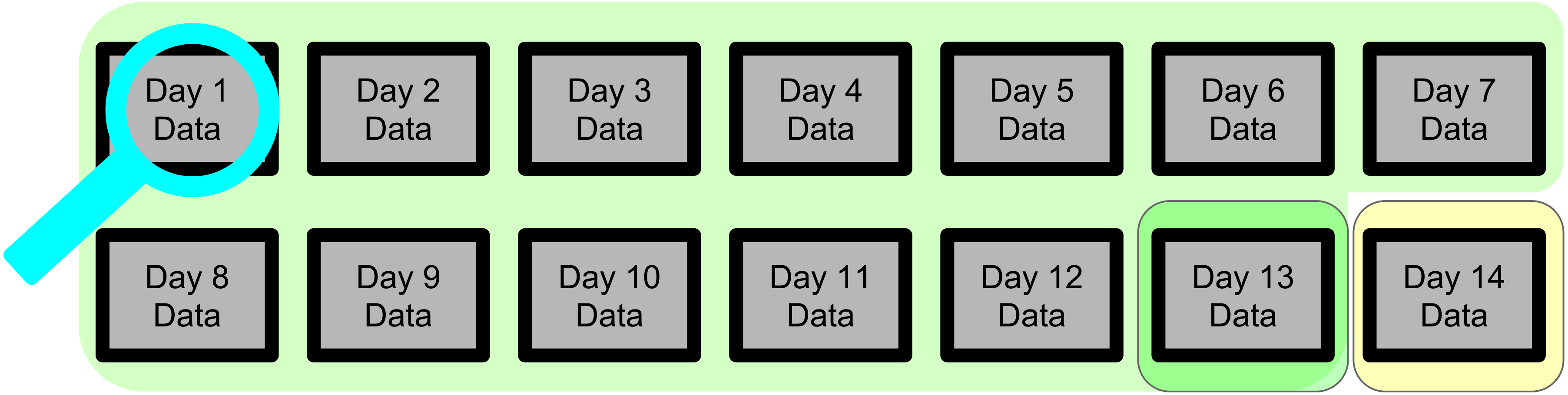


Continuously Arriving Training Data

Day 1 Data	Day 2 Data	Day 3 Data	Day 4 Data	Day 5 Data	Day 6 Data	Day 7 Data
Day 8 Data	Day 9 Data	Day 10 Data	Day 11 Data	Day 12 Data	Day 13 Data	Day 14 Data

Give new data a priority

Continuously Arriving Training Data



Control

Treatment

Compare new data to old data

Alerts Motivated By Engineering Problems

- Missing fields
- RPC Timeout
- Format changes

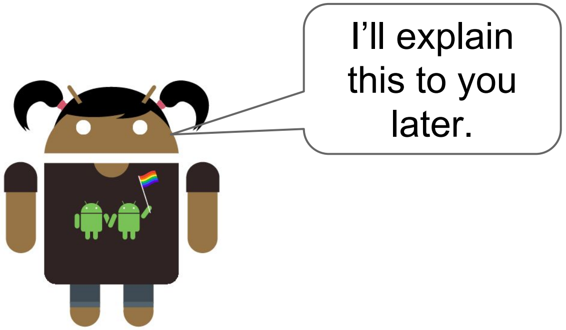
Alerts Motivated By Engineering Problems

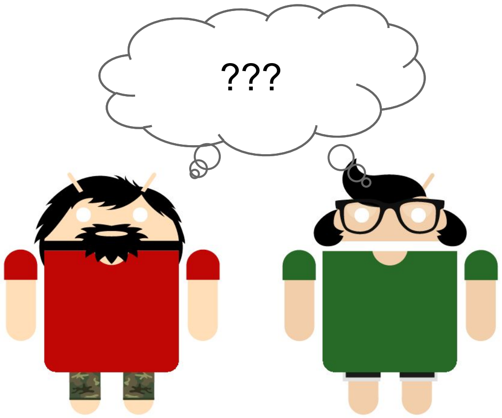
- Missing fields
  - Check if a field that was present is now absent.
- RPC Timeout
  - Check the most common value is not more common than before.
- Format changes
  - Check if the domain of values has increased.

Use common software engineering problems to design baseline checks.

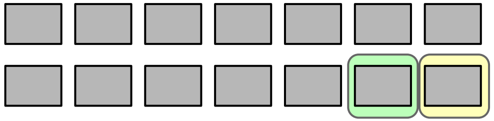
A Statistics Approach

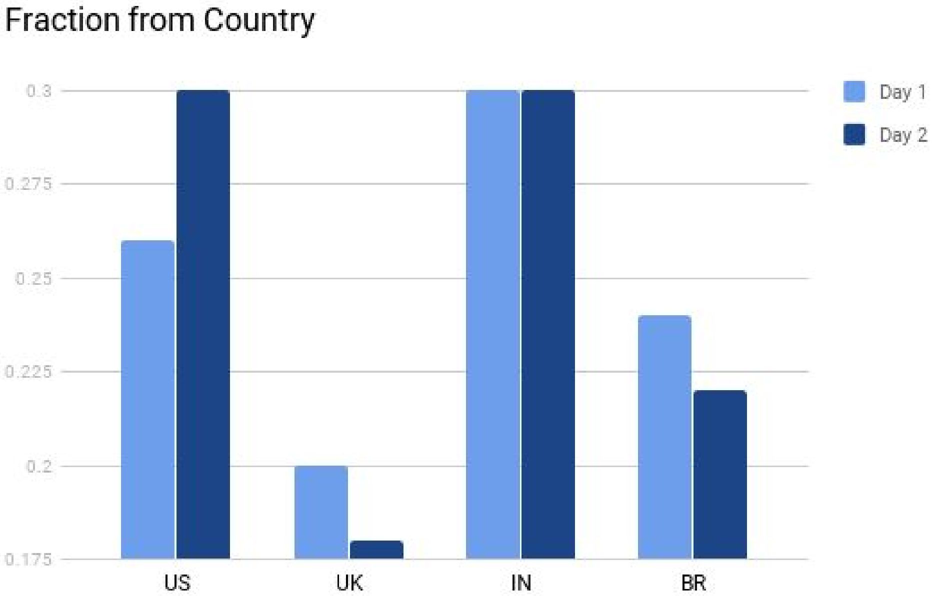
- Homogeneity tests, Analysis of variance (ANOVA)
- Time series analysis, Change Detection



