SE for ML

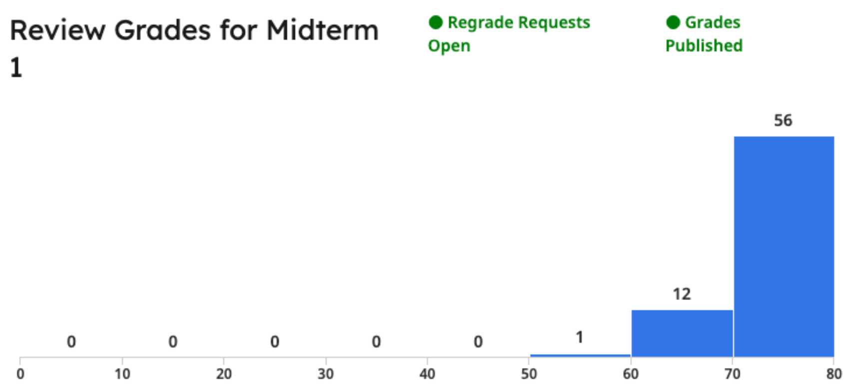
17-313 Spring 2023

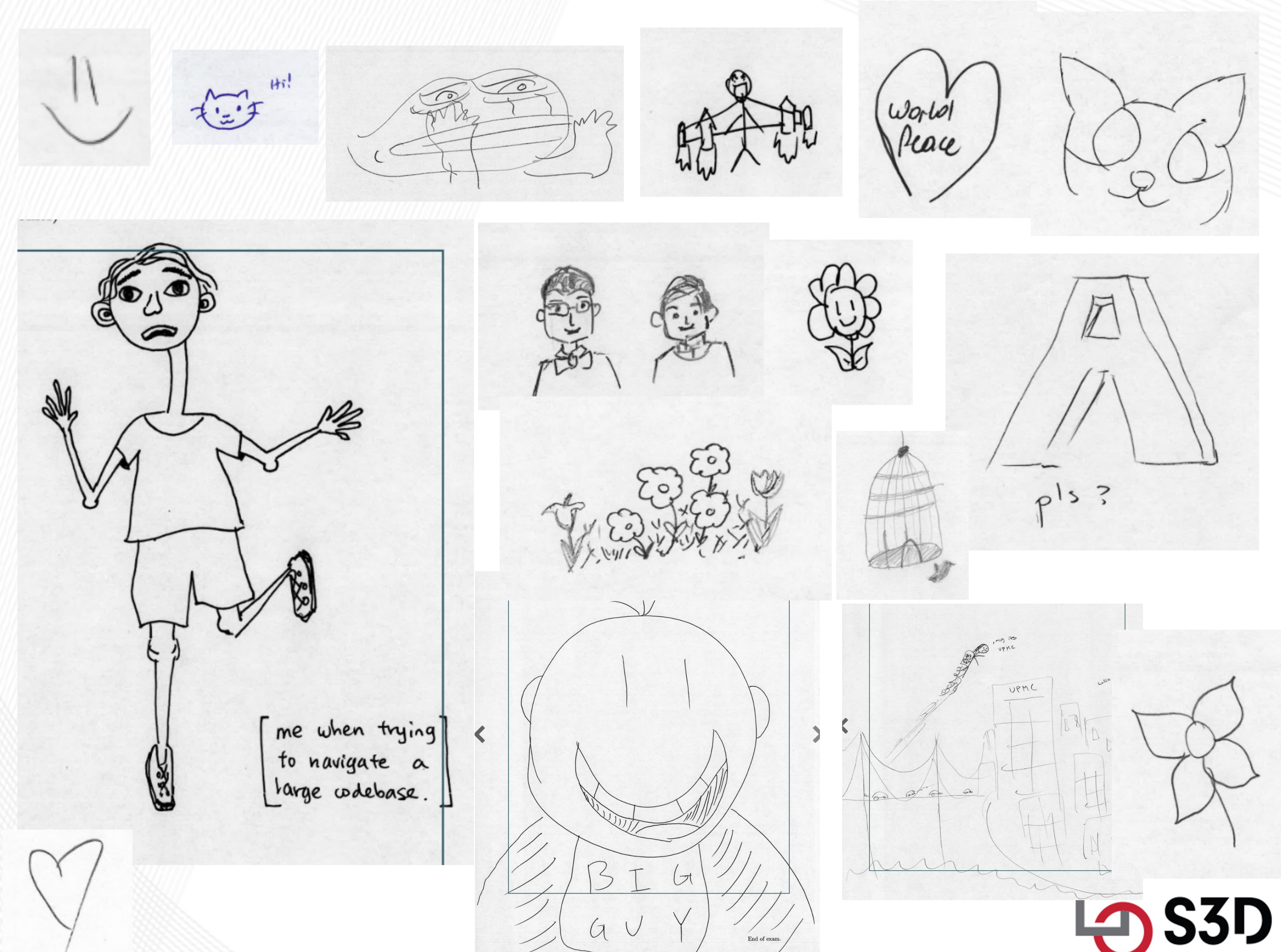
Carnegie Mellon University

S3D Software and Societal Systems Department

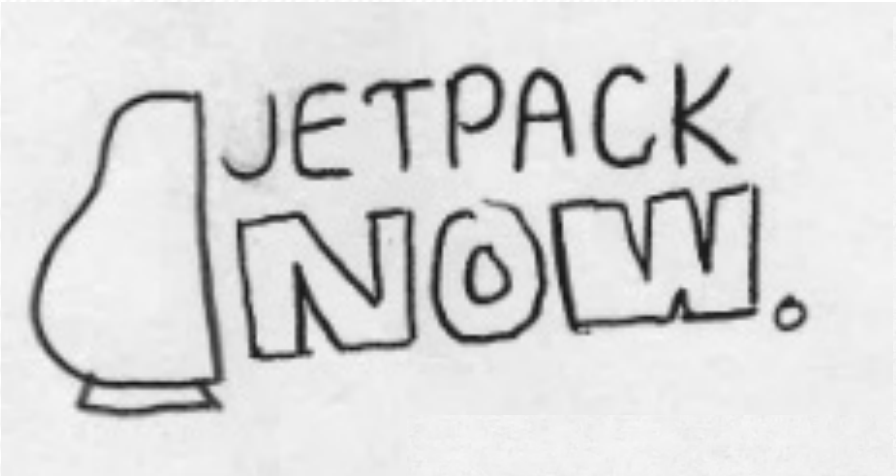
Administrivia

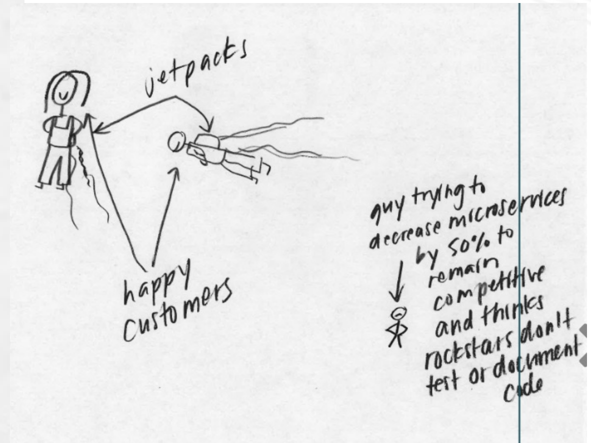
- **Project 3 Released**
  We have pushed back the checkpoint deadline for Project 3 from this Thursday to this Friday, given that fly.io has experienced two outages (which is atypical for fly.io) in the week that we release the project.
- We do not typically adjust deadlines and/or give extensions for projects in this class, as planning and delivering on deadlines is part of the learning objectives of this class, but we understand the outages are not within students' control.
- **Midterm is graded**

