Anomaly Detection

Finding the **Unusual**

Poul Petersen
CIO, BigML, Inc

What is Anomaly Detection?

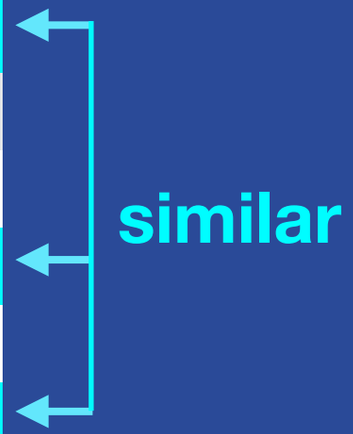
- An **unsupervised** learning technique
  - No labels necessary
  - Useful for finding unusual instances
  - Filtering, finding mistakes, 1-class classifiers
- Finds instances that do not match
  - Customer: big or small spender for profile
  - Medical: healthy patient despite indicative diagnostics
- Defines each unusual instance by an "**anomaly score**"
  - in BigML: 0 = normal, 1 = unusual, and 0.7 ≫ 0.6 > 0.5
  - Standard deviation, distributions, etc

Clusters

date	customer	account	auth	class	zip	amount
Mon	Bob	3421	pin	clothes	46140	135
Tue	Bob	3421	sign	food	46140	401
Tue	Alice	2456	pin	food	12222	234
Wed	Sally	6788	pin	gas	26339	94
Wed	Bob	3421	pin	tech	21350	2459
Wed	Bob	3421	pin	gas	46140	83
Thr	Sally	6788	sign	food	26339	51

Clusters

date	customer	account	auth	class	zip	amount
Mon	Bob	3421	pin	clothes	46140	135
Tue	Bob	3421	sign	food	46140	401
Tue	Alice	2456	pin	food	12222	234
Wed	Sally	6788	pin	gas	26339	94
Wed	Bob	3421	pin	tech	21350	2459
Wed	Bob	3421	pin	gas	46140	83
Thr	Sally	6788	sign	food	26339	51



Anomaly Detection

date	customer	account	auth	class	zip	amount
Mon	Bob	3421	pin	clothes	46140	135
Tue	Bob	3421	sign	food	46140	401
Tue	Alice	2456	pin	food	12222	234
Wed	Sally	6788	pin	gas	26339	94
Wed	Bob	3421	pin	tech	21350	2459
Wed	Bob	3421	pin	gas	46140	83
Thr	Sally	6788	sign	food	26339	51

Anomaly Detection

date	customer	account	auth	class	zip	amount	
Mon	Bob	3421	pin	clothes	46140	135	
Tue	Bob	3421	sign	food	46140	401	
Tue	Alice	2456	pin	food	12222	234	
Wed	Sally	6788	pin	gas	26339	94	
Wed	Bob	3421	pin	tech	21350	2459	←anomaly
Wed	Bob	3421	pin	gas	46140	83	
Thr	Sally	6788	sign	food	26339	51	

- Amount $**2,459** is higher than all other transactions
- It is the only transaction
  - In zip **21350**
  - for the purchase class "**tech**"

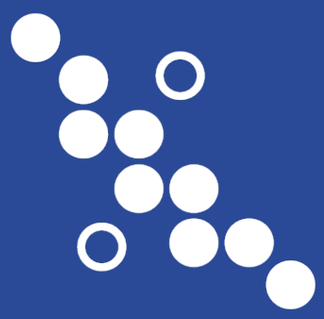
Use Cases

- Unusual instance discovery - "exploration"
- Intrusion Detection - "looking for unusual usage patterns"

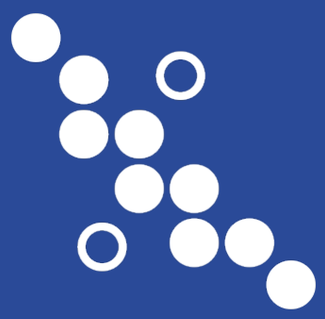
Intrusion Detection

- Dataset of command line history for users
- Data for each user consists of commands, flags, working directories, etc.
- Assumption: Users typically issue the same flag patterns and work in certain directories

GOAL: Identify unusual command line behavior per user and across all users that might indicate an intrusion.

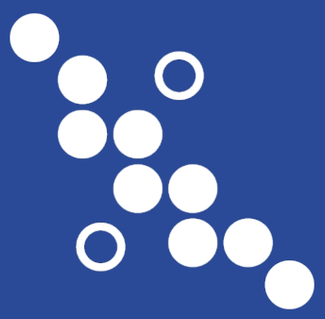


Per User



Per Dir

All User



All Dir

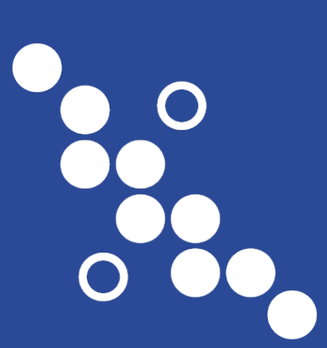
Use Cases

- Unusual instance discovery - "exploration"
- Intrusion Detection - "looking for unusual usage patterns"
- Fraud - "looking for unusual behavior"

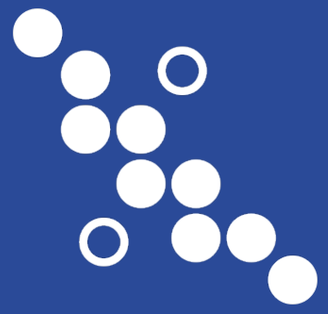
Fraud

- Dataset of credit card transactions
- Additional user profile information

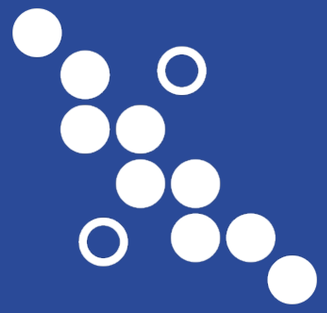
GOAL: Cluster users by profile and use multiple anomaly scores to detect transactions that are anomalous on multiple levels.