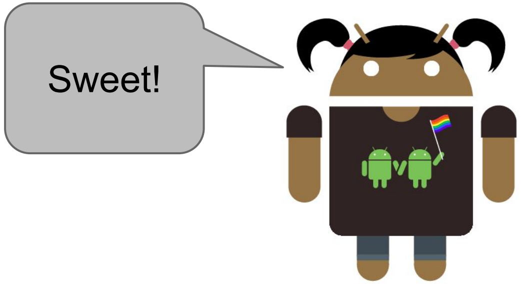


Catch “all” Statistical Measures for Data as it Arrives





Chi-Squared test for homogeneity [P00]: reject the null hypothesis for the distributions being the same.
ANOVA: analysis of variance ([F 21,F 25])

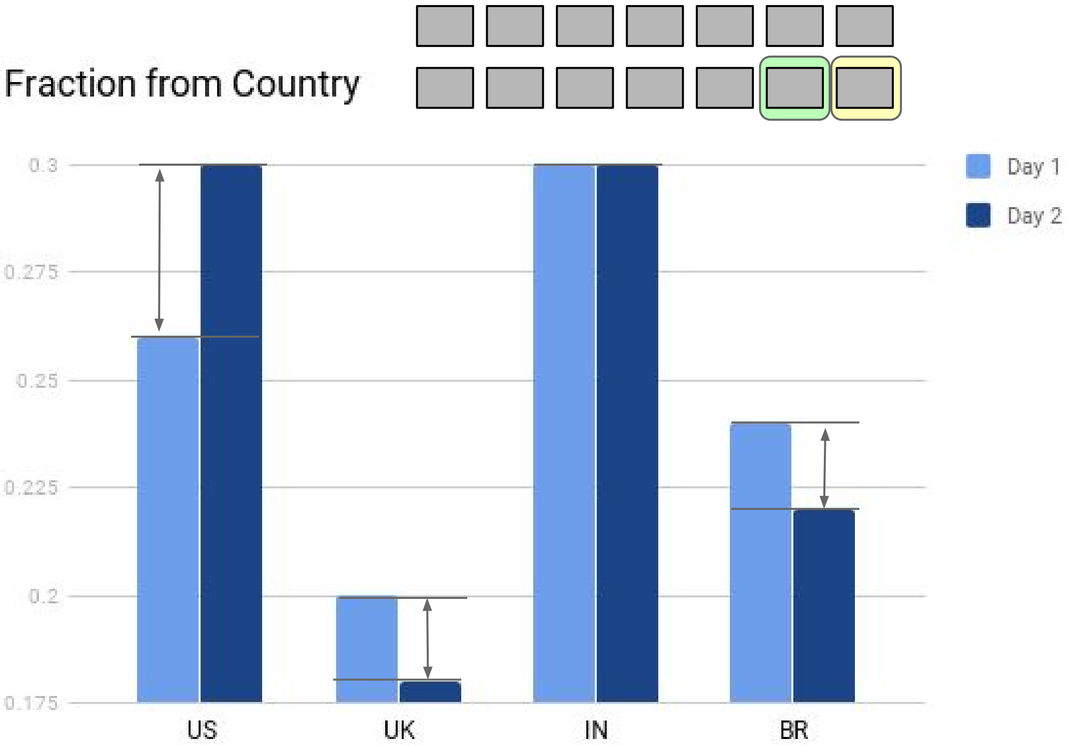


ML Expert
/Stats Expert

Problems with the Chi-Squared Statistic

- Statistically significant changes between days are common in big data.

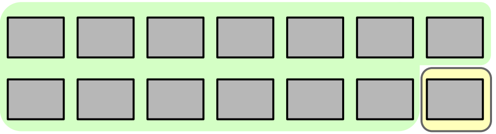
Catch "all" Measures for Data as it Arrives

