Detection of unusual travel patterns to prevent user account compromise

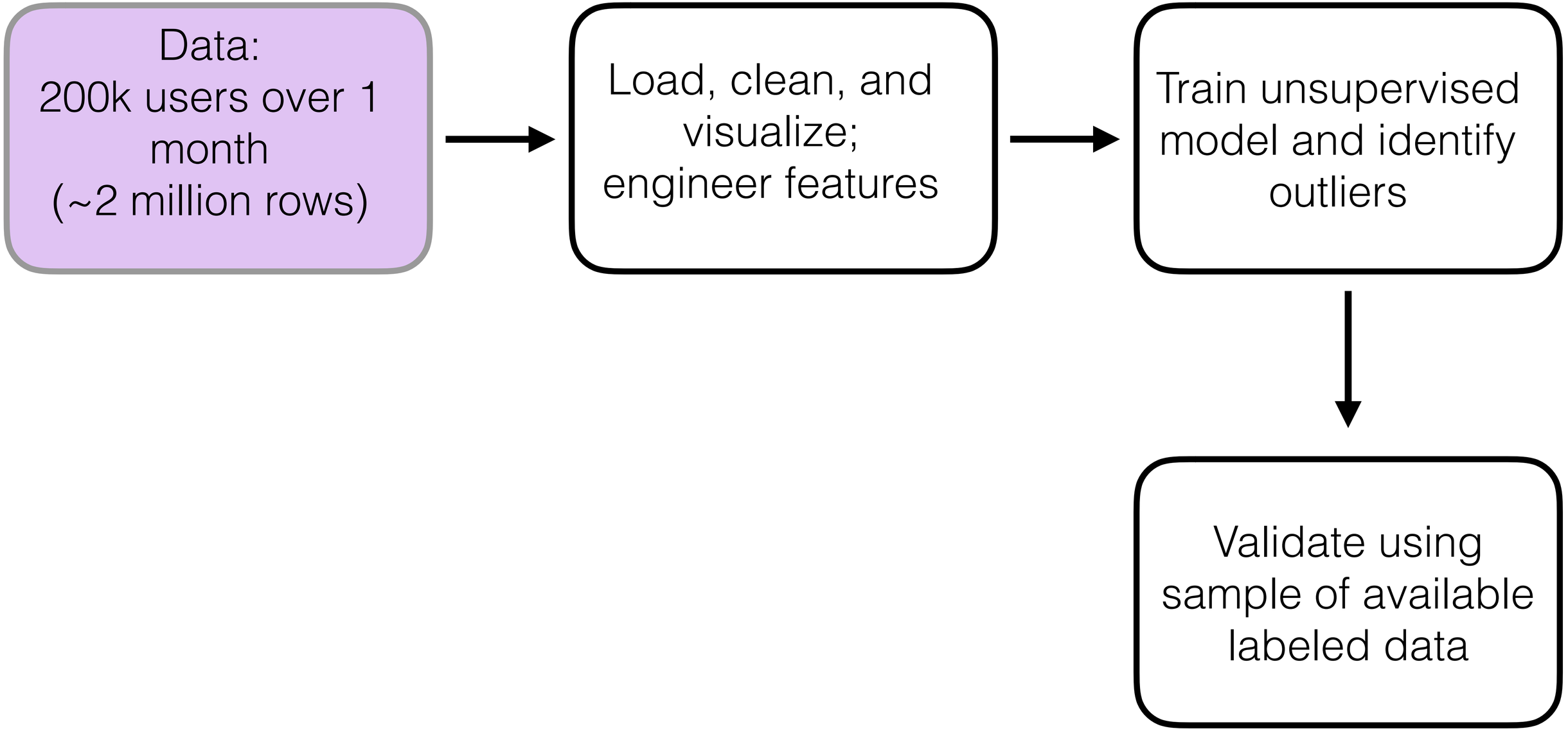
James Verbus

Collaboration with Castle (castle.io)

