

Data Science in the Wild

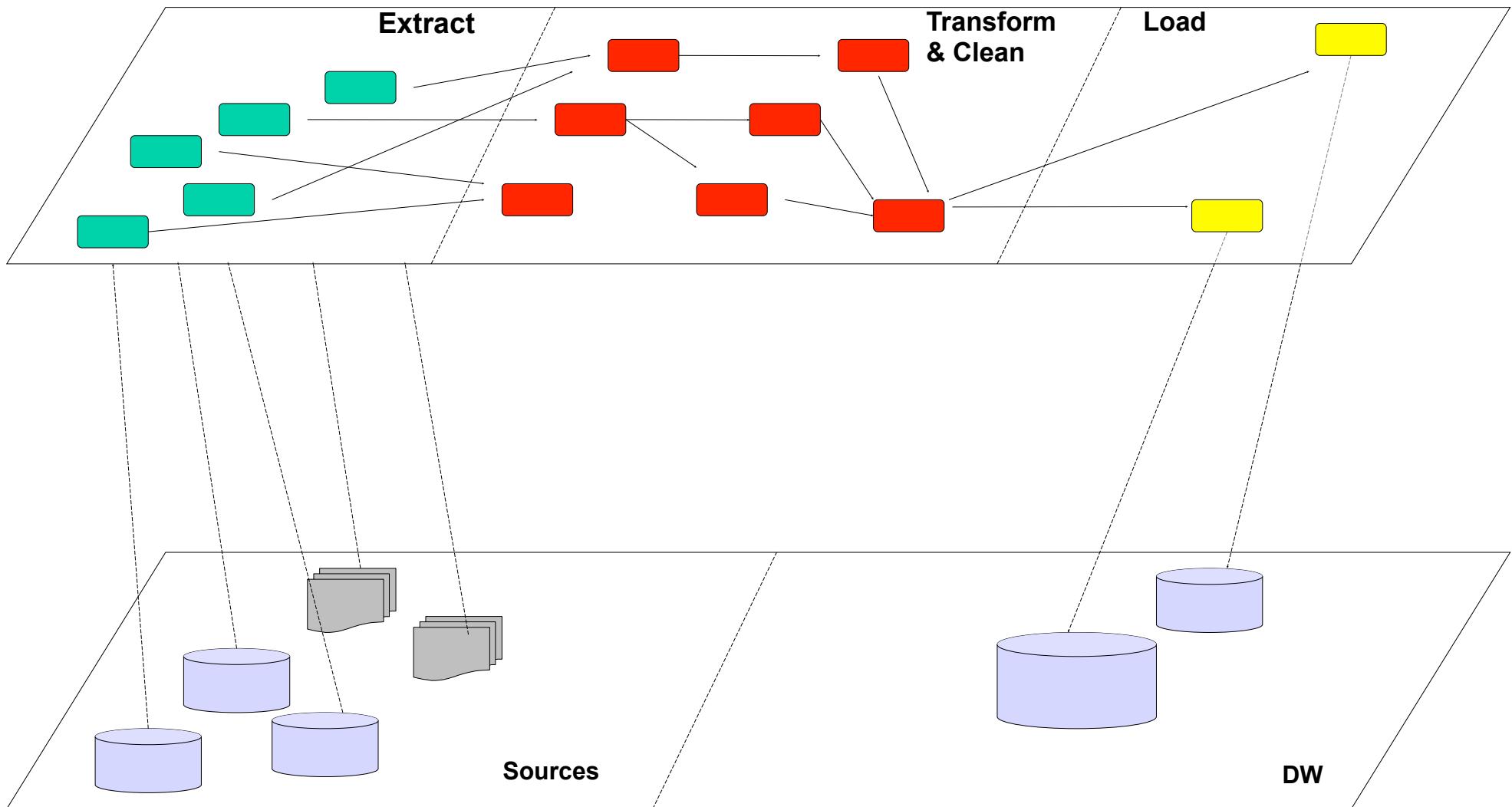
Lecture 5: ETL - Extract, Transform, Load - 2