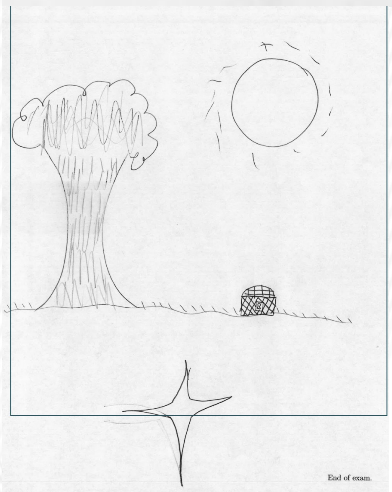


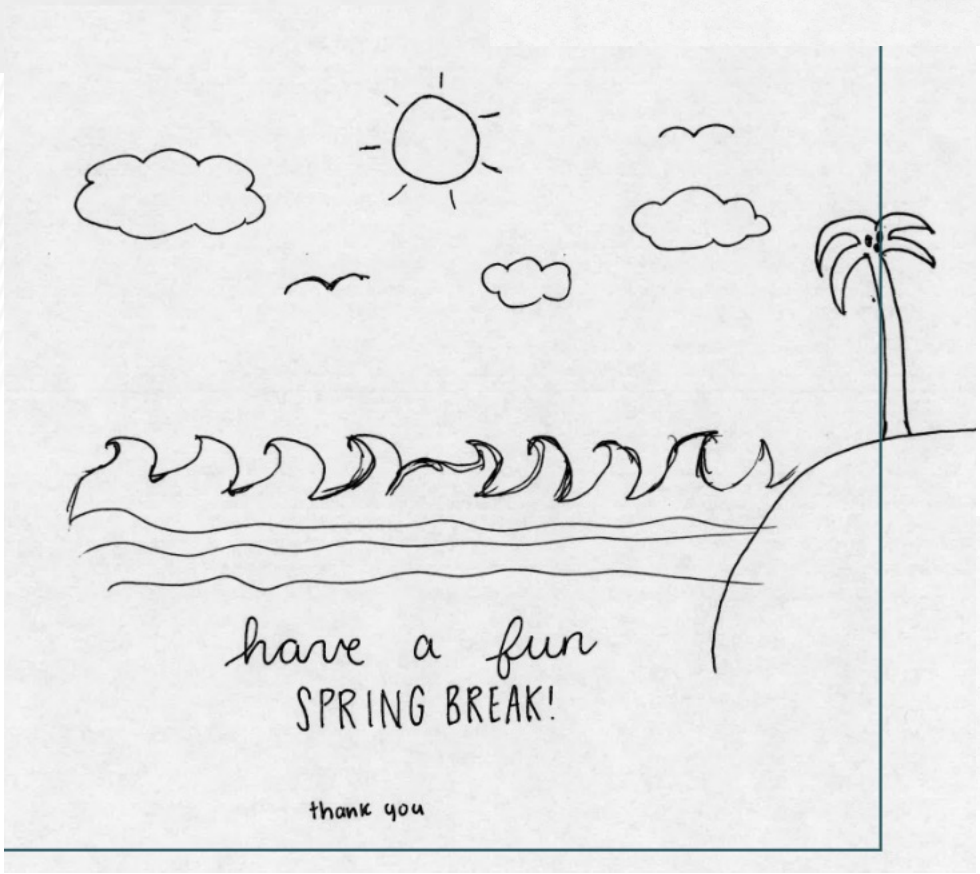


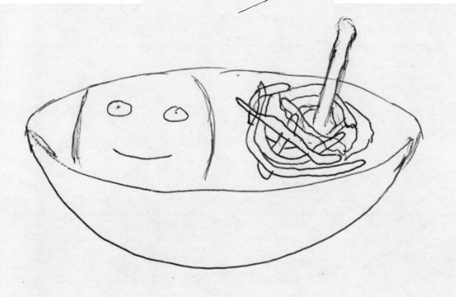
Hi!
World Peace
me when trying to navigate a large codebase.
pls ?
BIG GUY
End of exam.
UPMC


JETPACK
NOW.


jetpacks
happy customers
guy trying to decrease microservices by 50% to remain competitive and thinks rockstars don't test or document code


End of exam.


have a fun
SPRING BREAK!
thank you



Retrospectives

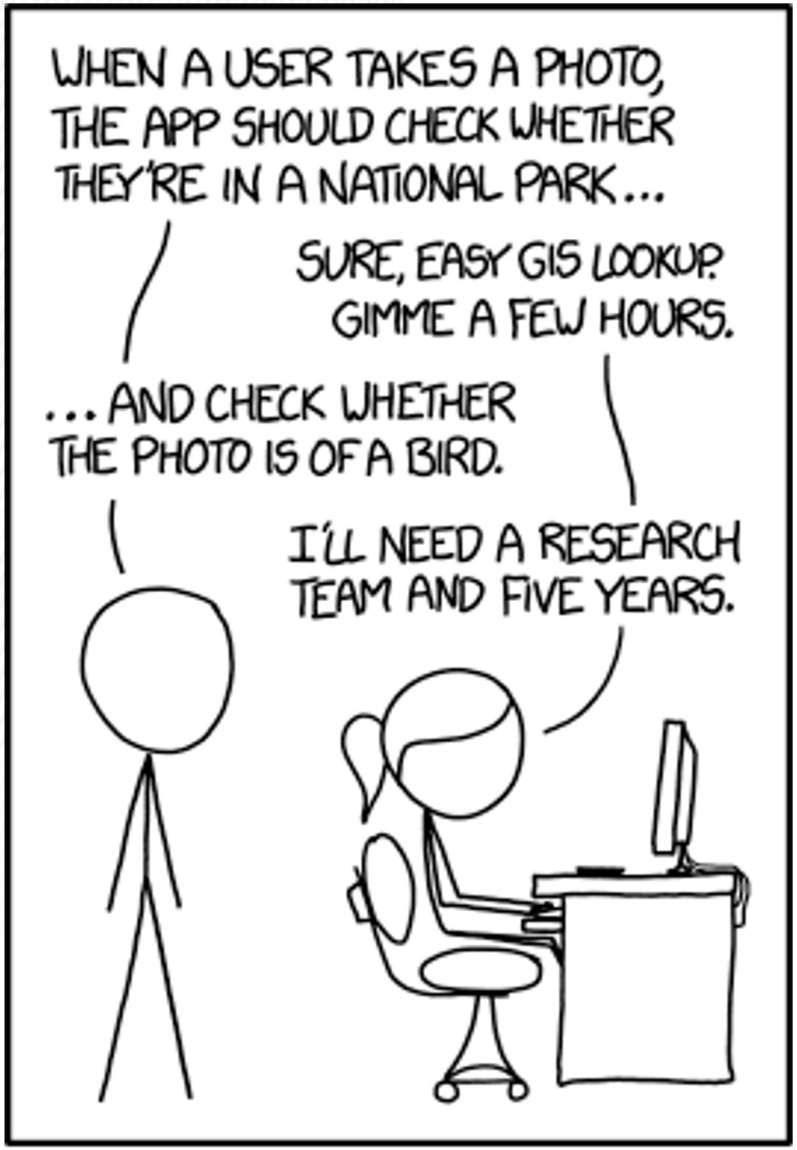
- “the purpose of the Sprint Retrospective is to plan ways to increase quality and effectiveness.” –Scrum.org

- We often use three questions:
- What should we:
  - Start doing?
  - Stop doing?
  - Keep doing?

Learning goals

- Identify differences between traditional software development and development of ML systems.
- Understand the stages that comprise the typical ML development pipeline.
- Identify challenges that must be faced within each stage of the typical ML development pipeline.