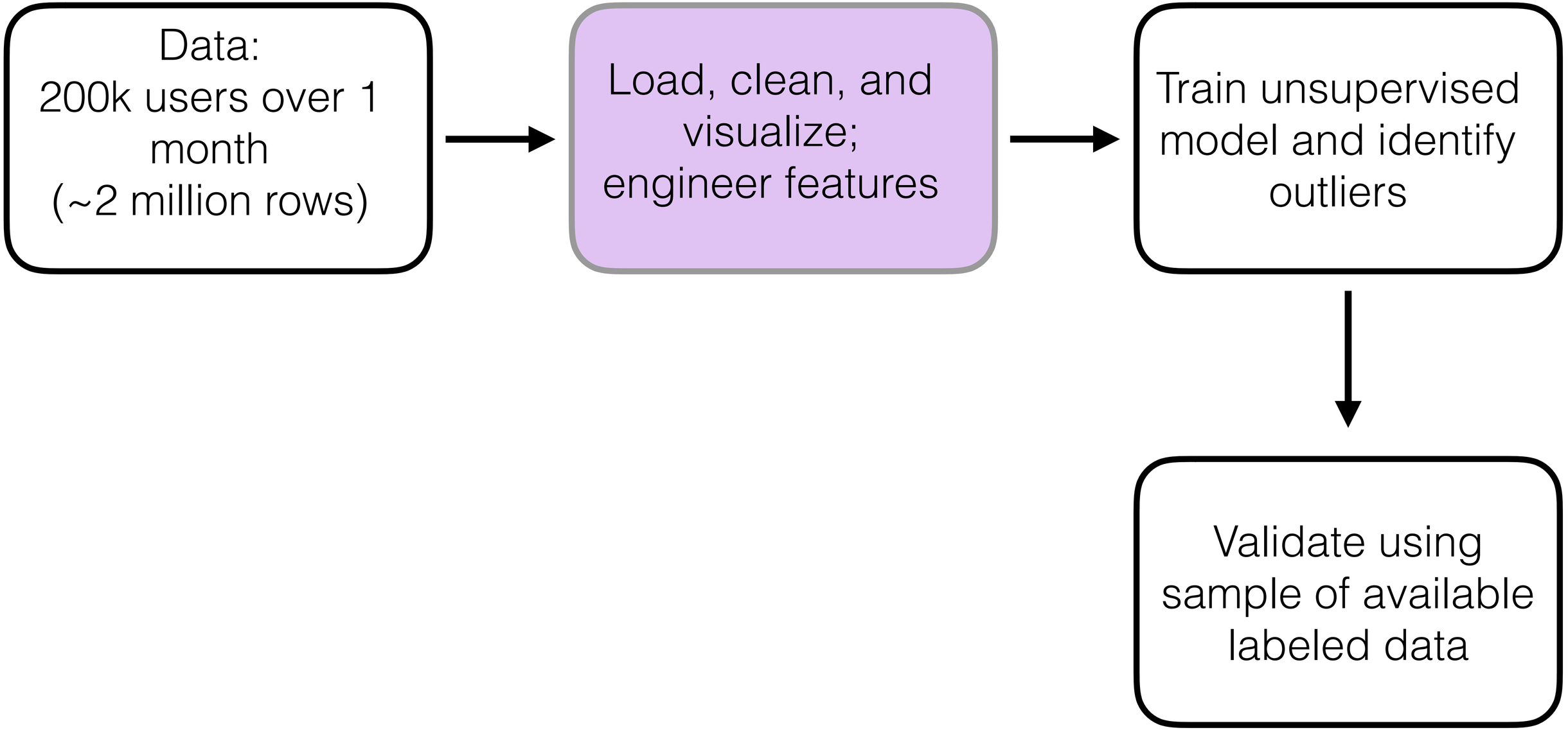
- Castle provides automated detection of compromised user accounts & hijack attempts for online businesses

- **Deliverable:** Develop a model to predict the likelihood that a new login belongs to a specific user