Eran Toch



**CORNELL
TECH**

ETL Pipeline



Agenda

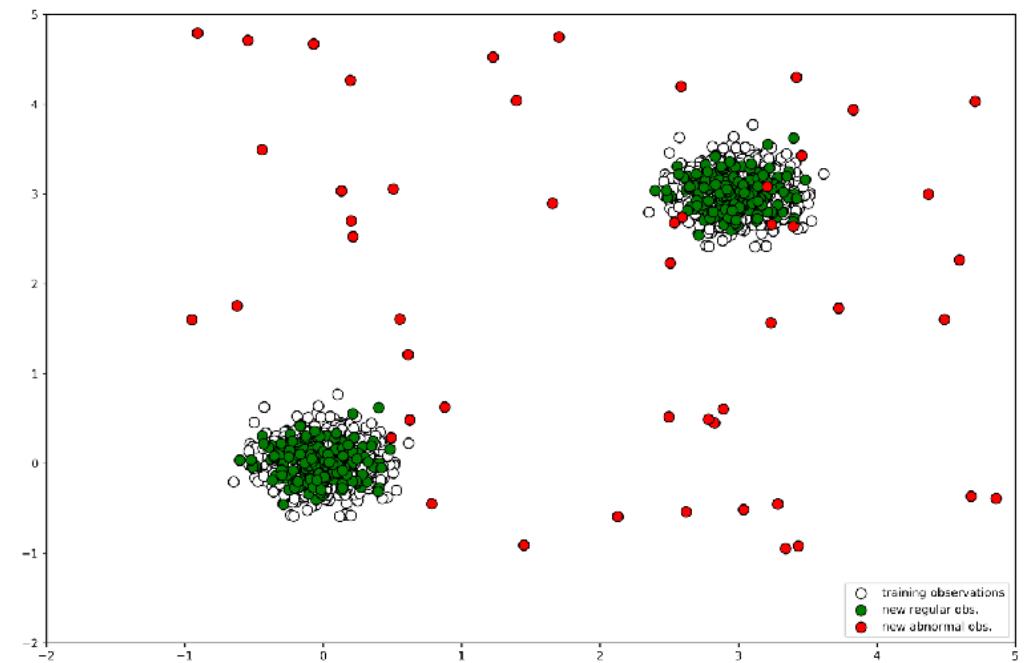
1. Unsupervised outlier detection
2. Labeling data with crowdsourcing
3. Quality assurance of labeling
4. Data sources



Outliers

Returning to our definition of outliers:

“An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different statistical mechanism” Hawkins (1980)



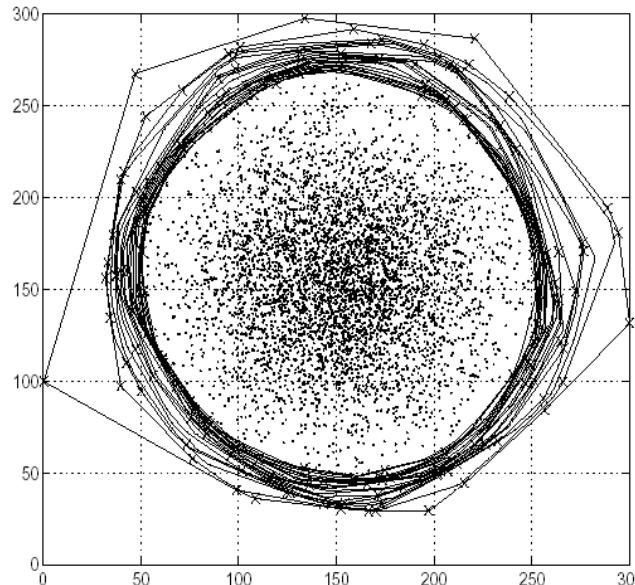
Handling Outliers

- First, identify if we have outliers
- Prepare a strategy:
 - Does our business cares about outliers?
 - Should we build a mechanism for the average case?
 - Some businesses are all about outliers
- What can be done?
 - Remove them
 - Handle them differently
 - Transform the value (e.g., switching to $\log(x)$)