Card Level



User Level



Similar User Level

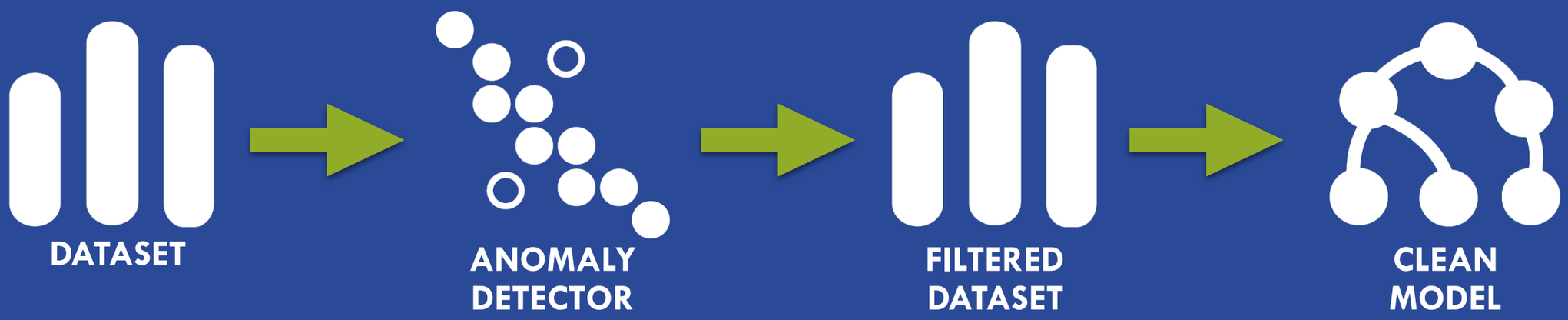
Use Cases

- Unusual instance discovery - "exploration"
- Intrusion Detection - "looking for unusual usage patterns"
- Fraud - "looking for unusual behavior"
- Identify Incorrect Data - "looking for mistakes"
- Remove Outliers - "improve model quality"

Removing Outliers

- Models need to generalize
- Outliers negatively impact generalization

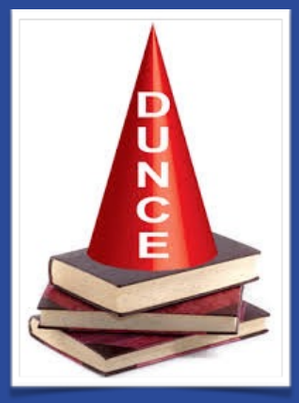
GOAL: Use anomaly detector to identify most anomalous points and then remove them before modeling.



Use Cases

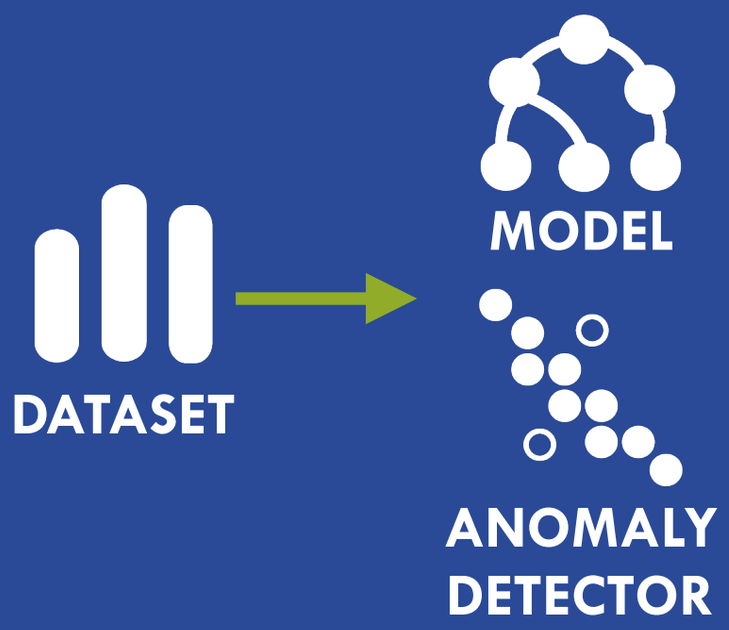
- Unusual instance discovery - "exploration"
- Intrusion Detection - "looking for unusual usage patterns"
- Fraud - "looking for unusual behavior"
- Identify Incorrect Data - "looking for mistakes"
- Remove Outliers - "improve model quality"
- Model Competence / Input Data Drift

Model Competence



- After putting a model it into production, data that is being predicted can become statistically different than the training data.
- Train an anomaly detector at the same time as the model.