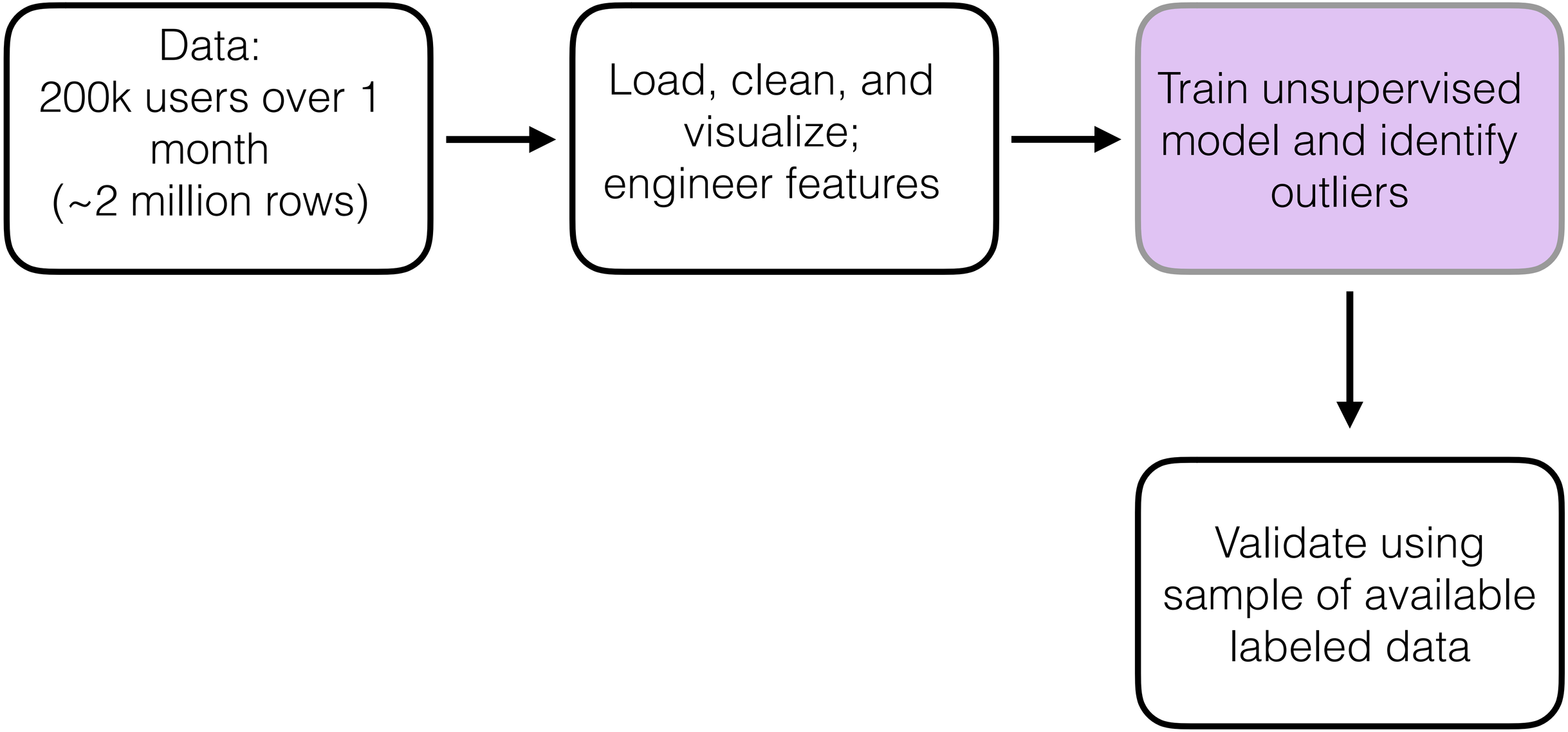
Data and Analysis Pipeline



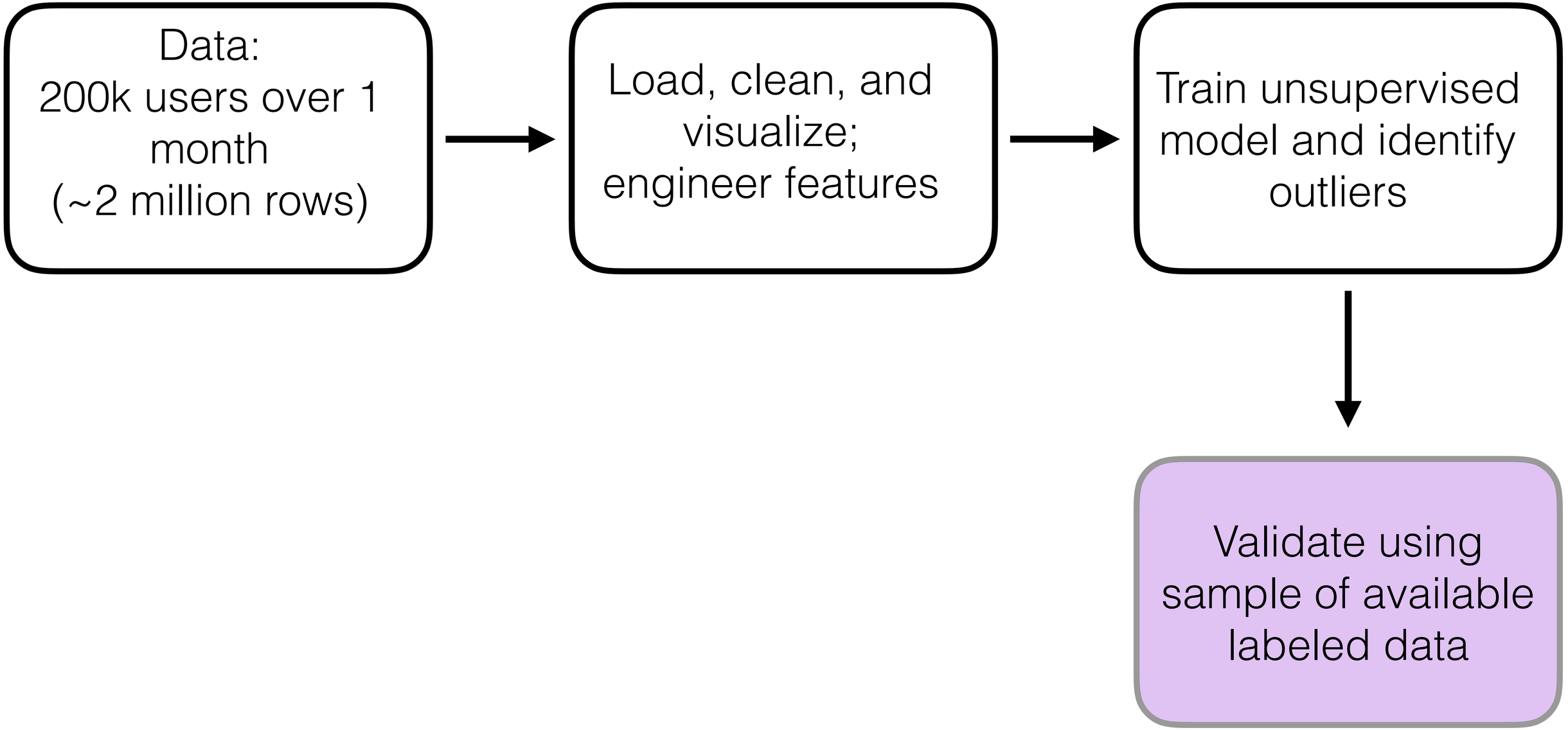
Data and Analysis Pipeline



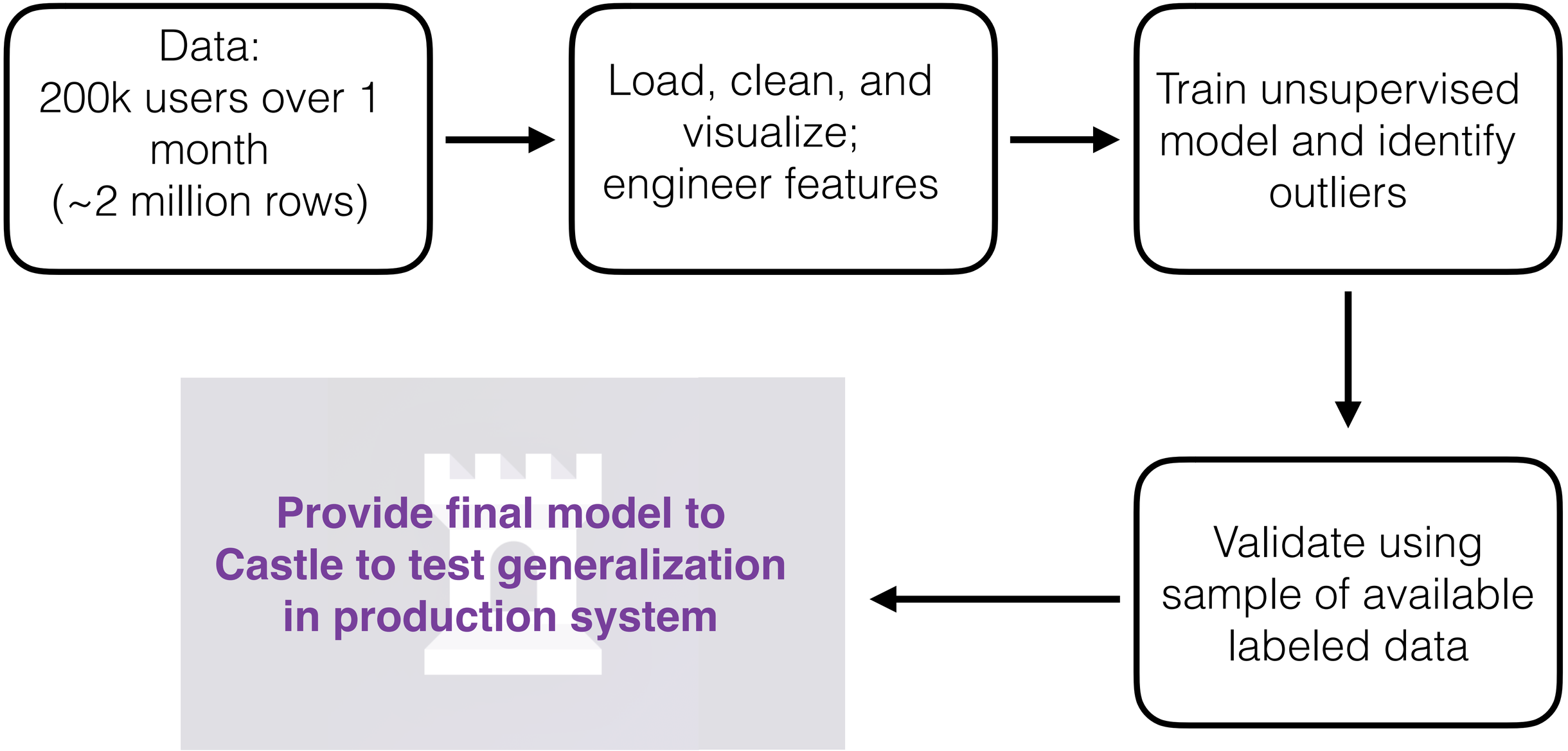
Data and Analysis Pipeline

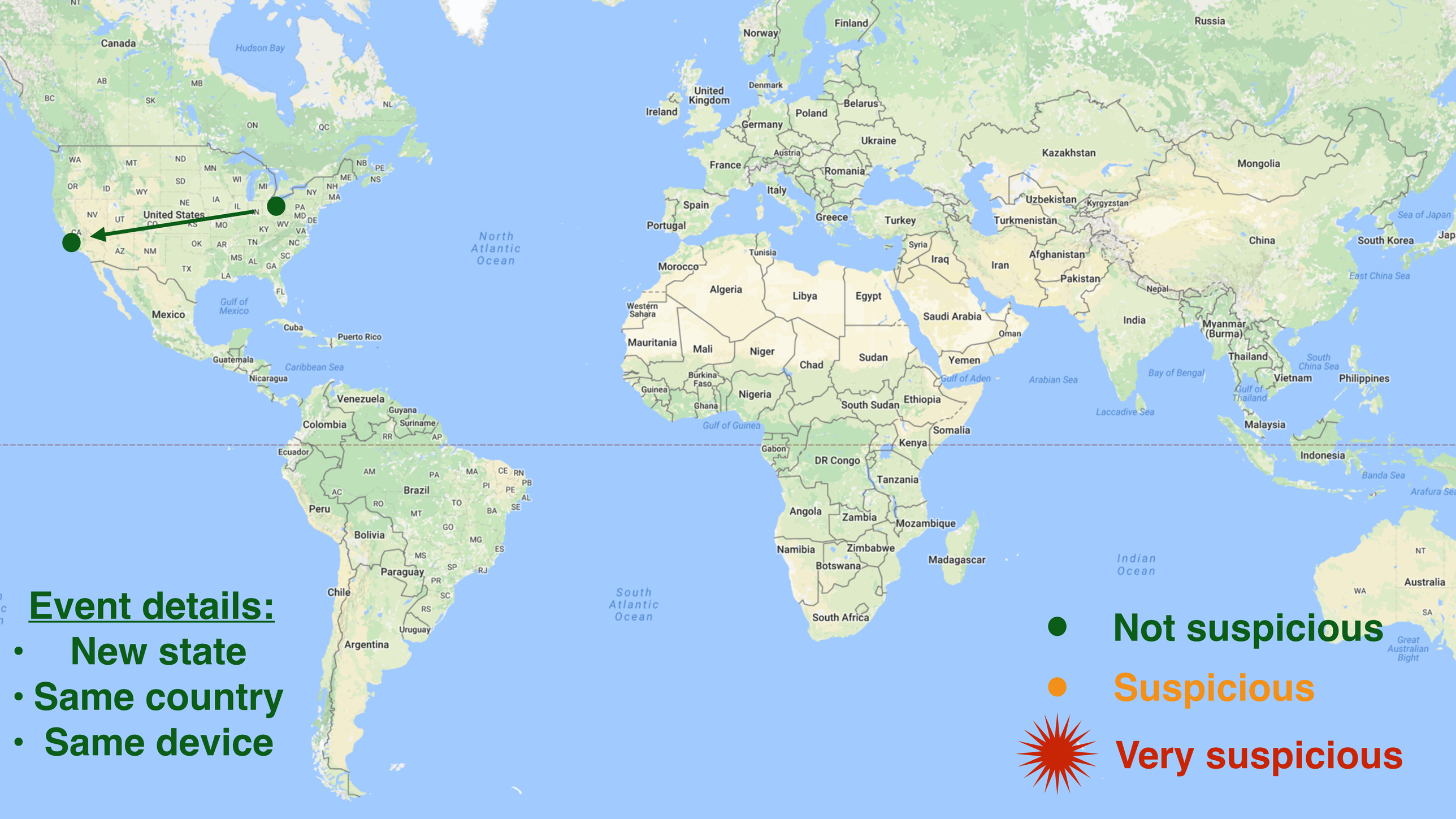