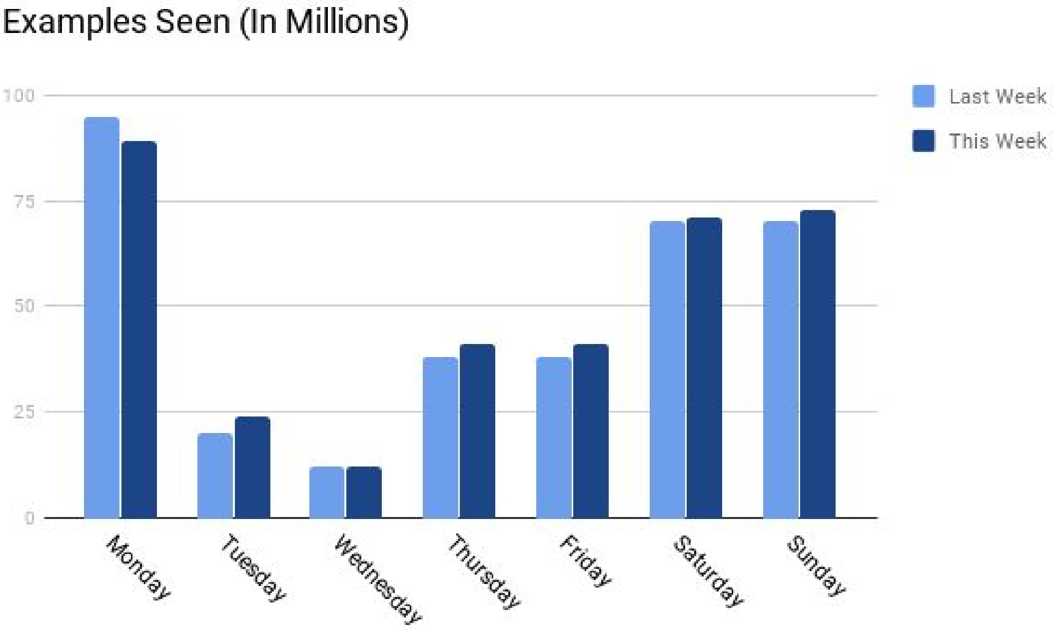


L1 Metric/total variance
L-infty Metric
Earth Mover's Distance
[GS 02,VRM+ VLDB15]

$$L_1(\mathcal{D}_1, \mathcal{D}_2) \leq \epsilon$$

Time Series Analysis/Change Detection





Use on critical metrics of data, (number of examples, number of positives), not everything.

[BN 93,DTS+VLDB08,BGK+AAS15]

Outline-Data Validation

- Why Data Validation?
  - Models cannot answer questions they are not asked.
  - Automated fixes would be great, but are hard.
  - Current Best Practice: Alert + Playbook
- What about People?
  - Balance recall and precision.
  - A good catch is one that leads to a fix.
  - Understand how fields are being used.
  - Prioritize alerts by impact/actionability.
- What Alerts?
  - Alerts motivated by engineering problems.
  - Alerts that bound drift, but acknowledge its existence.
  - Time series for critical metrics like the number of examples.

Future Work

- What alerts are best?
- Impact Analysis: If I fix this, how will the system improve?
- Automatically Generated Playbooks + Automatically Generated Fixes

Future Work

```
Class IntegerTest {
  // Test that parsing "-4" yields -4.
  @Test
  void testParseInt() {
    int actual = Integer.parseInt("-4");
    // Throws AssertionError on failure.
    Assert.assertEquals(
      "Failed to parse negative" // message
      -4,                        // expected
      actual);                   // actual value
  }
}
```

○

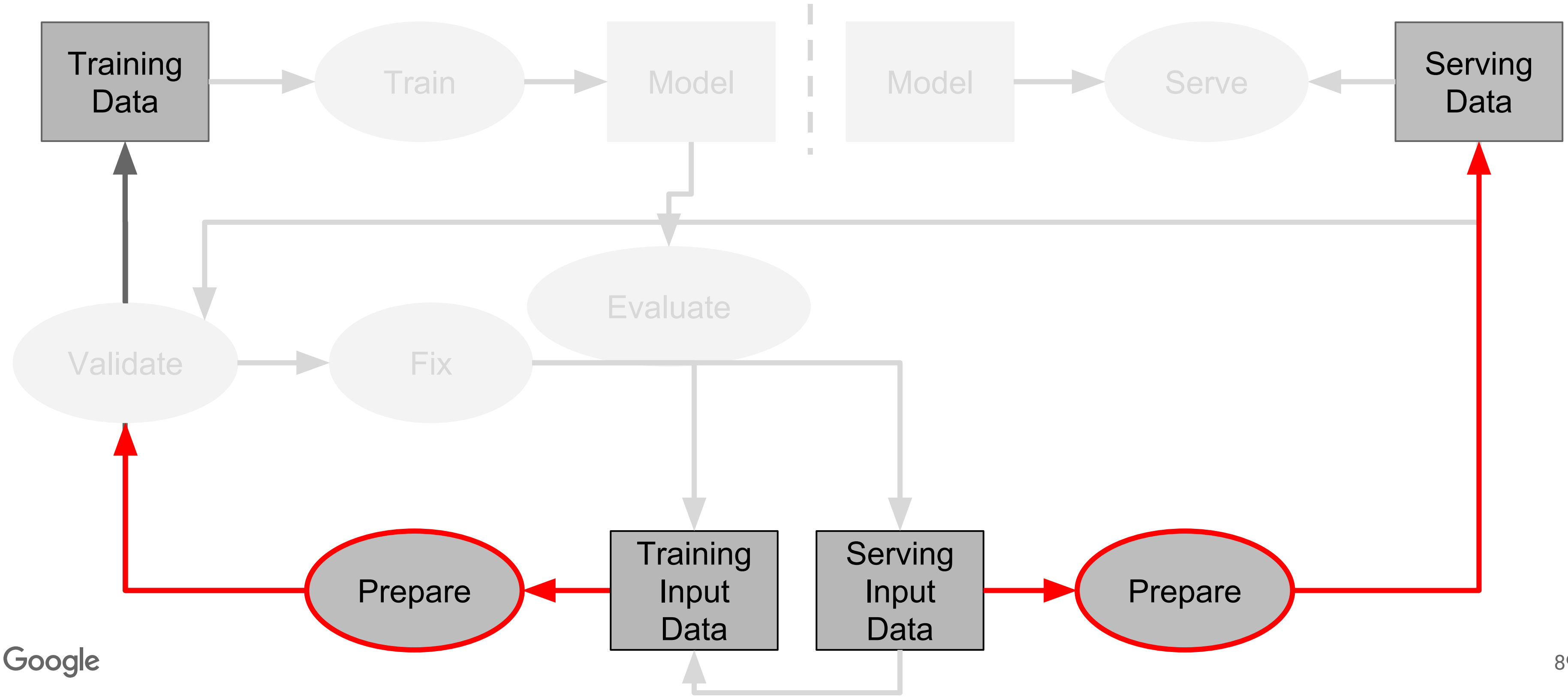
We need JUnit for Data Validation for Machine Learning

- Quick to write alerts/playbooks
- Easy to understand/update alerts
- Useful enough to catch errors
- **Improves the overall speed of innovation**

Google

Data Preparation

Life of an ML pipeline: Preparing the data



What is data preparation?