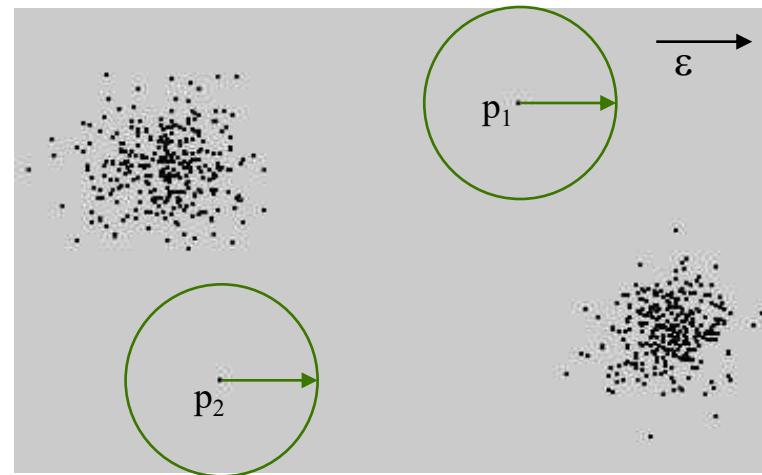
Limitations of statistical methods

- These simple methods are a good start, but they are not too robust
 - The mean and standard deviation are highly affected by outliers
 - These values are computed for the complete data set (including potential outliers)
 - Therefore, it is particularly problematic in small datasets
 - And are not robust for multi-dimensional data

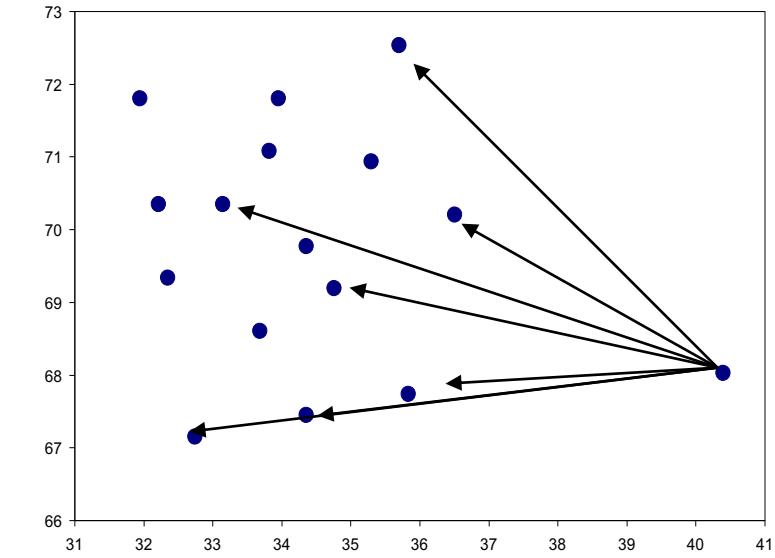
Other Approaches



Density-based
approaches
(DBSCAN, LOF)



Distance-based
Approaches (K-NN,
K-Means)



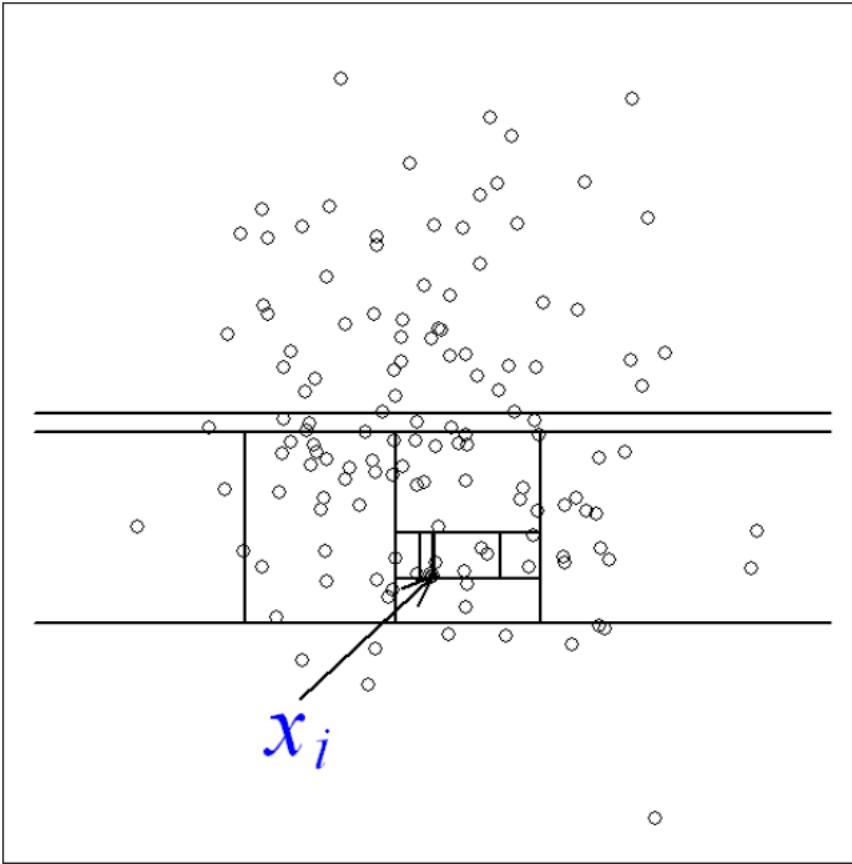
Parametric
Approaches (z-
scores etc)