Data and Analysis Pipeline



Data and Analysis Pipeline





Event details:

- New state
- Same country
- Same device

- Not suspicious
- Suspicious
- Very suspicious


Event details:
• New country
• Same device
Not suspicious
Suspicious
Very suspicious

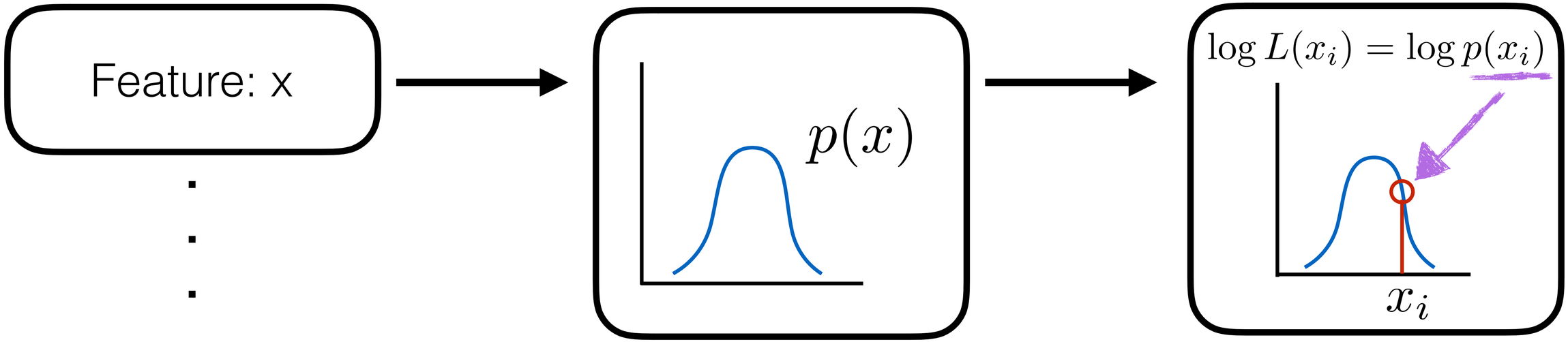

Event details:
• New country
• IP from data center
• New device
• Unusual event timing
Not suspicious
Suspicious
Very suspicious
Canada
United States
Mexico
Russia
China
India
Brazil
Australia
Europe
Africa
North Atlantic Ocean
South Atlantic Ocean
Indian Ocean
Arabian Sea
Caribbean Sea
Gulf of Mexico
Gulf of Guinea
Gulf of Aden
Bay of Bengal
South China Sea
East China Sea
Sea of Japan
Hudson Bay
Laccadive Sea
Banda Sea
Great Australian Bight