- Feature engineering
  - “.. difficult, time consuming, requires expert knowledge.” -- *Andrew Ng*
  - Involves trial-and-error

- Adding new attributes or examples to training data
  - Looking for external data sources to complement training data
  - More data not necessarily good

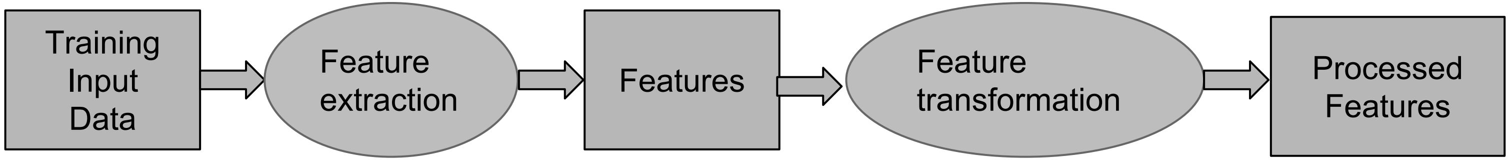
What is data preparation?

- **Feature engineering**
  - ".. difficult, time consuming, requires expert knowledge." -- *Andrew Ng*
  - Involves trial-and-error

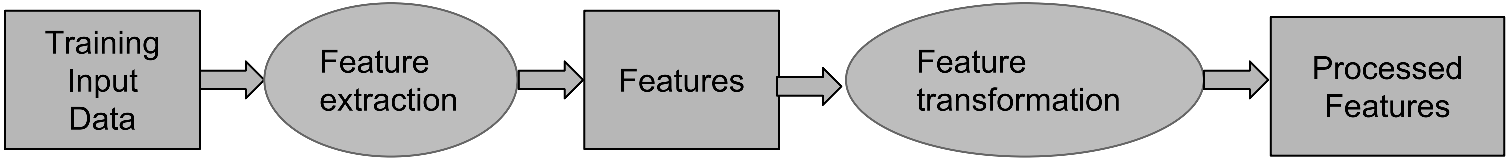
- Adding new attributes or examples to training data
  - Looking for external data sources to complement training data
  - More data not necessarily good

Google

Feature Engineering

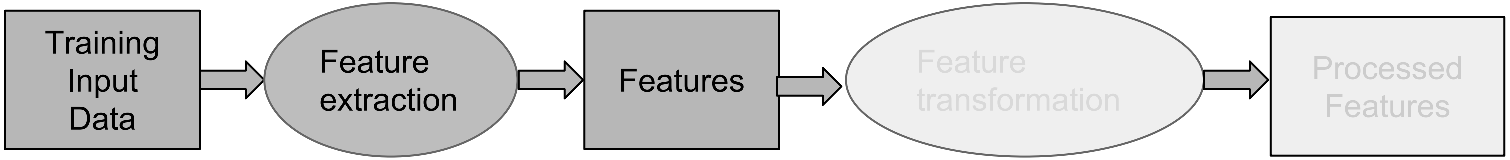


Feature Engineering - An example

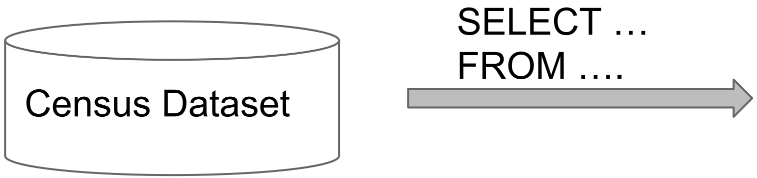


Objective: predict median housing price, at the granularity of city blocks.

Feature Engineering - An example



Objective: predict median housing price, at the granularity of city blocks.




```
{
  latitude:          118.7
  longitude:         35.6
  households:        532
  housing_age:       43
  crime_rate:        LOW
  median_price:      872909
}
```