At Training Time



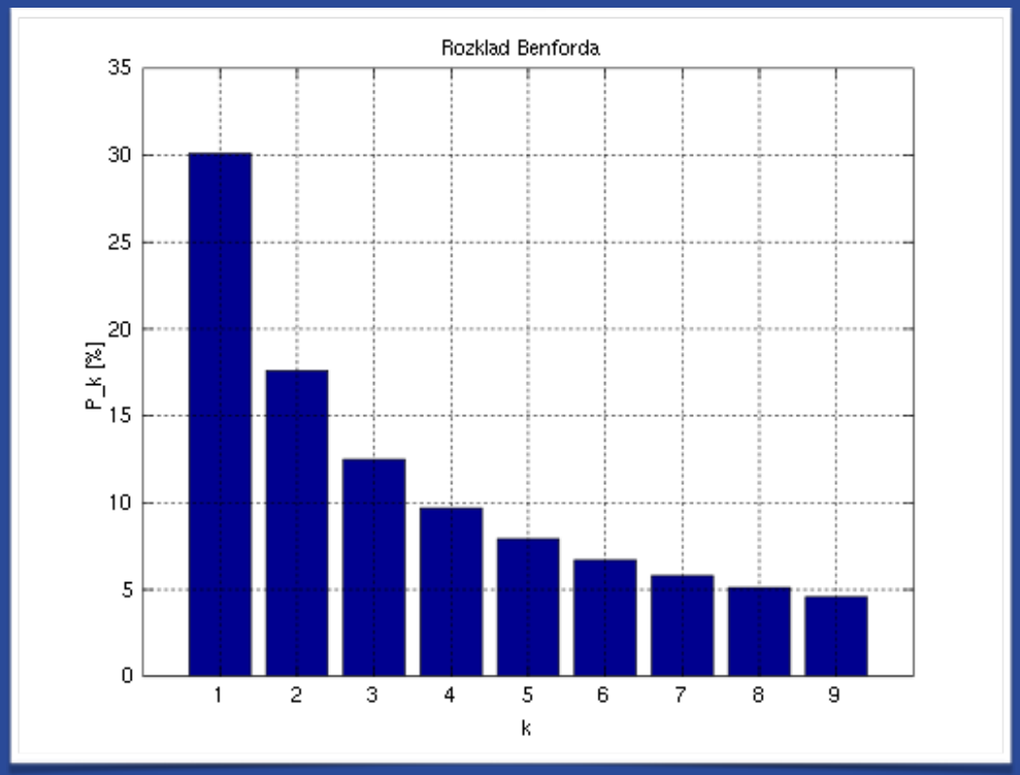
At Prediction Time

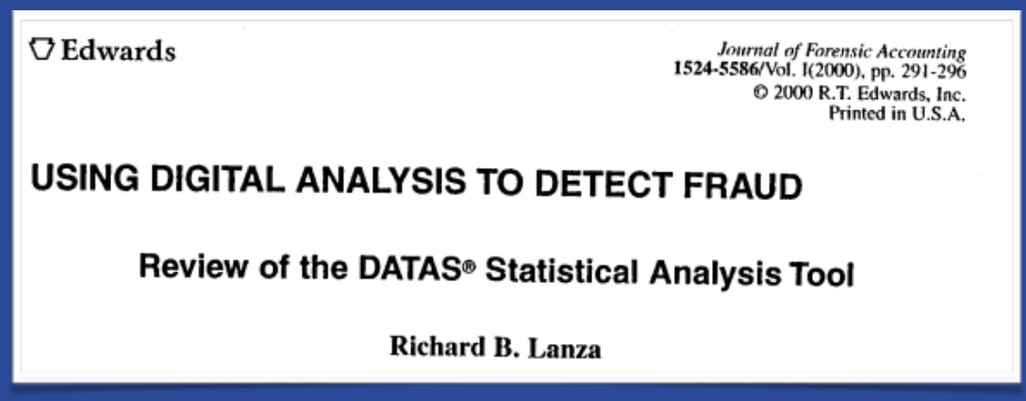
Prediction	T	T
Confidence	0.86	0.84
Anomaly Score	0.5367	0.7124
Competent?	**Y**	**N**

GOAL: For every prediction, compute an anomaly score. If the anomaly score is high, then the model may not be competent and should not be trusted.

Benford's Law

- In real-life numeric sets the small digits occur disproportionately often as leading significant digits.
- Applications include:
  - accounting records
  - electricity bills
  - street addresses
  - stock prices
  - population numbers
  - death rates
  - lengths of rivers
- Available in BigML API





Edwards

Journal of Forensic Accounting
1524-5586/Vol. I(2000), pp. 291-296
© 2000 R.T. Edwards, Inc.
Printed in U.S.A.