Unsupervised anomaly detection

All features

Calculate feature distribution

Likelihood of observed feature value



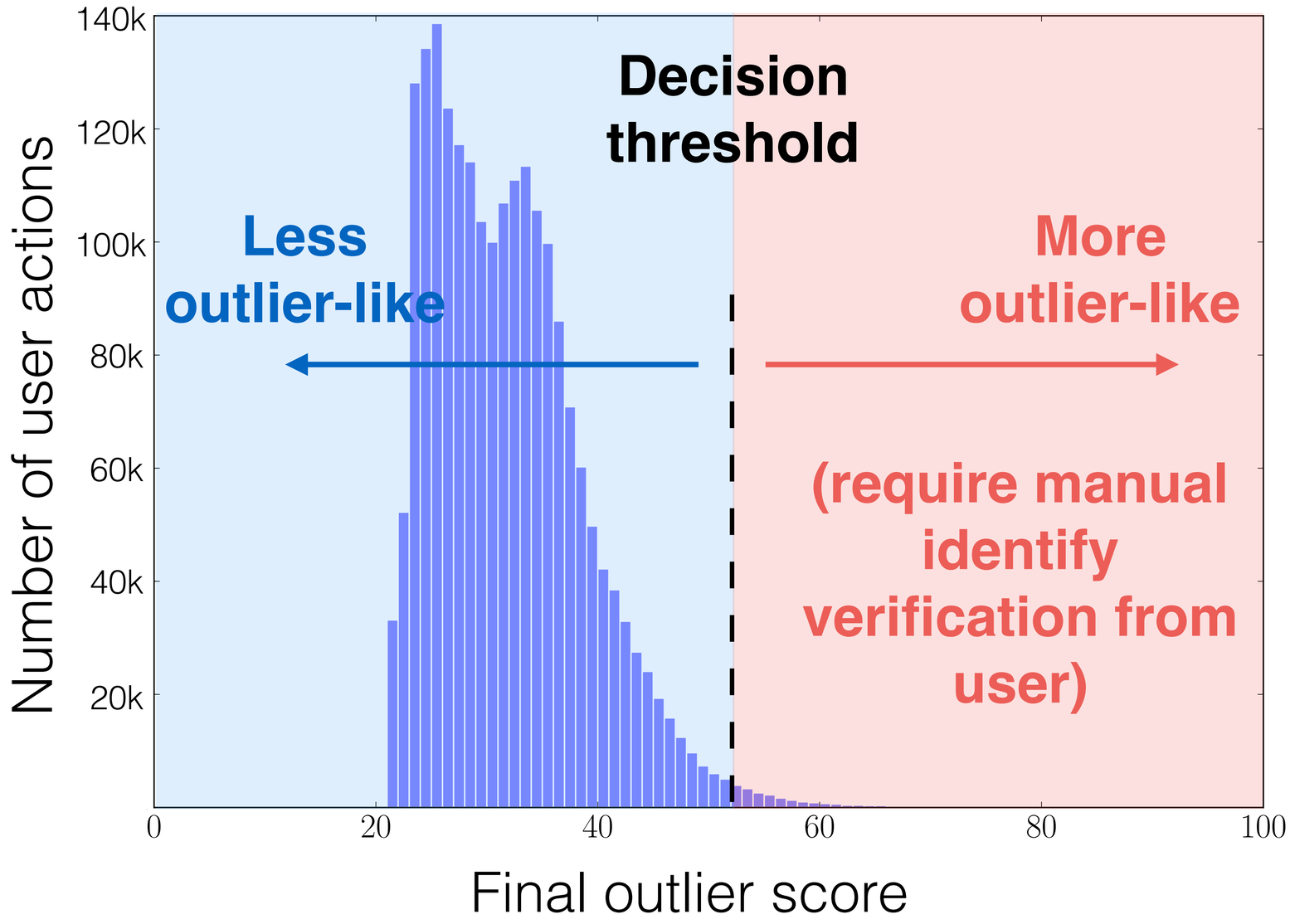
For event i, combine log-likelihood over all features:

$$\log L_i = \log L(x_i) + \log L(y_i) + ...$$

Single final score for outlier detection

Tune decision threshold

- Can **validate** using list of known compromised accounts
- Area under ROC curve = **0.95**
- For this choice of threshold: **79% recall** with **5% false positive rate**



Product delivers measurable improvement

- **Improved** the recall of compromised accounts by >2x compared to the baseline model with the same false positive rate
- **Fast** to train and use
- **Interpretable**
- **Extendable** to include new features
- **More details** available at jverbus.github.io

James Verbus