Outlier detection with Isolation Forests

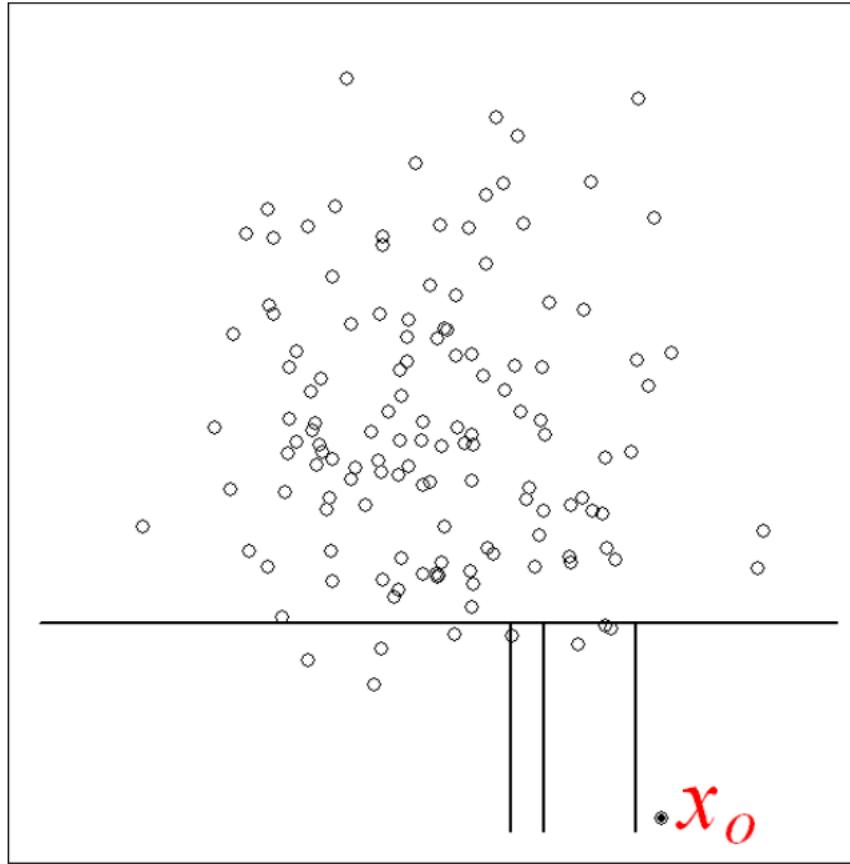
- Isolation forests is a method for multidimensional outlier detection using random forest
- The intuition is that outliers are less frequent than regular observations and are different from them in terms of values
- In random partitioning, they should be identified closer to the root of the tree (shorter average path length, i.e., the number of edges an observation must pass in the tree going from the root to the terminal node), with fewer splits necessary.