USING DIGITAL ANALYSIS TO DETECT FRAUD

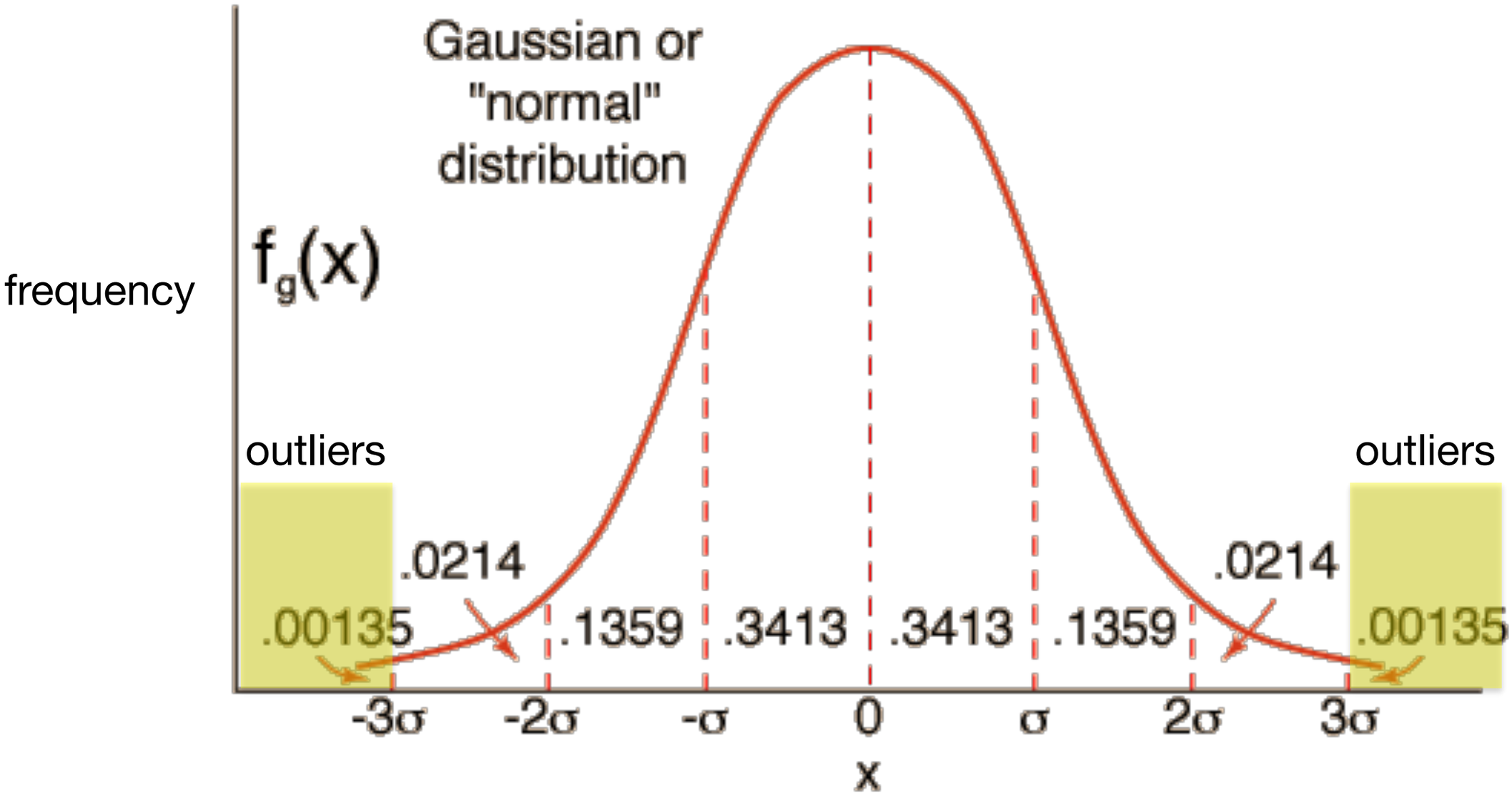
Review of the DATAS® Statistical Analysis Tool