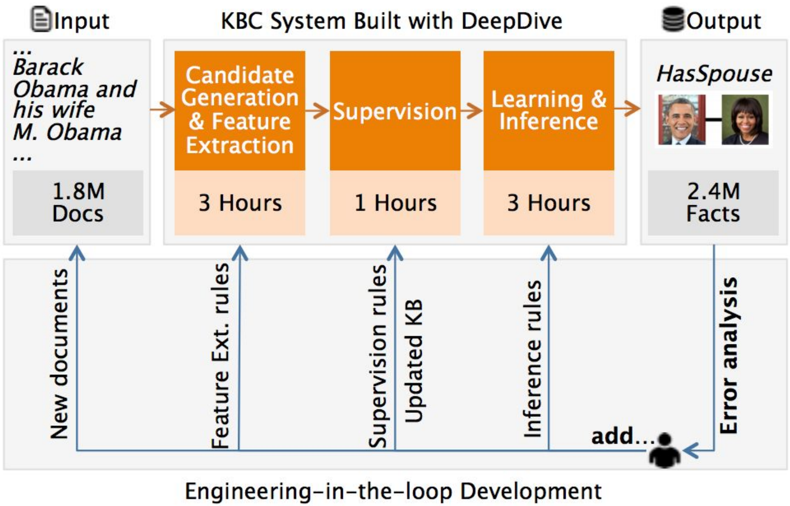
Google

Tools and techniques - extract data programmatically

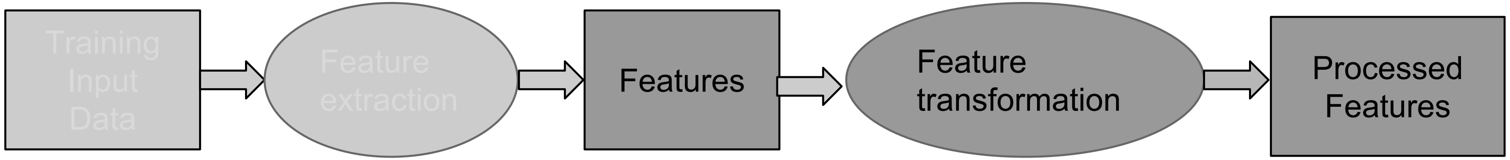
[ESR+ HILDA16,RSS+ TCDE14]

- Instead of generating a small high-quality dataset, programmatically generate a large low-quality dataset.

- Use feature engineers to tune extractors to improve quality.

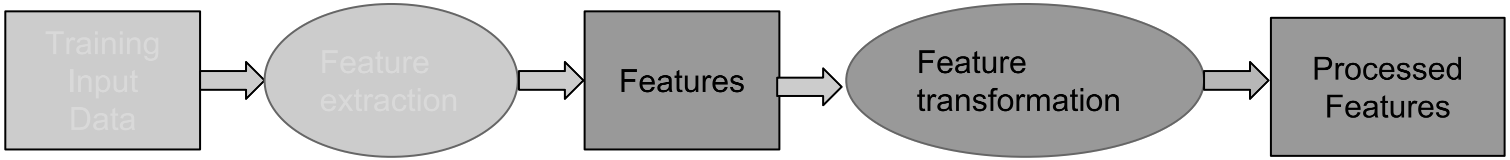


Feature Engineering - An example



Objective: predict median housing price, at the granularity of city blocks.

Feature Engineering - An example



Objective: predict median housing price, at the granularity of city blocks.

```
{
  latitude:             118.7
  longitude:            35.6
  households:           532
  housing_age:          43
  crime_rate:           LOW
  median_price:         872909
}
```

```
{
  latitude:     118.7
  longitude:            35.6
  households_bucket:          5
  housing_age:      43
  crime_rate_low:       1
  crime_rate_high:      0
  crime_rate_med:       0
  crime_rate_unknown:   0
  median_price:         872909
}
```

Bucketization

One-hot encoding

Typical feature transforms

- Standard set of techniques for feature transformation
  - Normalization
  - Bucketization
  - Winsorizing
  - One-hot encoding
  - Feature crosses
  - Use a pre-trained model or embedding to extract features [MCC+ ArXiv13]

- Exact feature transform required depends on both data as well as the ML training algorithm
  - Some algorithms may be able to do some of the transforms natively

Why not learn to engineer features?

- Feed training data directly to a deep neural network and let it figure out the features
  - Generally referred to as “representation learning” in the ML community
  - Some promising techniques like autoencoders, restricted Boltzmann Machines exist [BCV+ TPAMI13]

- Learning both the representations and the objective can require a lot of resources and data
  - Engineering features still required in most cases

Takeaways

- Feature engineering requires domain knowledge and involves trial-and-error
  - Invest in tools to make design and experimentation easier [RSS+ TCDE14, AC ICDE16, ESR+ HILDA16,]
- Designing good features is hard and time-consuming
  - Invest in tools and infrastructure that allow sharing, understanding, and maintenance of features
- Open question: Given an input set of features and the ML training algorithm, generate suitable feature transforms automatically
  - From our experience, this is "pain point" for users who do not necessarily understand the nuances of transforms

What is data preparation?

- Feature engineering
  - “.. difficult, time consuming, requires expert knowledge.” -- *Andrew Ng*
  - Involves trial-and-error

- **Adding new attributes or examples to training data**
  - Looking for external data sources to complement training data
  - More data not necessarily good

Adding more features

Scenario:  wants to improve the prediction accuracy. She decides to add other features (average per capita income, population density, etc.) to the training data.

Challenges for

-  : Which features will improve model performance the most?
-  : How do I add a feature to an existing pipeline? Will it be available at serving time? Am I allowed to use it? What is the ROI for adding this feature?
-  : This introduces new dependency. How can I make sure that the pipeline is robust? What will be the effect on model size and prediction latency?

Adding more features

Scenario:  wants to improve the prediction accuracy. She decides to add other features (average per capita income, population density, etc.) to the training data.

Steps:

-  struggles to find data that she can “add” to her training data. She experiments and decides to add `median_per_capita_income` as an additional feature.
-  ensures that this feature is available for all training data as well as at serving time.
- Train an experimental model, evaluate it offline as well as online (on 1 % traffic)
- She also does model analysis to understand the impact of this feature
- She launches the new model!

Add more examples

Scenario: You find your initial training data does not have good coverage for a slice of the data. You need more examples for that slice.

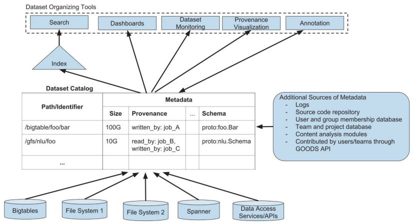
Challenges for 

- Where can I find training data for this slice?

Tools and techniques - Finding data

- Organizations often have a large number of datasets siloed within product areas.