F. T. Liu, et al., Isolation Forest, Data Mining, 2008. ICDM'08, Eighth IEEE International Conference

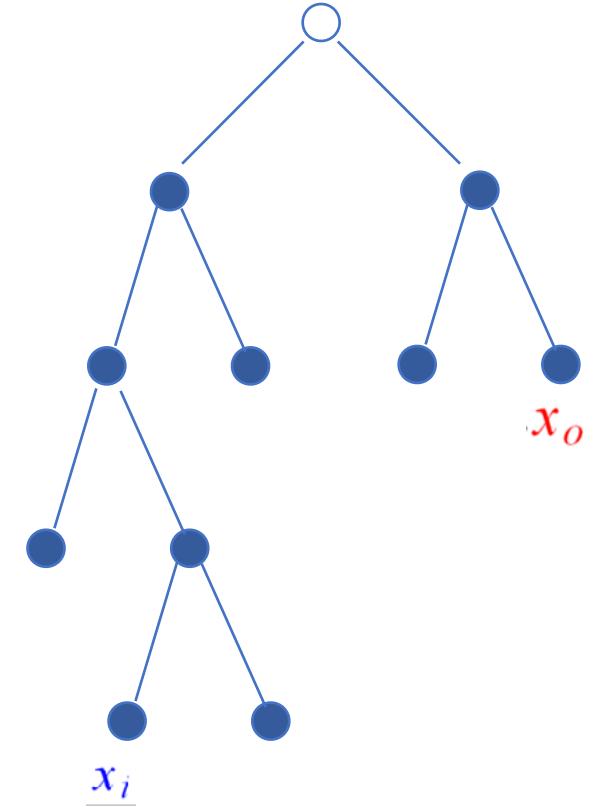
Partitioning



(a) Isolating x_i



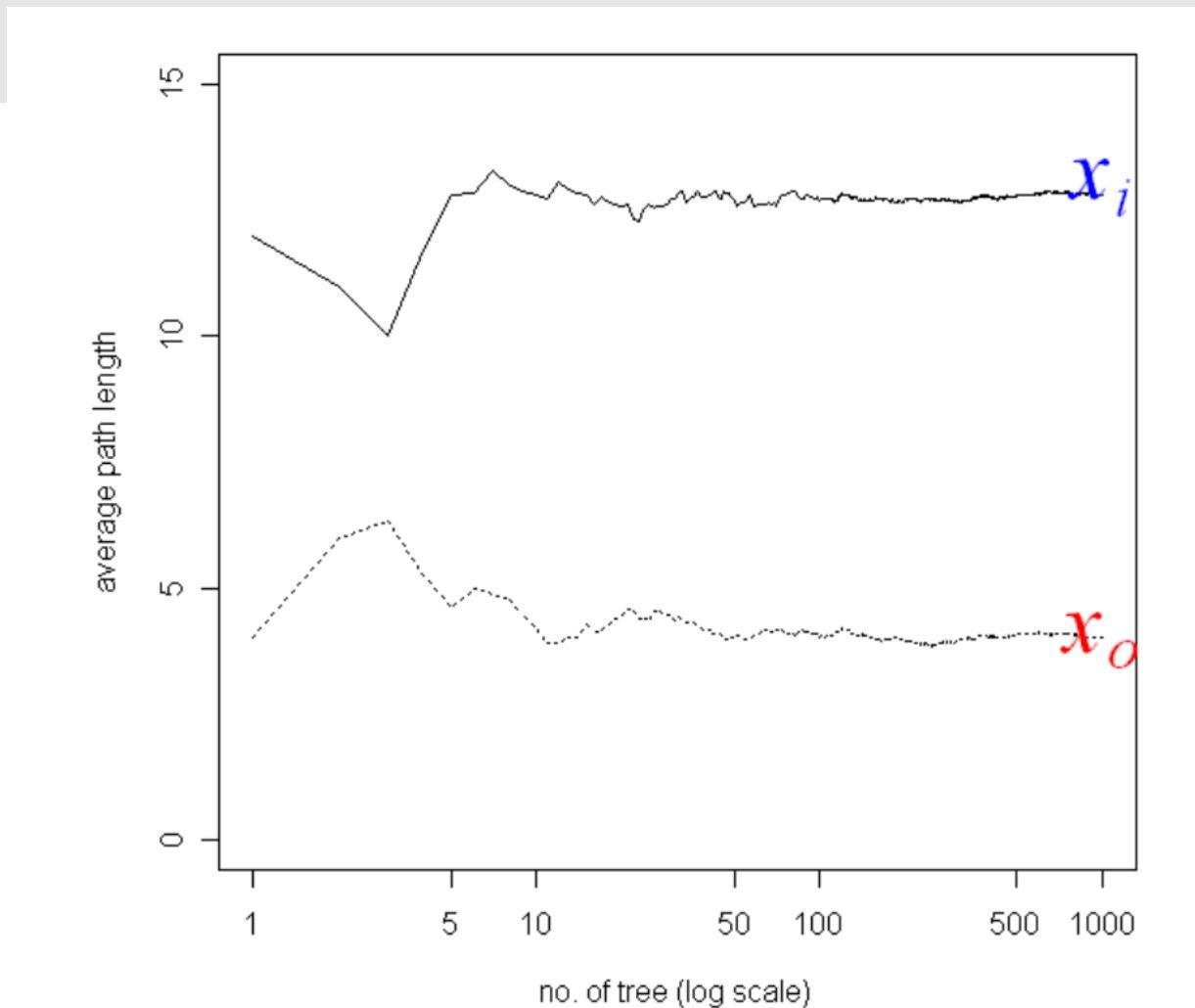
(b) Isolating x_o



A normal point (on the left) requires more partitions to be identified than an abnormal point (right).

Partitioning and outliers

- The number of partitions required to isolate a point is equivalent to the traversal of path length from the root node to a terminating node