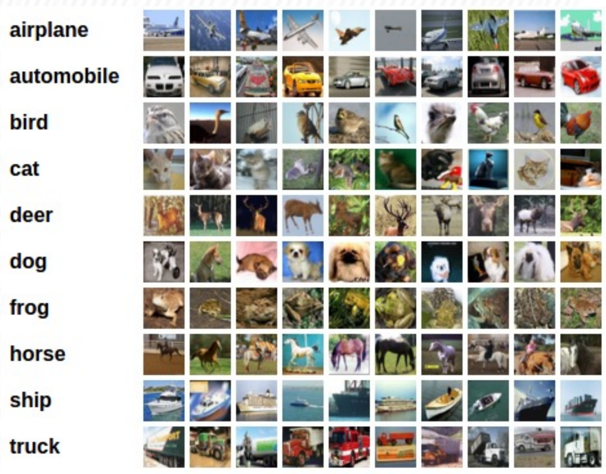
Quick poll:
Have you taken a machine learning course before?



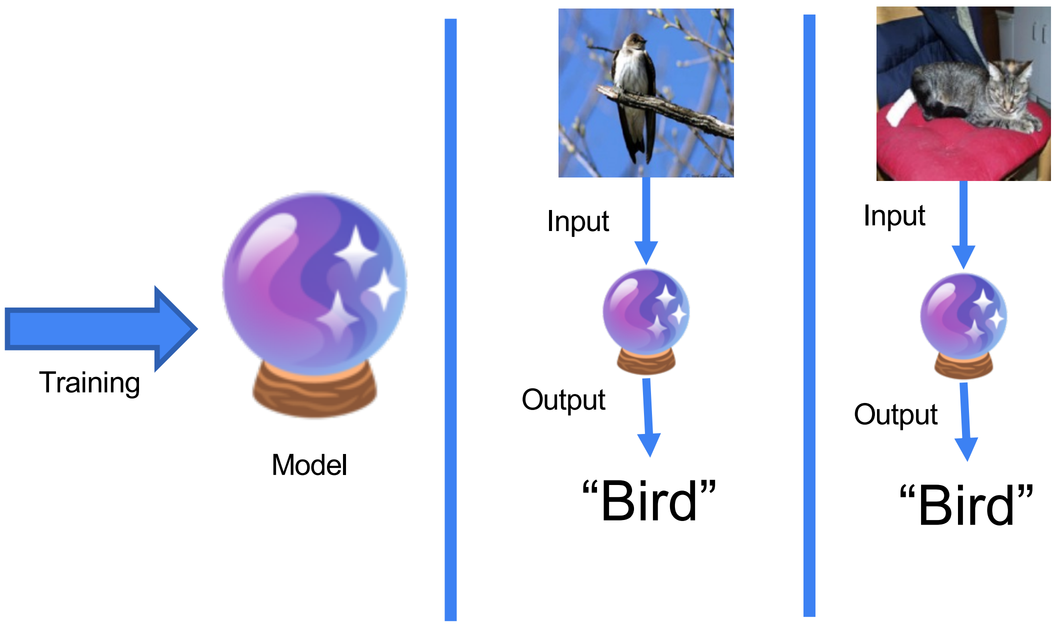
Source: https://xkcd.com/1425/

Machine Learning in One Slide

(Supervised)

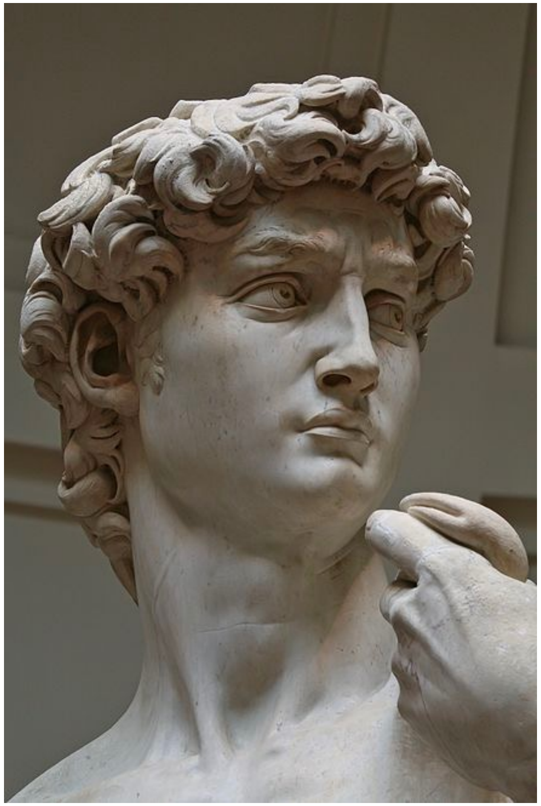


Lots of labelled data
(Inputs, outputs)



Traditional Software Development

“It is easy. You just chip away the stone that doesn’t look like David.” –(probably not) Michelangelo



S3D

ML Development

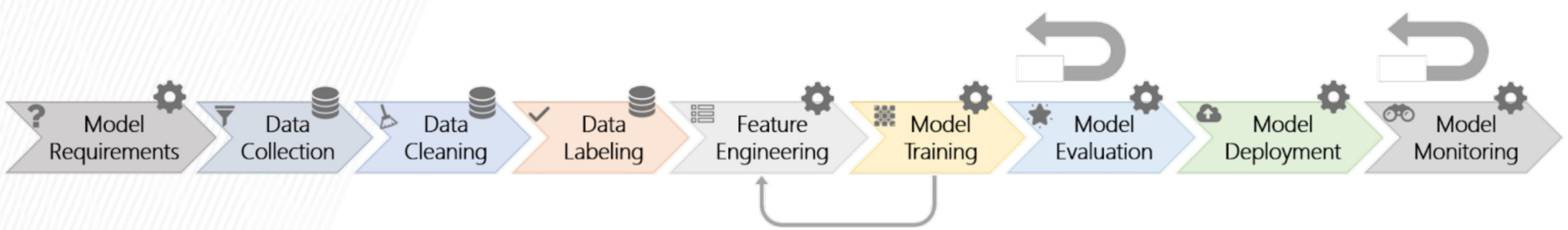
- Observation
- Hypothesis
- Predict
- Test
- Reject or Refine Hypothesis

S3D

Black-box View of Machine Learning

Image: https://xkcd.com/1838/

Microsoft's view of Software Engineering for ML



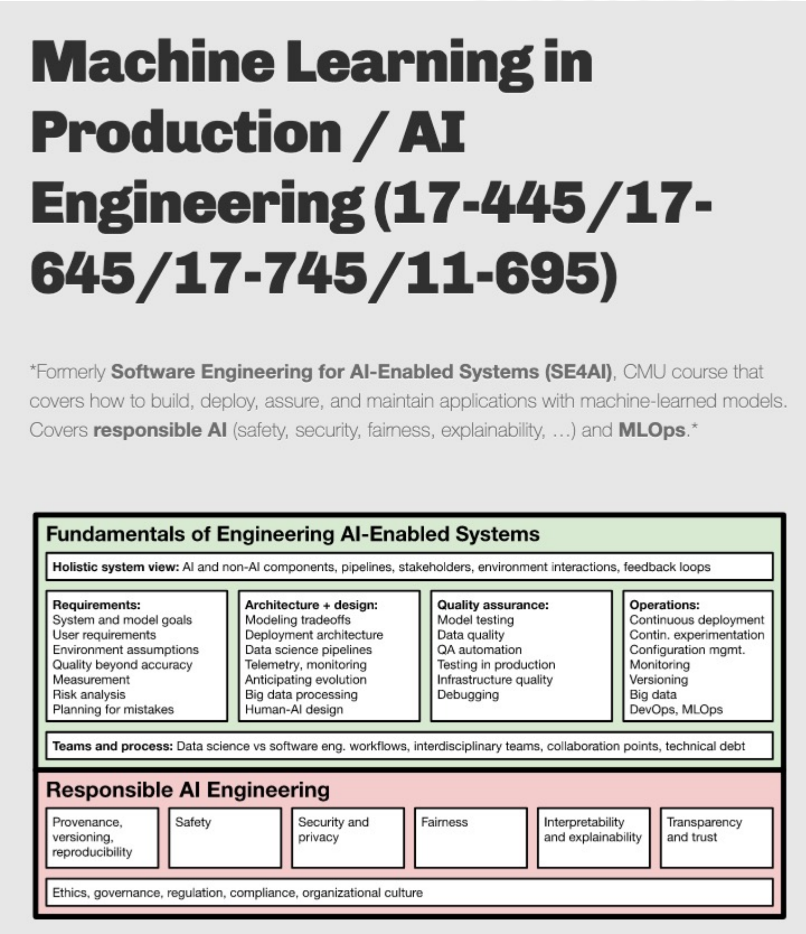
Source: "Software Engineering for Machine Learning: A Case Study" by Amershi et al. ICSE