GOODS



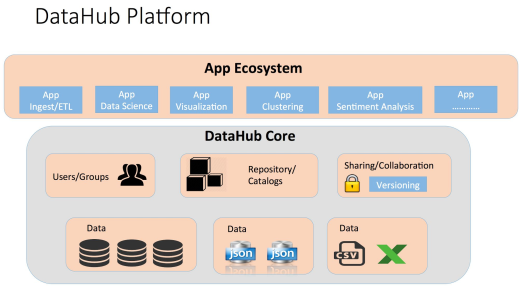
[HKN+ SIGMOD16]

Ground

ground

[HSG+ CIDR17]

Datahub

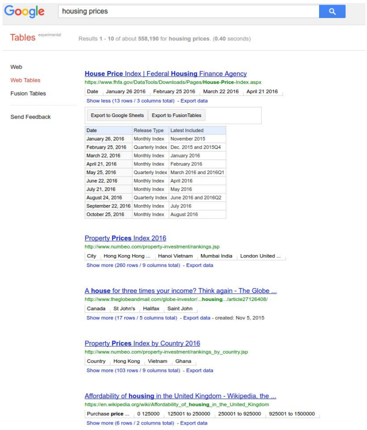


[BBC+ CIDR15]

Tools and techniques - Finding data

- Over the web, many scientific datasets are published independently by organizations but no central repository for searching.

Webtables



[CHW+ VLDB08]

Kaggle

Data Civilizer

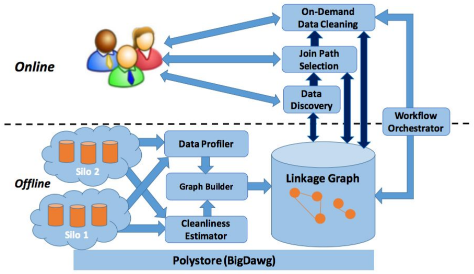


Figure 1: Data Civilizer Architecture

Add more examples

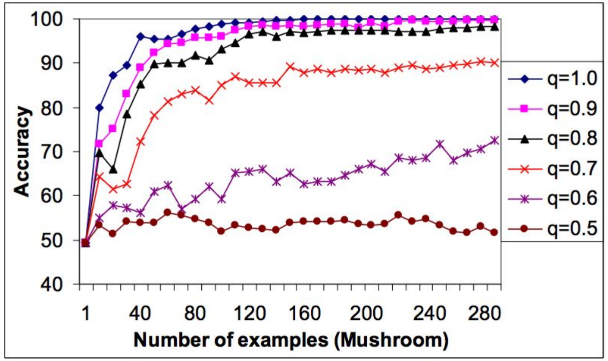
Scenario: Collecting training data may require manually extracting this information from raw data like images, video, speech, and text.

Challenges for 

- Where can I find training data for this slice?
- **How can I extract structured information easily from the raw data?**
- **Crowd-workers are expensive. How do I select and prioritize tasks?**

Tools and techniques - more labels or better labels

- Low-cost labeling can produce noisy data [SPI+ KDD08]
- Improving label quality can give bigger boosts than more examples



- Need tools to help decide whether to get more labels on new data, or multiple labels on the same data.

Tools and techniques - active learning

- Semi-supervised learning technique in which the learning procedure decides and interactively requests labels for examples

- Important when labeling task is complex and expensive

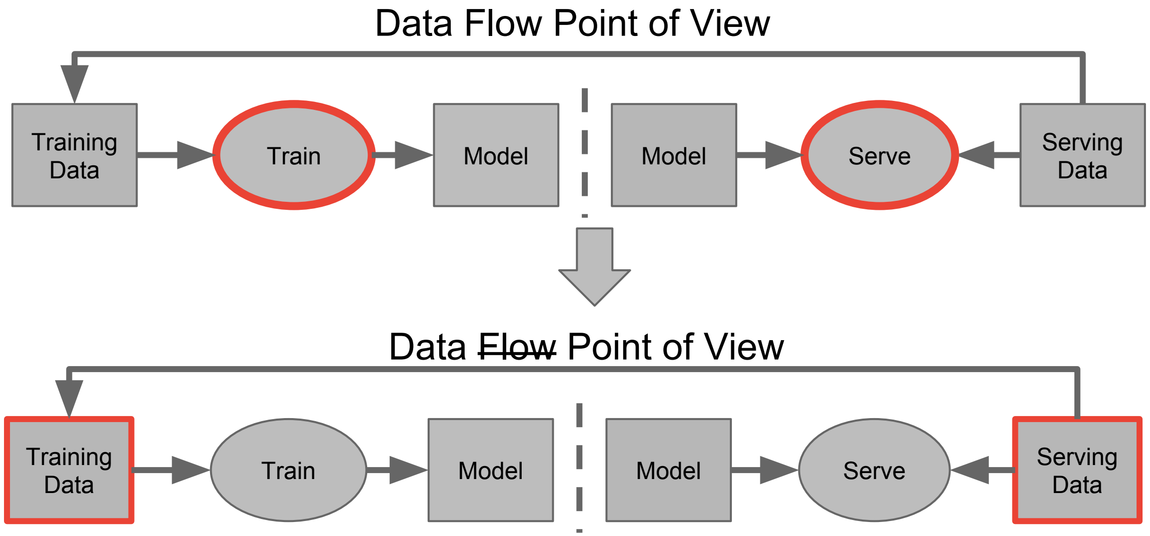
- Well-studied sub-field in machine learning
    - Tutorial on active learning [DL ICML09]
    - Active Learning Survey [S_12]
    - Active learning for NLP [O_09]

Takeaways - adding more attributes and examples

- Adding new features to production machine learning pipelines is a complex process
  - When designing ML systems think of the user journey for feature addition
  - Help users avoid accumulate technical debt [DHG+ SE4ML, KNP+ SIGMOD16]
- Collecting data from training can be hard and expensive
  - Better tooling to make it easier to find, share, and reuse collected data
- Important to help developers understand the trade-off between more data and higher quality data

Parting Thoughts

Lesson 1: Data problems beyond performance optimization



Data management community has a lot to offer and a lot to learn from the machine learning community.