- Since each partition is randomly generated, individual trees are generated with different sets of partitions
- The path length is averaged over a number of trees



(c) Average path lengths converge

Anomaly Score

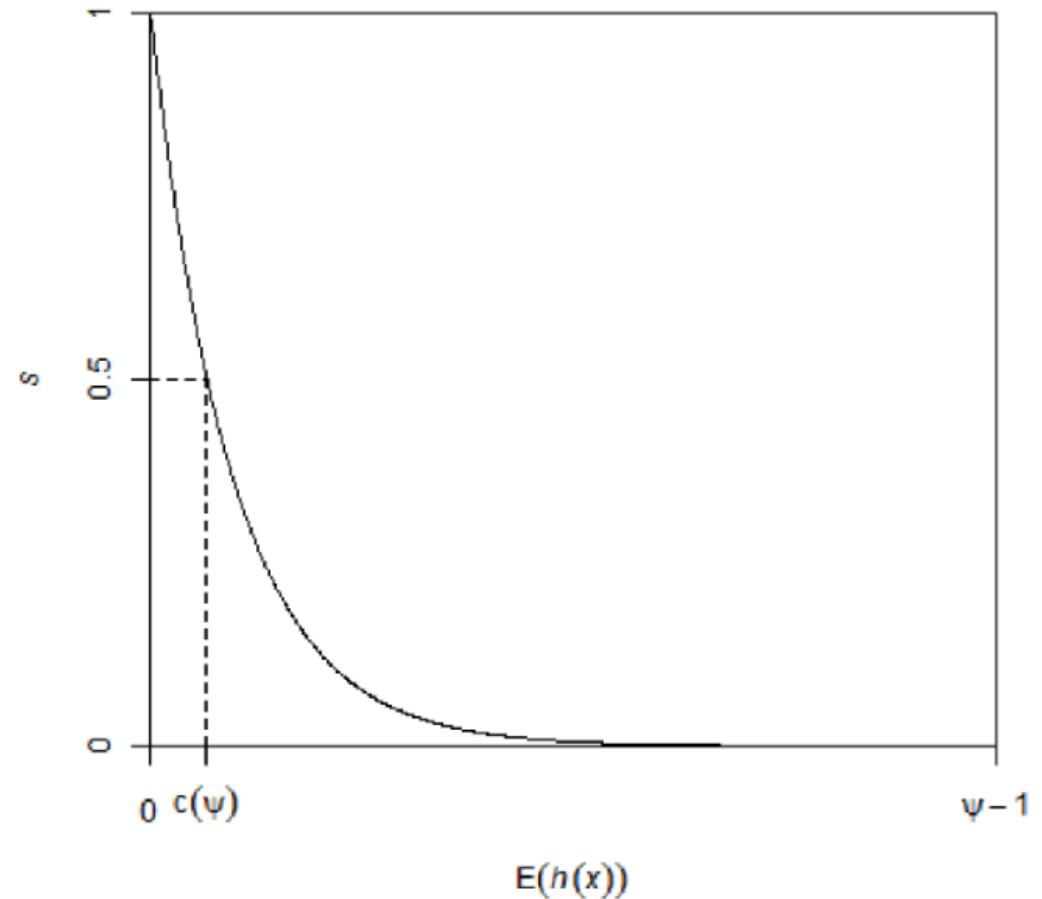
- $h(x)$ is the path length of observation x
- $c(\psi)$ is the average path length of unsuccessful search in a Binary Search Tree
- ψ is the number of external nodes