Three Fundamental Differences:

- Data discovery and management
- Customization and Reuse
- No modular development of model itself

Case Study

- Case study developed by
- Christian Kästner
- https://ckaestne.github.io/seai/




現金のみ
Japanese
English
cash only


PhoDUCK
TIBETAN
INDIAN
NEPALI
FOOD
POTALA
POTALA RESTAURANT
명동교자
아하
노래방
빤닭빤닭
鸡肋
タッカルビ
DAKGALBI
명동본점 2F
MINI BEER PUB
홍콩반점
김밥
정성본
banila co.
THEFACESHOP
SALE
age FOTOSTOCK
Paul Brown
Z8Z-2650464

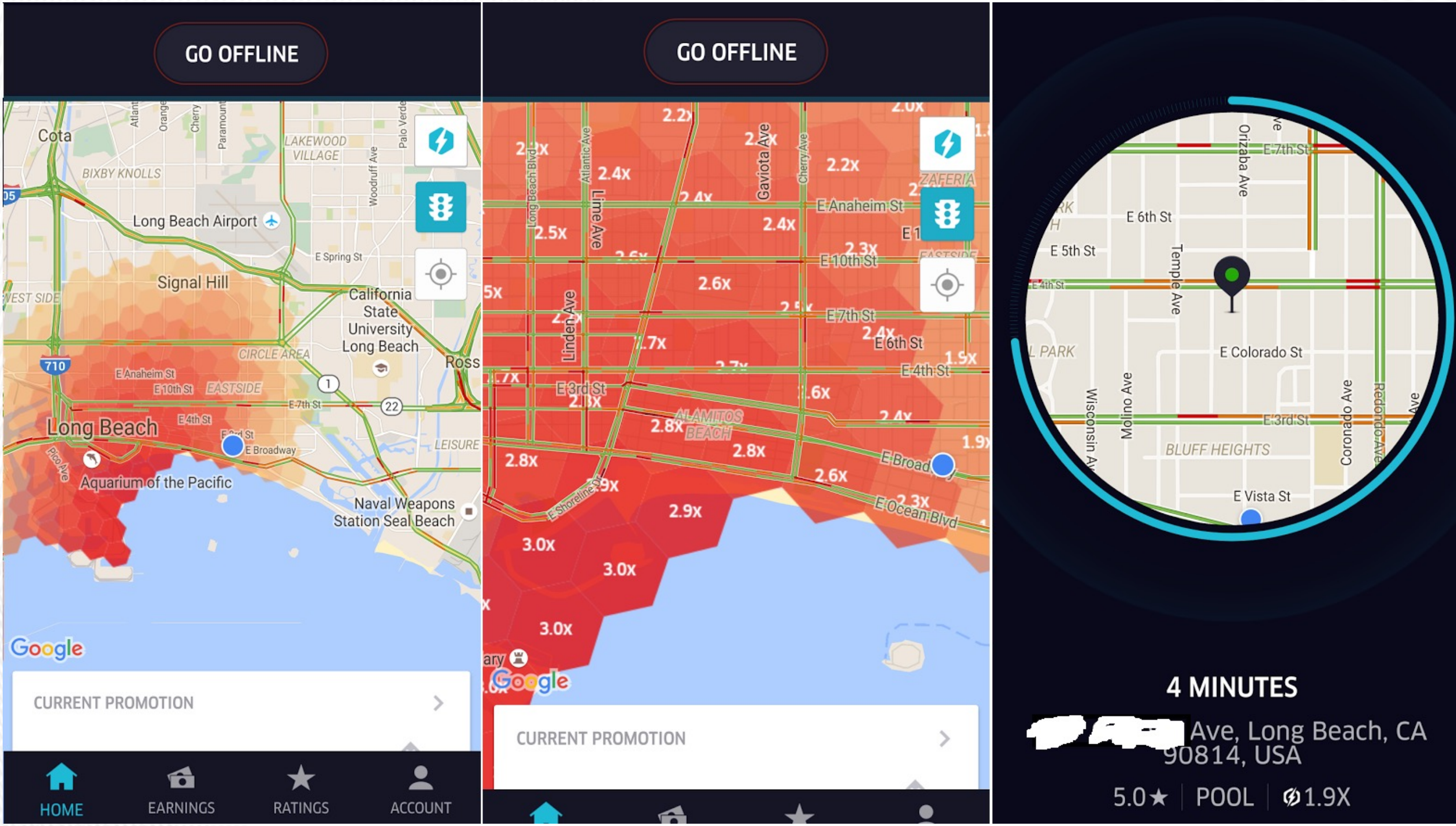

The Next Generation
of Spectacles

Qualities of Interest?