Lesson 2: Be realistic about assumptions you make

- Data does not live in a DBMS; data often resides in multiple storage systems that have different characteristics
- Data life cycle in production ML pipelines is quite complex
- ML is moving fast; keep abreast and apply to the state-of-the-art in ML

Lesson 3: Production ML systems have a diverse set of users

ML Expert

SWE

SRE

Lesson 4: Develop tools that integrate into workflow smoothly

- The launch and iterate cycle time for ML pipelines is small

- To ensure adoption of tools and techniques, it is critical to
  - integrate well into the development workflow
  - make long-term benefits of using it obvious

Check out how we addressed some of these issues!

KDD' 2017

The Anatomy of a Production-Scale Continuously-Training Machine Learning Platform

References

- [AAO+PVLDB16] Z. Abedjan, C. Akcora, M. Ouzzani, P. Papotti M. Stonebraker. “Temporal Rules for Web Data Cleaning”. PVLDB 2016.
- [AC ICDE16] Michael R. Anderson, Michael Cafarella. “Input Selection for Fast Feature Engineering.” ICDE 2016
- [ACD+PVLDB16] Z. Abedjan, X. Chu, D. Deng, R. Fernandez, I. F. Ilyas, M. Ouzzani, P. Papotti, M. Stonebraker, N. Tang, “Detecting Data Errors: Where are we and what needs to be done?”. PVLDB 2016.
- [AMP+ Eurosys13] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, I. Stoica. “BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data”. Eurosys, 2013.
- [BBC+ CIDR15] Anant Bhardwaj, Souvik Bhattacherjee, Amit Chavan, Amol Deshpande, Aaron J. Elmore, Samuel Madden, Aditya G. Parameswaran. “Datahub: Collaborative Data Science & Dataset Version Management at Scale”. CIDR 2015.
- [BCV+ TPAMI13] Yoshua Bengio, Aaron C. Courville and Pascal Vincent. “Representation Learning: A Review and New Perspectives”. TPAMI 2013.
- [BDE+ VLDB16] Matthias Boehm , Michael W. Dusenberry, Deron Eriksson, Alexandre V. Evfimievski , Faraz Makari Manshadi, Niketan Pansare, Berthold Reinwald , Frederick R. Reiss, Prithviraj Sen, Arvind C. Surve, Shirish Tatikonda. “SystemML: Declarative Machine Learning on Spark”. PVLDB 2016
- [BFF+SIGMOD05] P. Bohannon, W. Fan, M. Flaster, R. Rastogi, “A cost-based model and effective heuristic for repairing constraints by value modification”, SIGMOD 2005.

References

- [BGK+ AAS15] K. Brodersen, F. Gallusser, J. Koehler, N. Remy, S. Scott, “Inferring causal impact using Bayesian structural time-series models”, Annals of Applied Statistics, 2015.
- [BGM+ SIGMOD17] P. Bailis, E. Gan, S. Madden, D. Narayanan, K. Rong, S. Suri. “MacroBase: Prioritizing Attention in Fast Data”, SIGMOD 2017.
- [BIG+ICDE13] G. Beskales, I. F. Ilyas, L. Golab, and A. Galiullin, “On the relative trust between inconsistent data and inaccurate constraints,” ICDE 2013.
- [BN 93] M. Basseville, I. Nikiforov. *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall, Inc. 1993.
- [BSK+ CIDR17] C. Binnig, L. De Stefani, T. Kraska, E. Upfal, E. Zgraggen, Z. Zhao. “Towards Sustainable Insights or why polygamy is bad for you”. CIDR, 2017.
- [BVD 10] Bock, Velleman, De Veaux, “Stats: Modeling the World”, Pearson, 2010.
- [CBG+ CIDR15] Daniel Crankshaw, Peter Bailis, Joseph E. Gonzalez, Haoyuan Li, Zhao Zhang, Michael J. Franklin, Ali Ghodsi, Michael I. Jordan. “The Missing Piece in Complex Analytics: Low Latency, Scalable Model Management and Serving with Velox”. CIDR 2015
- [CDG 16] L. Caruccio, V. Deufemia, and G. Polese, “Relaxed Functional Dependencies— A Survey of Approaches”, IEEE TKDE, 2016.
- [CIP ICDE13] X. Chu, I. F. Ilyas, and P. Papotti. “Holistic data cleaning: Putting violations into context”. ICDE 2013.
- [CHW+ PVLDB08] Michael J. Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. “WebTables: Exploring the Power of Tables on the Web”. PVLDB 2008.