$$s(x, \psi) = 2^{-\frac{E(h(x))}{c(\psi)}}$$

1. when $E(h(x)) \rightarrow 0$, $s \rightarrow 1$;
2. when $E(h(x)) \rightarrow \psi - 1$, $s \rightarrow 0$; and
3. when $E(h(x)) \rightarrow c(\psi)$, $s \rightarrow 0.5$.

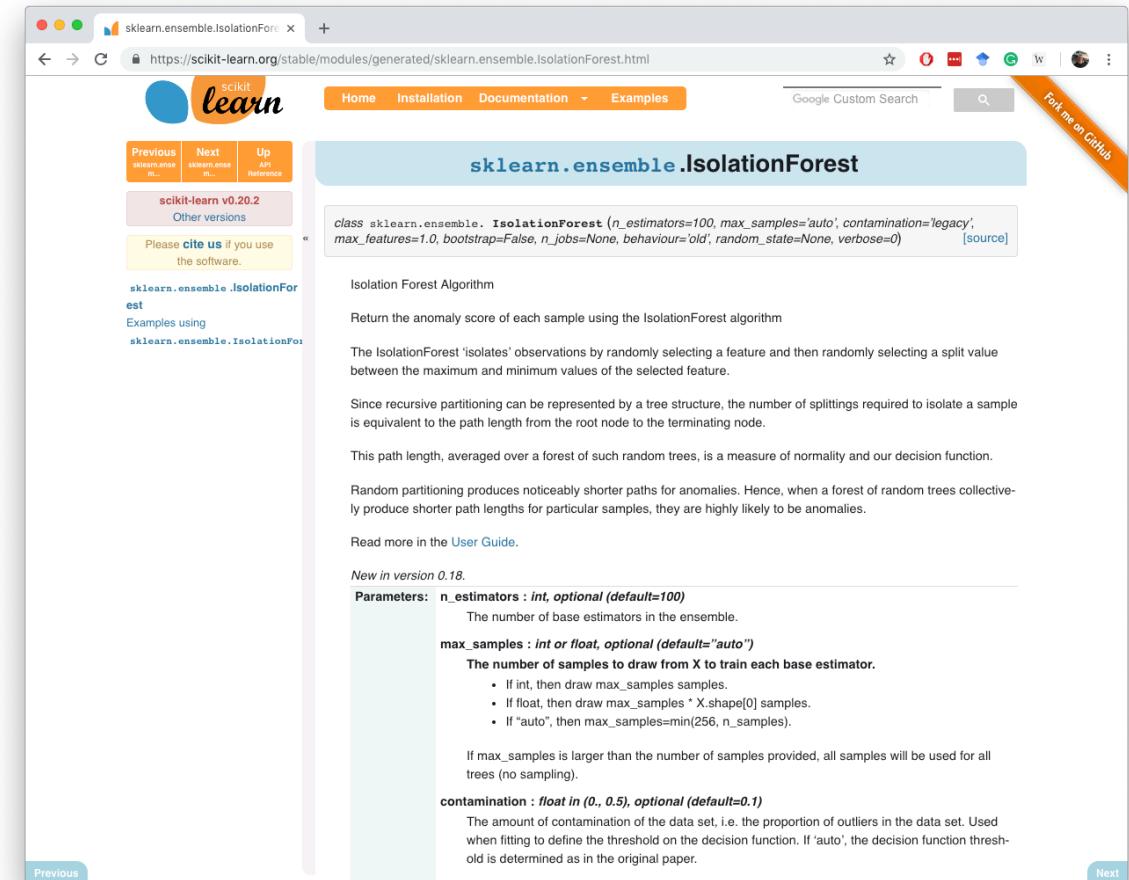
Anomalies and s

1. If instances return s very close to 1, then they are definitely anomalies,
2. If instances have s much smaller than 0.5, then they are quite safe to be regarded as normal instances, and
3. If all the instances return $s \approx 0.5$, then the entire sample does not really have any distinct anomaly.



Implementation

- Isolation Forest (IF) became available in scikit-learn v0.18
- The algorithm includes two steps:
 - Training stage involves building iForest
 - Testing stage involves passing each data point through each tree to calculate average number of edges required to reach an external node