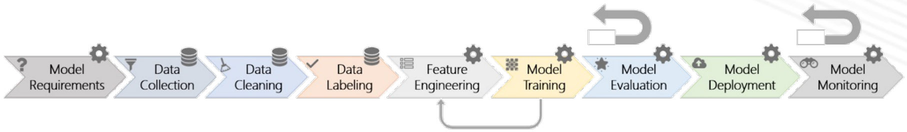
A

B

C

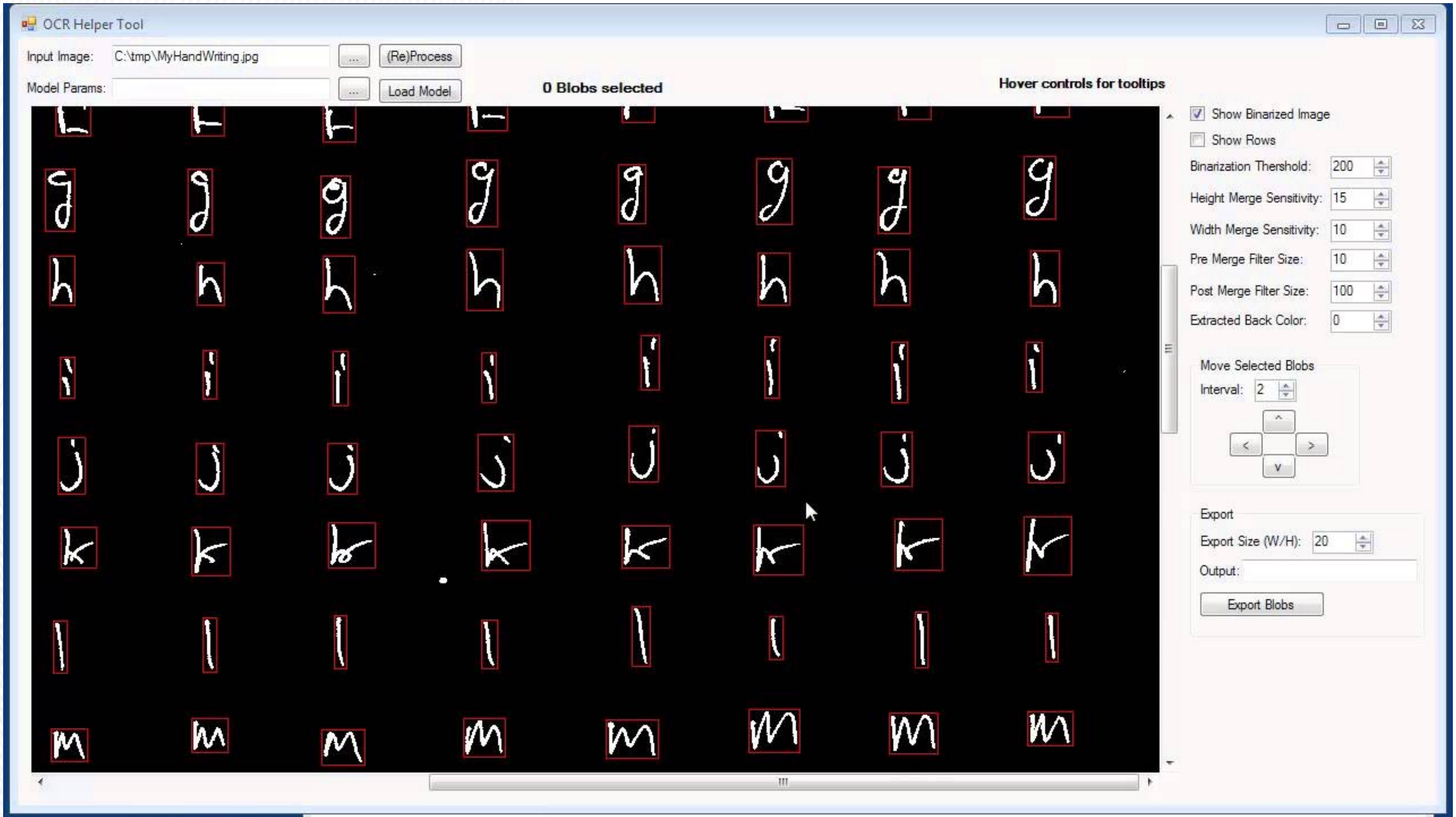

GO OFFLINE
Long Beach Airport
Signal Hill
California State University Long Beach
Long Beach
Aquarium of the Pacific
Naval Weapons Station Seal Beach
CURRENT PROMOTION
HOME
EARNINGS
RATINGS
ACCOUNT
GO OFFLINE
3.0x
2.9x
2.8x
2.6x
CURRENT PROMOTION
4 MINUTES
[redacted] Ave, Long Beach, CA 90814, USA
5.0★
POOL
1.9X

Typical ML Pipeline

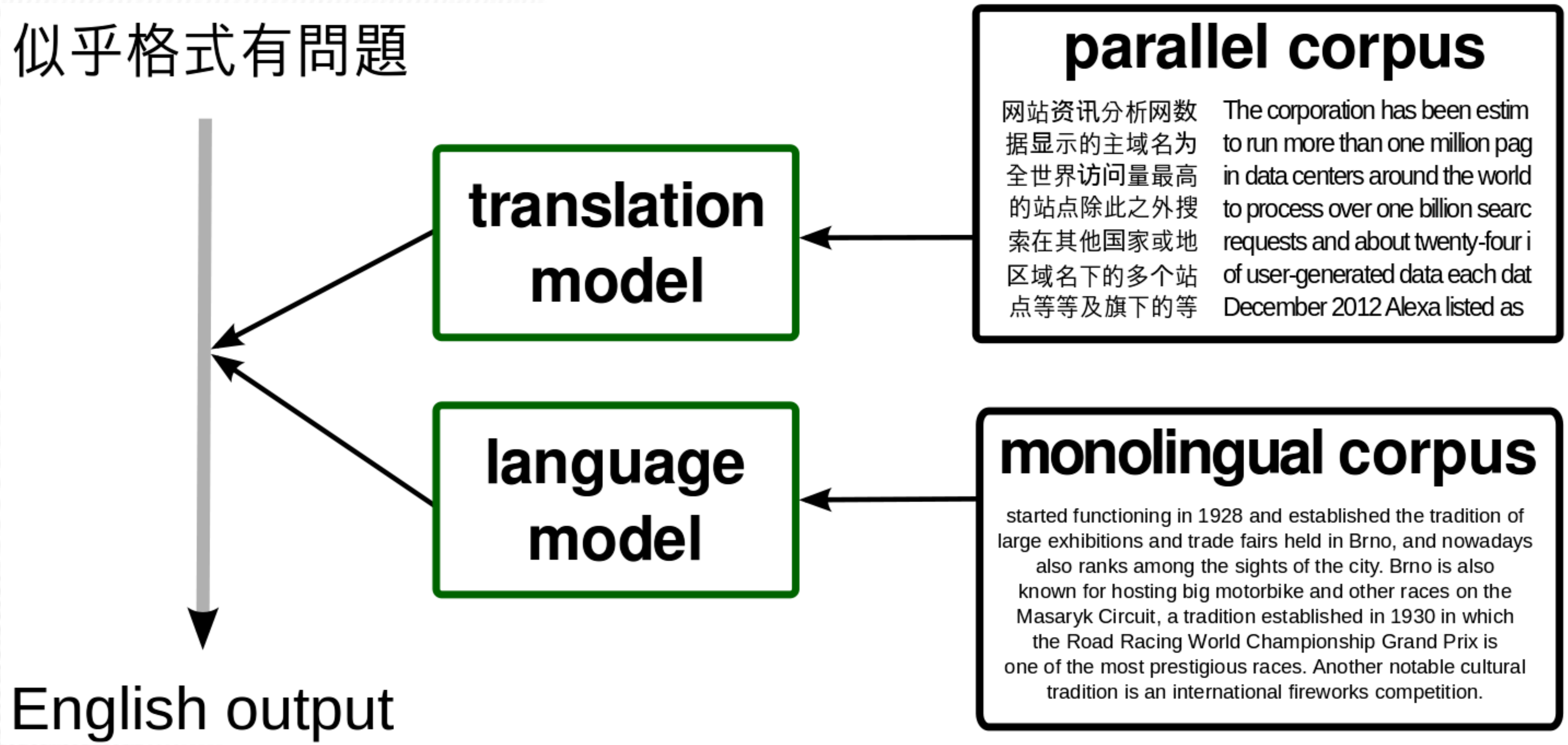


- Static
  - Get labeled data (data collection, cleaning and, labeling)
  - Identify and extract features (feature engineering)
  - Split data into training and evaluation set
  - Learn model from training data (model training)
  - Evaluate model on evaluation data (model evaluation)
  - Repeat, revising features
- with production data
  - Evaluate model on production data; monitor (model monitoring)
  - Select production data for retraining (model training + evaluation)
  - Update model regularly (model deployment)