References

- [CM ICDE11] F. Chiang and R. J. Miller, “A unified model for data and constraint repair,” ICDE, 2011.
- [DEE+SIGMOD13] M. Dallachiesa, A. Ebaid, A. Eldawy, A. Elmagarmid, I. F. Ilyas, M. Ouzzani, and N. Tang. NADEEF: A commodity data cleaning system. In SIGMOD, 2013.
- [DHG+ SE4ML] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. “Machine Learning: The High Interest Credit Card of Technical Debt”. NIPS 2015
- [DL ICML09] Sanjoy Dasgupta, John Langford. “Active Learning Tutorial” ICML 2009
- [DTS+VLDB08] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, E. Keogh. “Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures.” VLDB 2008.
- [E 02] W. Eckerson. “Data Quality and the Bottom Line: Achieving Business Success through a Commitment to High Quality Data”. Technical report, The Data Warehousing Institute, 2002.
- [EIV 07] Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S., “ Duplicate record detection: A survey”, IEEE Transactions on Knowledge and Data Engineering, 2007.
- [ESR+ HILDA16] Henry R. Ehrenberg, Jaeho Shin, Alexander J. Ratner, Jason A. Fries, and Christopher Ré. “Data Programming with DDLite: Putting Humans in a Different Part of the Loop”. HILDA 2016
- [F 21] R. Fisher, “On the "Probable Error" of a Coefficient of Correlation Deduced from a Small Sample”, Metron, 1921.
- [F 25] R. Fisher. “Statistical Methods for Research Workers”, Oliver and Boyd, 1925.
- [FH 76] Fellegi, I. and Holt, D. “A systematic approach to automatic edit and imputation”, J. Amer. Statist. Assoc. 1976.

Google

References

- [FLM+SIGMOD11] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. “Interaction between record matching and data repairing”, SIGMOD 2011.
- [GS 02] A. Gibbs and F. Su, “On Choosing and Bounding Probability Metrics”, International Statistical Review, 2002.
- [GSS ArXiv15] I.J. Goodfellow, J. Shlens, C. Szegedy. “Explaining and Harnessing Adversarial Examples”. arXiv:1412.6572
- [HKN+ SIGMOD16] A. Halevy, F. Korn, N. F. Noy, C. Olston, N. Polyzotis, S. Roy, S.E. Whang. “Goods: Organizing Google’s Datasets”. SIGMOD, 2016.
- [HSG+ CIDR17] J. Hellerstein et al. “Ground: A Data Context Service”. CIDR, 2017.
- [JGP ICDE16] M. Joglekar, H. Garcia-Molina, A. Parameswaran. “Interactive Data Exploration with Smart Drill-down”. ICDE, 2016.
- [KFC HILDA16] M. Kahng, D. Fang, D. Horng. “Visual Exploration of Machine Learning Results using Data Cube Analysis”. HILDA, 2016.
- [KL ICDT09] S. Kolahi, L. Lakshmanan. “On approximating optimum repairs for functional dependency violations”. ICDT 2009.
- [KNP+ SIGMOD16] Arun Kumar, Jeffrey Naughton, Jignesh M. Patel, and Xiaojin Zhu. “To Join or Not to Join? Thinking Twice about Joins before Feature Selection”. SIGMOD 2016.
- [MAD ArXiv16] H. Miao, A. Chavan, A. Deshpande. “ProvDB: A System for Lifecycle Management of Collaborative Analysis Workflows”. arXiv:1610.04963.

References

- [MCC+ ArXiv13] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space"
- [MGL+ PVLDB10] S. Melnik et al. "Dremel: Interactive Analysis of Web-Scale Datasets". PVLDB, 2010.
- [MLD+ ICDE17] H. Miao, A. Li, L. S. Davis, A. Deshpande. "Towards Unified Data and Lifecycle Management for Deep Learning". ICDE 2017.
- [O_09] Fredrick Olsson. "A literature survey of active machine learning in the context of natural language processing". SICS Technical Report, 2009.
- [OR PVLDB11] C. Olston and B. Reed, "Inspector Gadget: A Framework for Custom Monitoring and Debugging of Distributed Dataflows", PVLDB 2011.
- [P 00] K. Pearson, "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling", Philosophical Magazine Series 5, 1900.
- [PTS+ CIDR17] Shoumik Palkar, James J. Thomas, Anil Shanbhag, Deepak Narayanan, Holger Pirk, Malte Schwarzkopf , Saman Amarasinghe, Matei Zaharia. "Weld: A Common Runtime for High Performance Data Analytics". CIDR 2017.
- [RR KER13] A. Romei, S. Ruggieri. "A Multidisciplinary Survey on Discrimination Analysis". The Knowledge Engineering Review, 5(29).
- [RSS+ TCDE14] Christopher Ré, Amir Abbas Sadeghian, Zifei Shan, Jaeho Shin, Feiran Wang, Sen Wu, Ce Zhang. "Feature Engineering for Knowledge Base Construction". TCDE 2014

References

- [S 12] Burr Settles. *Active Learning: Synthesis Lectures on Artificial Intelligence and Machine Learning*. 2012.
- [SKL+ PVLDB16] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, A. Parameswaran. “Effortless Data Exploration with zenvisage: An Expressive and Interactive Visual Analytics System”. PVLDB, 2016.
- [SS VLDB01] G. Sathe and S. Sarawagi, “Intelligent Rollups in Multidimensional OLAP Data”. VLDB, 2001.
- [SPI+ KDD08] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. “Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers”. SIGKDD 2008
- [T62] J. Tukey. “The Future of Data Analysis”. *The Annals of Mathematical Statistics*, 1962
- [VCSM ICDE14] M. Volkovs, F. Chiang, J. Szlichta, and R. Miller. “Continuous Data Cleaning”, ICDE, 2014.
- [VRM+ PVLDB15] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, N. Polyzotis. “SeeDB: Efficient Data-Driven Visualization Recommendations to Support Visual Analytics”, PVLDB, 2015.
- [WDM SIGMOD15] Xiaolan Wang, Xin Luna Dong, Alexandra Meliou, “Data X-Ray: A Diagnostic Tool for Data Errors”. SIGMOD 2015.
- [WFM+VLDBJ15] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. “Towards certain fixes with editing rules and master data”. VLDB Journal
- [ZSZ+ SIGMOD17] Z. Zhao, L. De Stefani, E. Zgraggen, C. Binnig, E. Upfal, T. Kraska. “Controlling False Discoveries During Interactive Data Exploration”. SIGMOD, 2017.