Richard B. Lanza

Univariate Approach

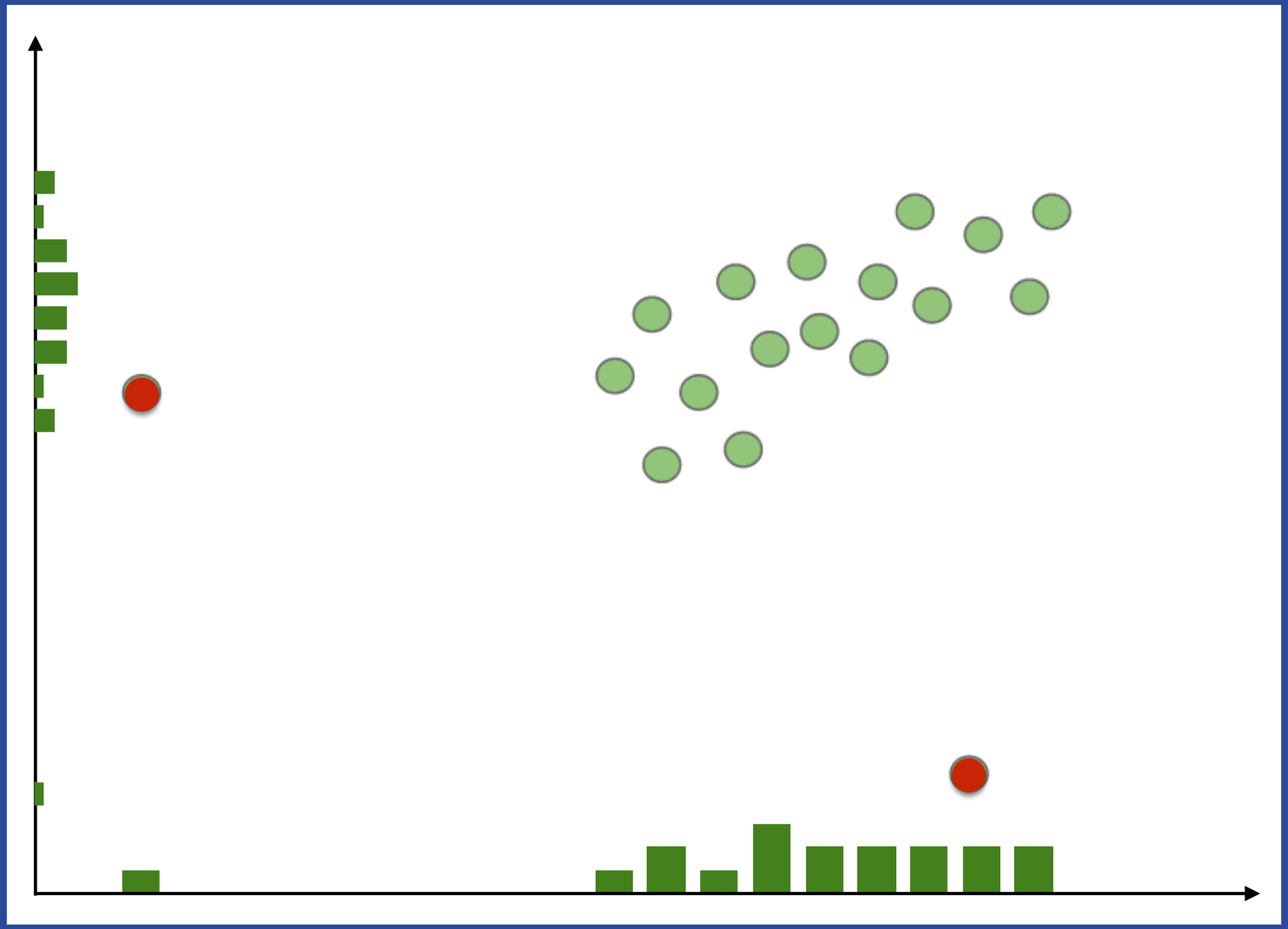
- Single variable: heights, test scores, etc
- Assume the value is distributed “normally”
- Compute standard deviation
  - a measure of how “spread out” the numbers are
  - the square root of the variance (The average of the squared differences from the Mean.)
  - Depending on the number of instances, choose a “multiple” of standard deviations to indicate an anomaly. A multiple of 3 for 1000 instances removes ~ 3 outliers.