Example Data



Learning Data



Example Data

UserId	PickupLocation	TargetLocation	OrderTime	PickupTime
5	….	…	18:23	18:31
…				

Feature Engineering

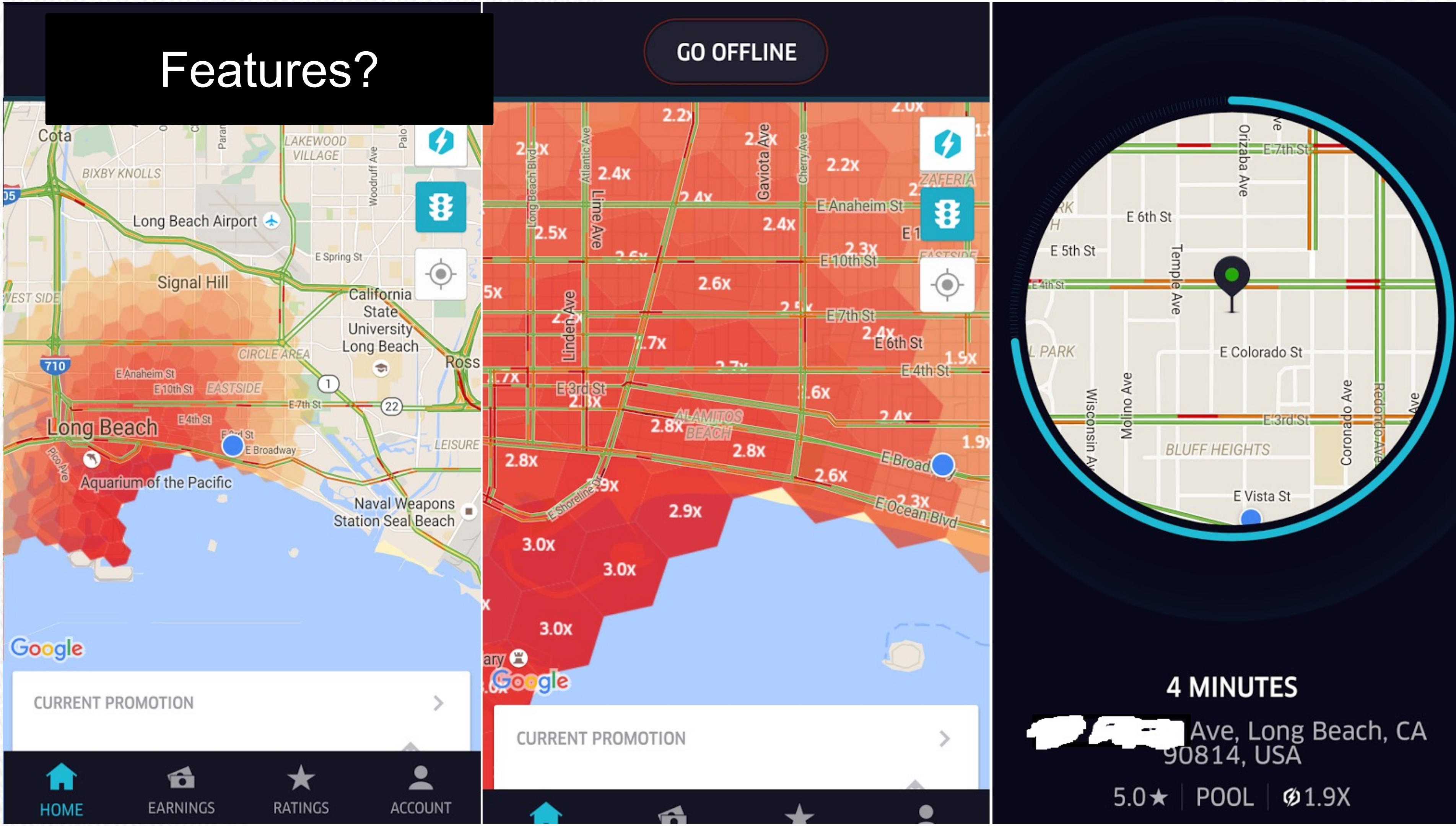
- Identify parameters of interest that a model may learn on
- Convert data into a useful form
- Normalize data
- Include context
- Remove misleading things

Features?

Feature Extraction

- In OCR/translation:
  - Bounding boxes for text of interest
  - Character boundaries
  - Line segments for each character
  - GPS location of phone (to determine likely source language)

Features?



S3D

Feature Extraction

- In surge prediction:
    - Location and time of past surges
    - Events
    - Number of people traveling to an area
    - Typical demand curves in an area
    - Demand in other areas
    - Weather

Data Cleaning

- Removing outliers
- Normalizing data
- Missing values
- …

Learning

- Build a predictor that best describes an outcome for the observed features

Evaluation

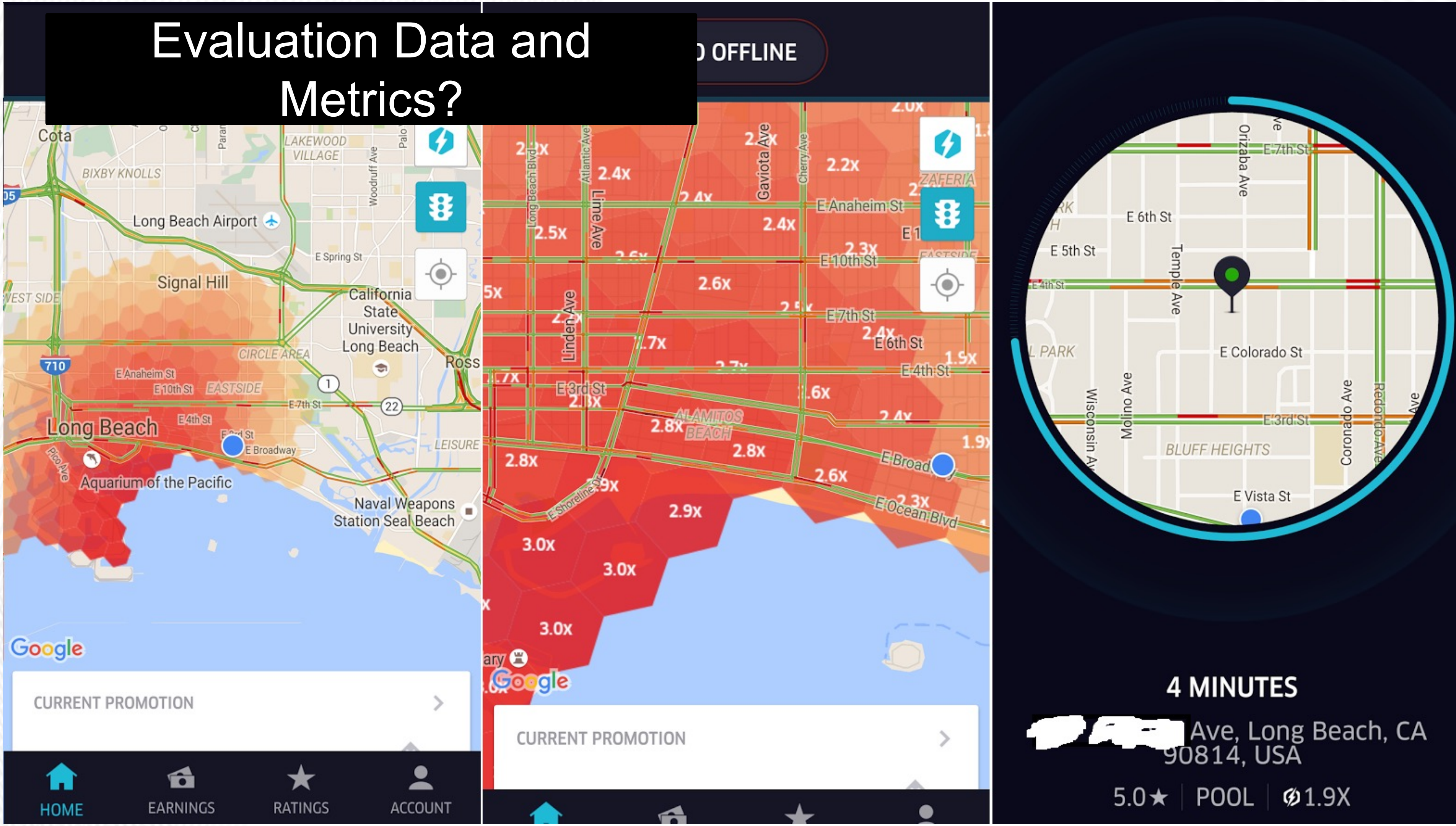
- Prediction accuracy on learned data vs
- Prediction accuracy on unseen data
  - Separate learning set, not used for training

- For binary predictors: false positives vs. false negatives, precision vs. recall
- For numeric predictors: average (relative) distance between real and predicted value
- For ranking predictors: top-K, etc.

S3D

Evaluation Data and Metrics?

Evaluation Data and Metrics?



Learning and Evaluating in Production

- Beyond static data sets, **build telemetry**
- Design challenge: identify mistakes in practice

- Use sample of live data for evaluation
- Retrain models with sampled live data regularly
- Monitor performance and intervene

ML Model Tradeoffs

- Accuracy
- Capabilities (e.g. classification, recommendation, clustering…)
- Amount of training data needed
- Inference latency
- Learning latency; incremental learning?
- Model size
- Explainable? Robust?
- …

Where should the model live?

Glasses

Phone

Cloud

OCR Component

Translation Component

Where should the model live?

Vehicle

Phone

Cloud

Surge Prediction

Considerations

- How much data is needed as input for the model?
- How much output data is produced by the model?
- How fast/energy consuming is model execution?
- What latency is needed for the application?
- How big is the model? How often does it need to be updated?
- Cost of operating the model? (distribution + execution)
- Opportunities for telemetry?
- What happens if users are offline?

Typical Designs

- Static intelligence in the product
  - difficult to update
  - good execution latency
  - cheap operation
  - offline operation
  - no telemetry to evaluate and improve
- Client-side intelligence
  - updates costly/slow, out of sync problems
  - complexity in clients
  - offline operation, low execution latency

Typical Designs

- Server-centric intelligence
  - latency in model execution (remote calls)
  - easy to update and experiment
  - operation cost
  - no offline operation
- Back-end cached intelligence
  - precomputed common results
  - fast execution, partial offline
  - saves bandwidth, complicated updates
- Hybrid models

Other Considerations

- Coupling of ML pipeline parts
- Coupling with other parts of the system
- Ability for different developers and analysists to collaborate
- Support online experiments
- Ability to monitor

Reactive Systems

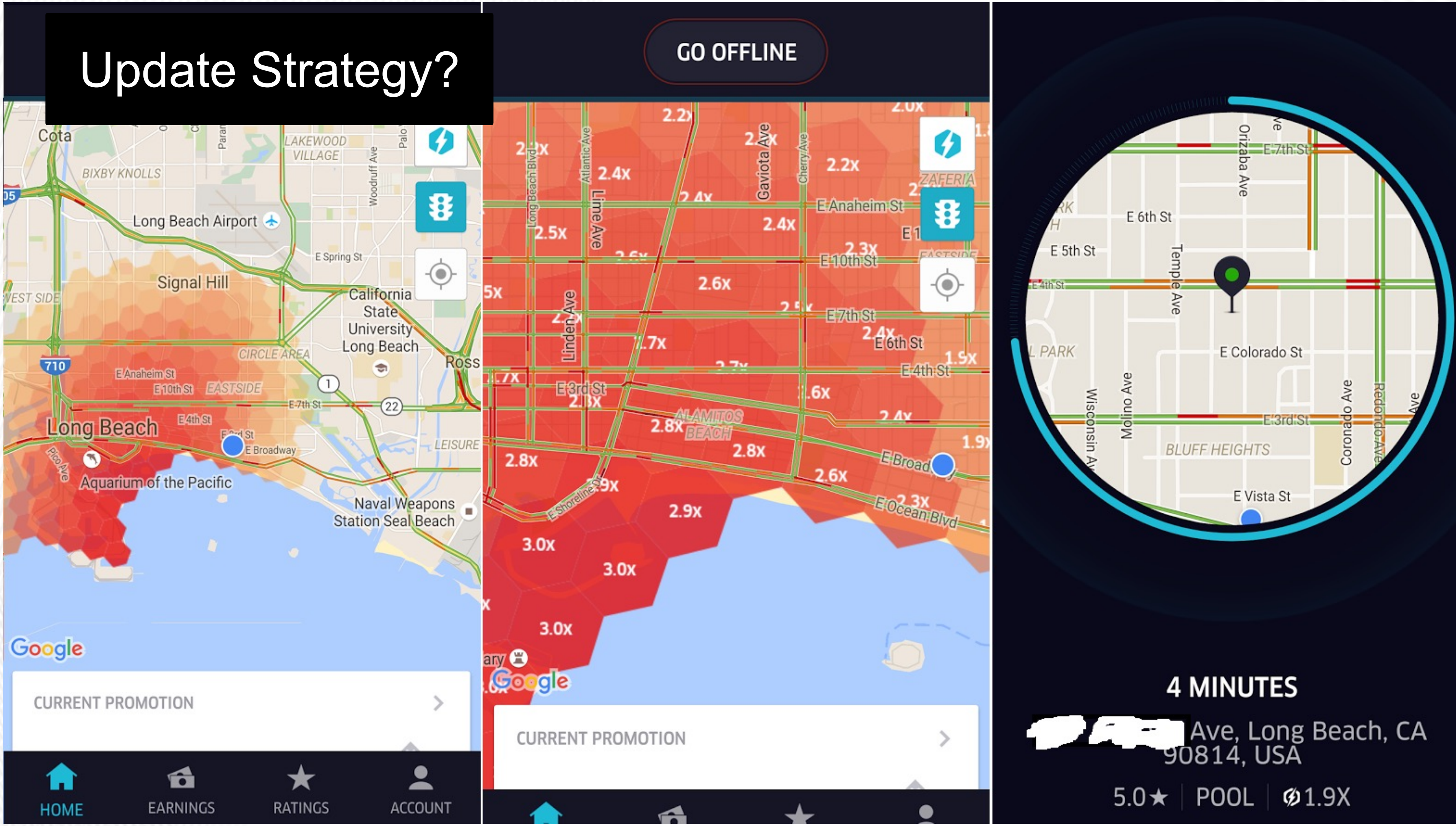
- Responsive
  - consistent, high performance
- Resilient
  - maintain responsive in the face of failure, recovery, rollback
- Elastic
  - scale with varying loads

Updating Models

- Models are rarely static outside the lab
- Data drift, feedback loops, new features, new requirements
- When and how to update models?
- How to version? How to avoid mistakes?

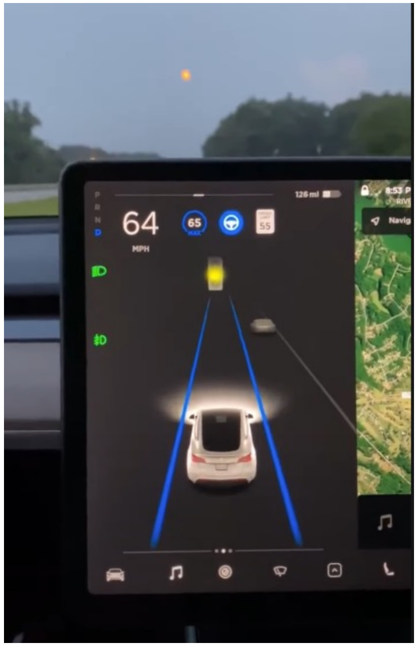
Update Strategy?

Update Strategy?



Mistakes will happen

- No specification
- ML components detect patterns from data (real and spurious)
- Predictions are often accurate, but mistakes always possible
- Mistakes are not predicable or explainable or similar to human mistakes
- Plan for mistakes
- Telemetry to learn about mistakes?