Univariate Approach

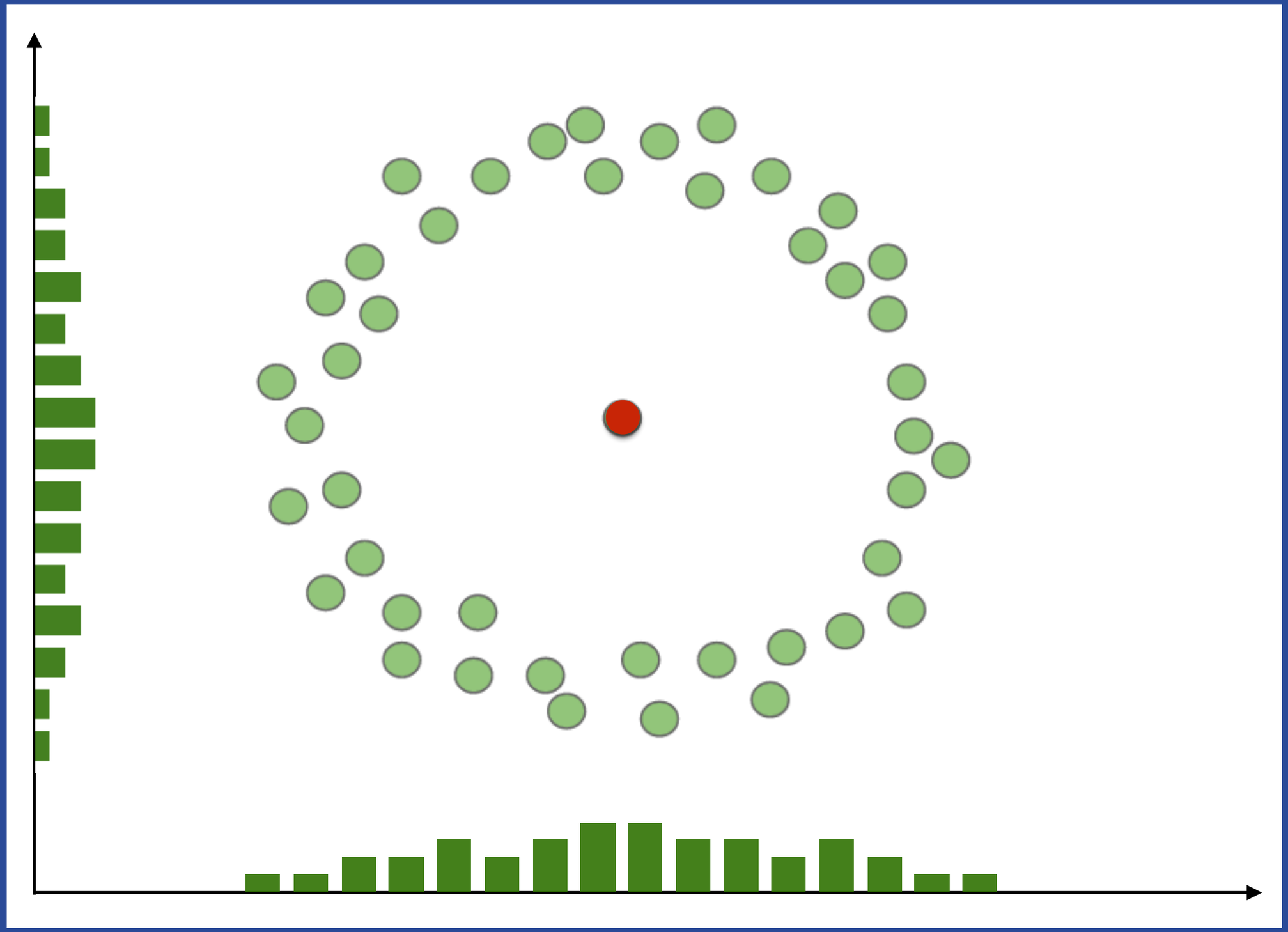


- *Available in BigML API*

Multivariate Matters

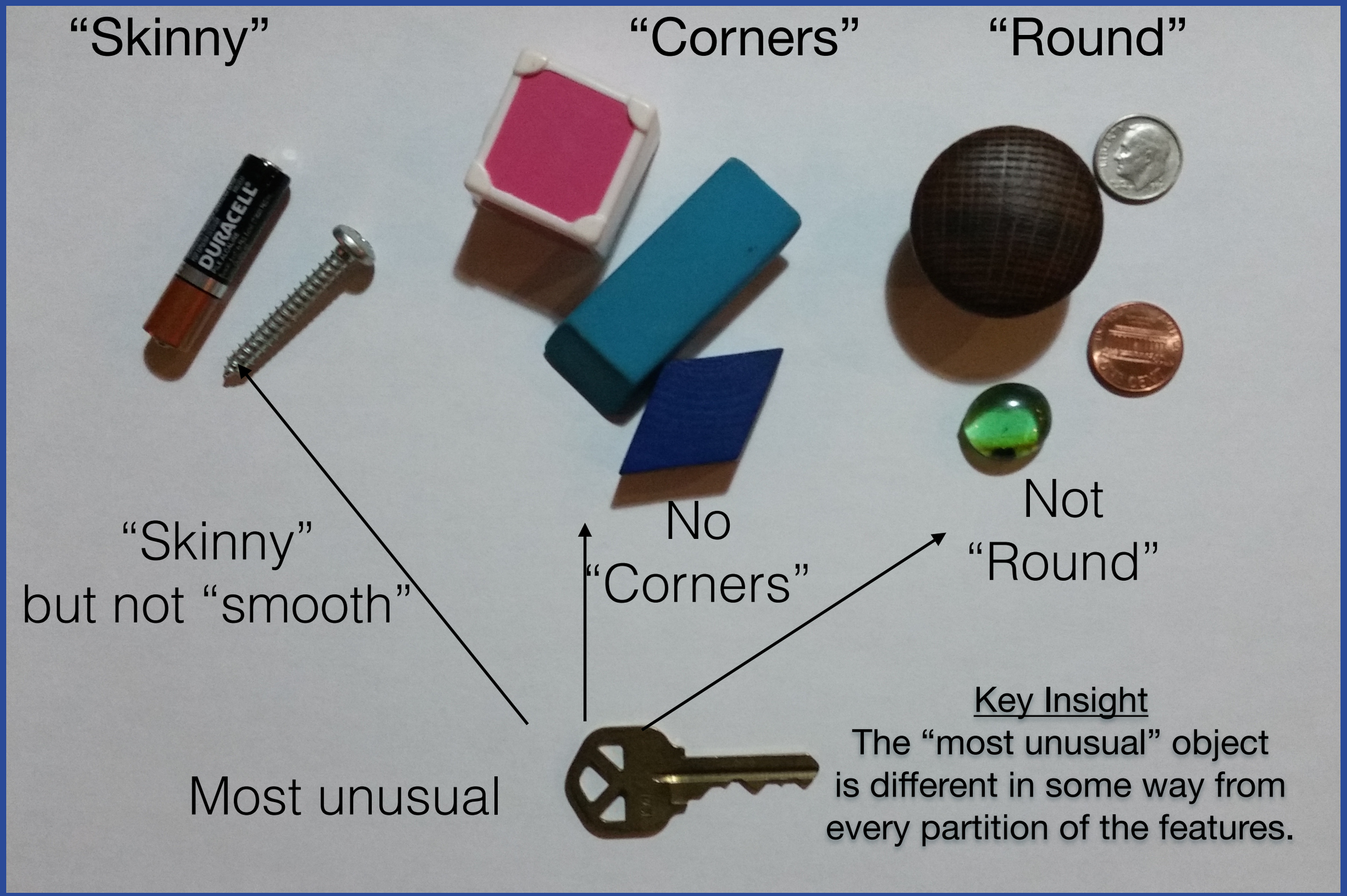


Multivariate Matters



Human Expert

Human Expert



Human Expert

- Human used prior knowledge to select possible features that separated the objects.
- “round”, “skinny”, “smooth”, “corners”
- Items were then separated based on the chosen features
- Each cluster was then examined to see which object fit the least well in its cluster and did not fit any other cluster