How Models can Break

- System outage
- Model outage
  - model tested? deployment and updates reliable? file corrupt?
- Model errors
- Model degradation
  - data drift, feedback loops

Hazard Analysis

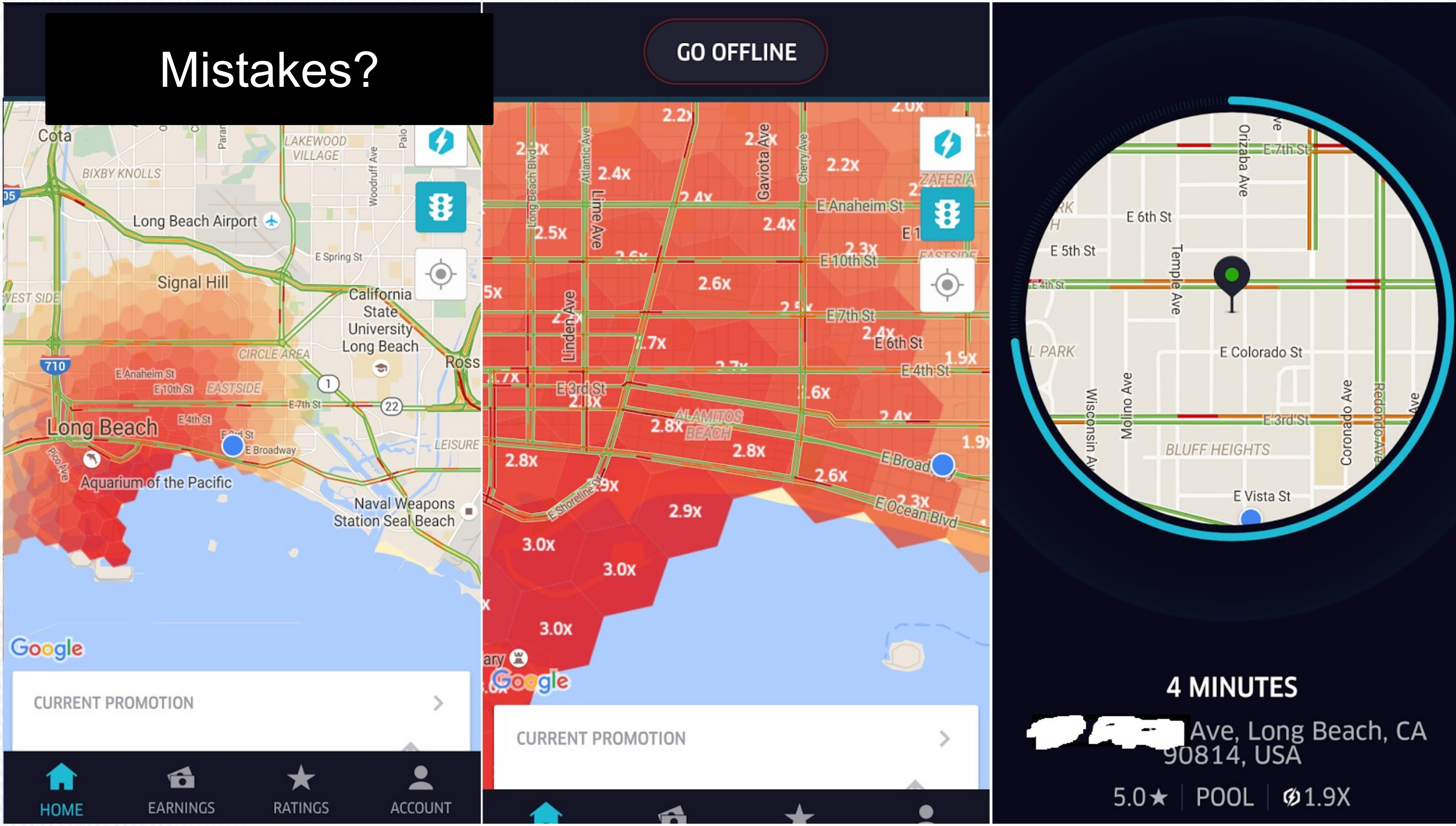
- Worst thing that can happen?
- Backup strategy? Undoable? Nontechnical compensation?

Mitigating Mistakes

- Investigating in ML
  - e.g., more training data, better data, better features, better engineers
- Less forceful experience
  - e.g., prompt rather than automate decisions, turn off
- Adjust learning parameters
  - e.g., more frequent updates, manual adjustments
- Guardrails
  - e.g., heuristics and constraints on outputs
- Override errors
  - e.g., hardcode specific results

Mistakes?

Mistakes?



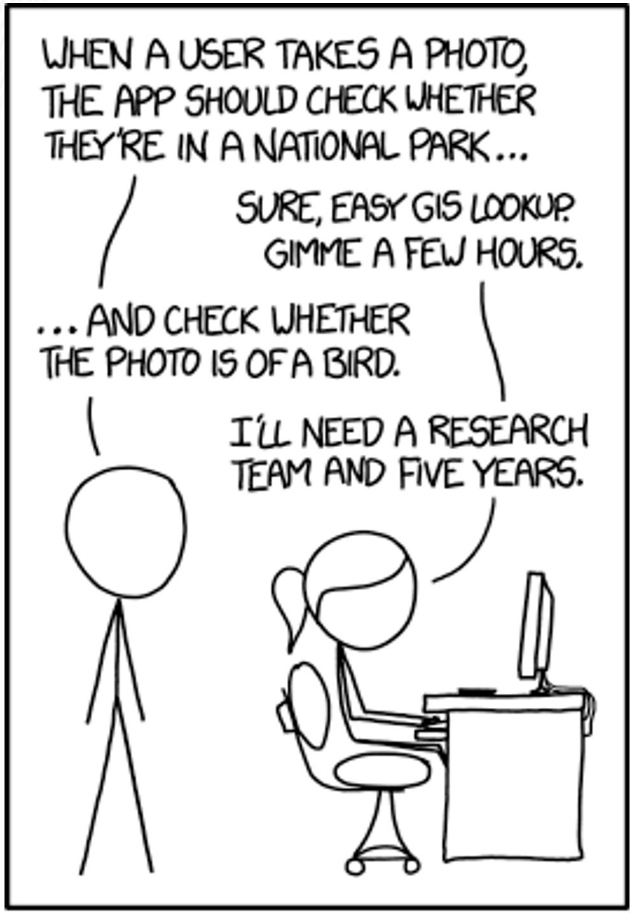
Cota
BIXBY KNOLLS
LAKEWOOD VILLAGE
Long Beach Airport
E Spring St
Signal Hill
California State University Long Beach
CIRCLE AREA
EASTSIDE
E Anaheim St
E 10th St
E 7th St
E 4th St
Long Beach
E Broadway
LEISURE
Aquarium of the Pacific
Naval Weapons Station Seal Beach
Google
CURRENT PROMOTION
HOME
EARNINGS
RATINGS
ACCOUNT
GO OFFLINE
2.2x
2.4x
2.5x
2.6x
2.7x
2.8x
2.9x
3.0x
2.3x
1.9x
E Anaheim St
E 10th St
E 7th St
E 6th St
E 4th St
E 3rd St
ALAMITOS BEACH
E Ocean Blvd
Linden Ave
Lime Ave
Gaviota Ave
Google
CURRENT PROMOTION
E 7th St
E 6th St
E 5th St
Orizaba Ave
Temple Ave
E Colorado St
E 3rd St
BLUFF HEIGHTS
E Vista St
Wisconsin Ave
Molino Ave
Coronado Ave
Redondo Ave
4 MINUTES
[redacted] Ave, Long Beach, CA 90814, USA
5.0★ POOL 1.9X

Telemetry

- Purpose:
  - monitor operation
  - monitor success (accuracy)
  - improve models over time (e.g., detect new features)
- Challenges:
  - too much data – sample, summarization, adjustable
  - hard to measure – intended outcome not observable? proxies?
  - rare events – important but hard to capture
  - cost – significant investment must show benefit
  - privacy – abstracting data

Requirements and estimation

- Talking to stakeholders



Source: https://xkcd.com/1425/

Summary

- Machine learning in production systems is challenging
- Many tradeoffs in selecting ML components and in integrating them in larger system
- Plan for updates
- Manage mistakes, plan for telemetry