Human Expert

Create **features** that capture these object differences

- **Length/Width**
  - greater than 1 => “skinny”
  - equal to 1 => “round”
  - less than 1 => invert
- **Number of Surfaces**
  - distinct surfaces require “edges” which have corners
  - easier to count
- **Smooth** - true or false

Anomaly Features

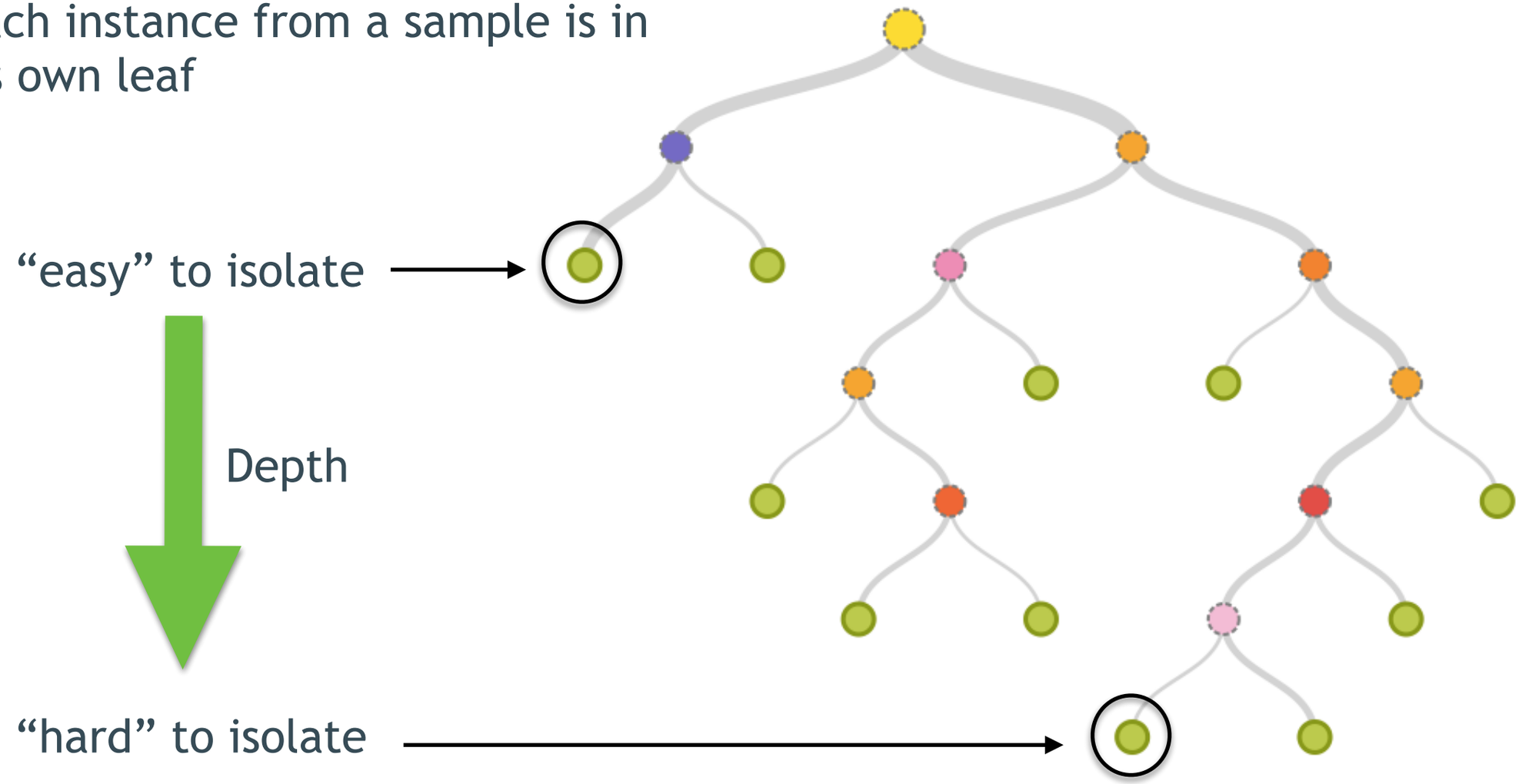
Object	Length / Width	Num Surfaces	Smooth
penny	1	3	TRUE
dime	1	3	TRUE
knob	1	4	TRUE
eraser	2.75	6	TRUE
box	1	6	TRUE
block	1.6	6	TRUE
screw	8	3	FALSE
battery	5	3	TRUE
key	4.25	3	FALSE
bead	1	2	TRUE

Random Splits

Know that “splits” matter - don’t know the order

Isolation Forest

Grow a random decision tree until each instance from a sample is in its own leaf



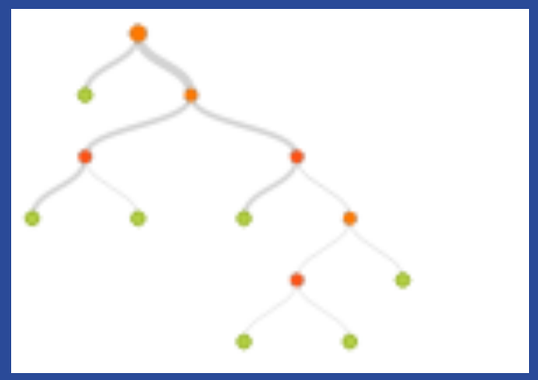
Now repeat the process several times and use average Depth to compute anomaly score: 0 (similar) -> 1 (dissimilar)

Isolation Forest Scoring

	f_1	f_2	f_3
i_1	red	cat	ball
i_2	red	cat	ball
i_3	red	cat	box
i_4	blue	dog	pen

For the instance, i_2

D = 3



D = 6

D = 2

Find the depth in each tree

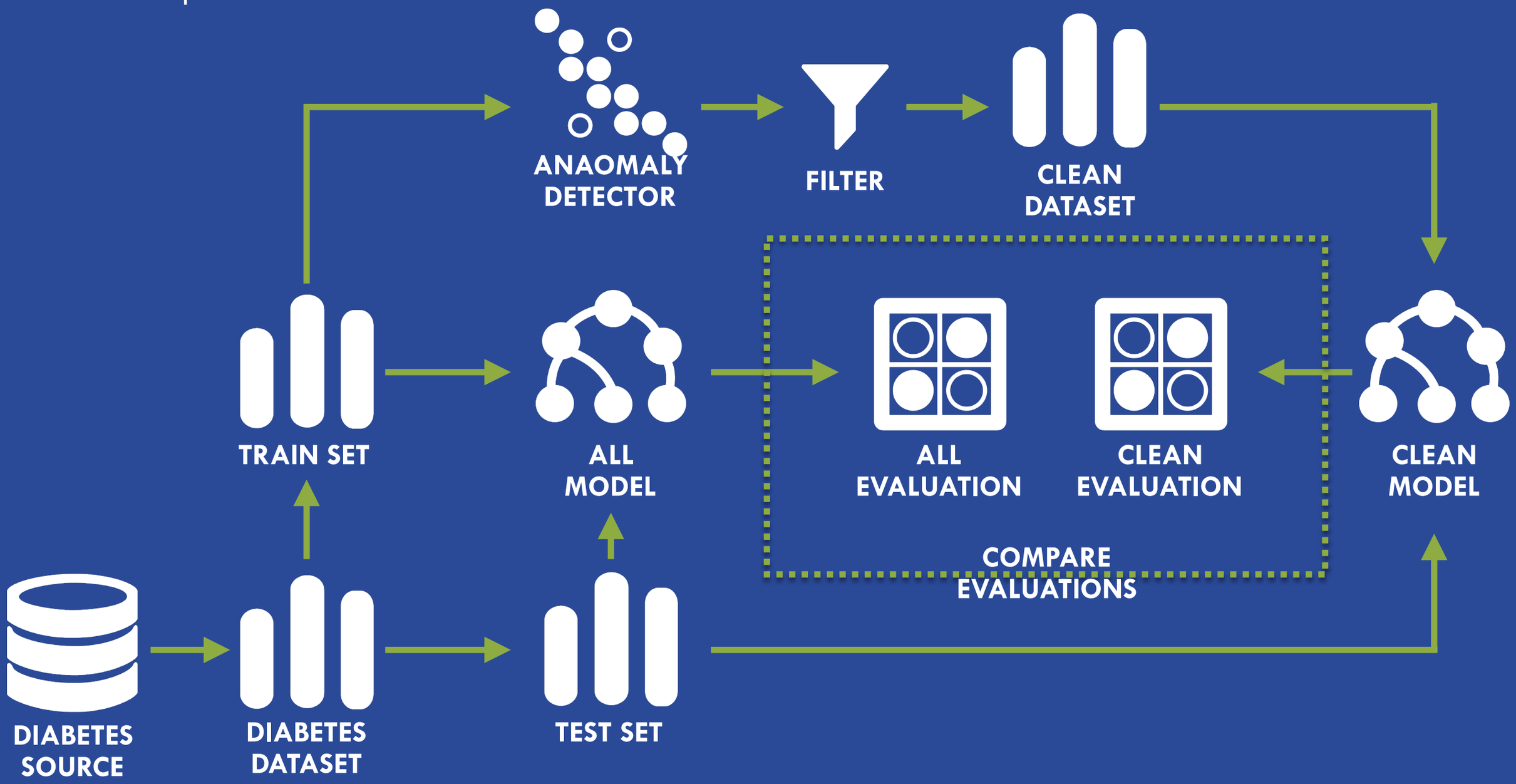
S=0.45

Map avg depth to final score

Anomaly Demo

Your Turn!

- Build an Anomaly Detector for the **Diabetes 80% Training** set
- Filter out the **top 2 Anomalies**
- Build a "Clean" model from the new Dataset and Evaluate it
- Which performs better?



1-Class Classifier?

- You place an advertisement in a local newspaper
- You collect demographic information about all responders
- Now you want to market in a new locality with direct letters
- To optimize mailing costs, need to predict who will respond
- But, can not distinguish not interested from didn't see the ad
- Train an anomaly detector on the 1-class data
- Pick the households with the lowest scores for mailing:
  - If a household has a low anomaly score, then they are "similar" to enough of your positive responders and therefore may respond as well
  - If an individual has a high anomaly score, then they are dissimilar from all previous responders and therefore are less likely to respond.


bigml