```

# importing libraries ----
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pylab import savefig
from sklearn.ensemble import IsolationForest
# Generating data ----

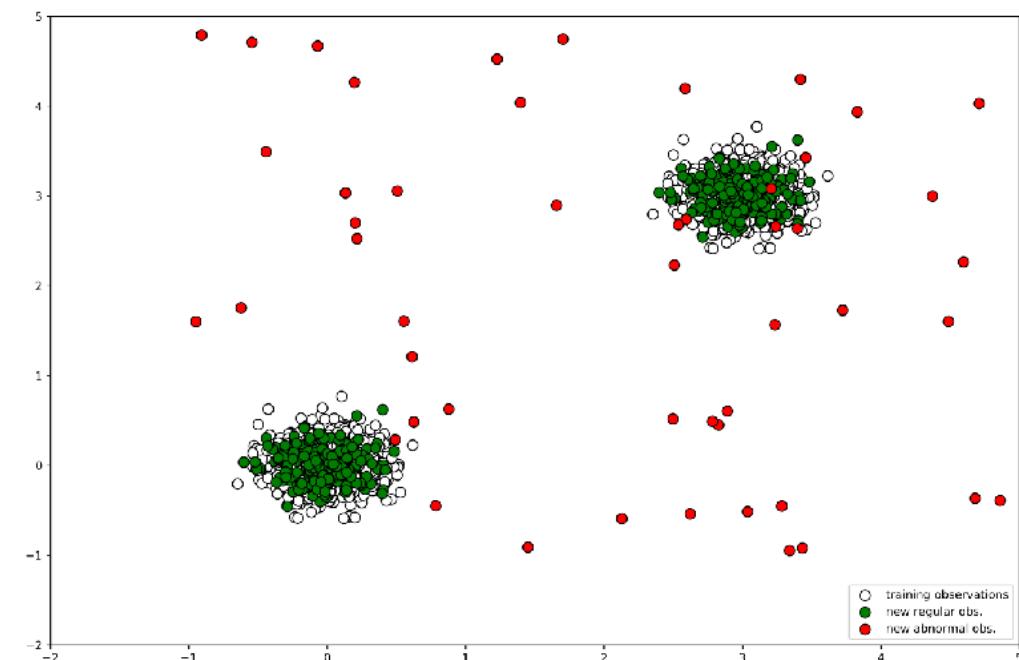
rng = np.random.RandomState(42)

# Generating training data
X_train = 0.2 * rng.randn(1000, 2)
X_train = np.r_[X_train + 3, X_train]
X_train = pd.DataFrame(X_train, columns = ['x1', 'x2'])

# Generating new, 'normal' observation
X_test = 0.2 * rng.randn(200, 2)
X_test = np.r_[X_test + 3, X_test]
X_test = pd.DataFrame(X_test, columns = ['x1', 'x2'])

# Generating outliers
X_outliers = rng.uniform(low=-1, high=5, size=(50, 2))
X_outliers = pd.DataFrame(X_outliers, columns = ['x1', 'x2'])

```



Training the Isolation Forest

Isolation Forest ----

```
# training the model
clf = IsolationForest(max_samples=100, contamination = 0.1, random_state=rng)
clf.fit(X_train)

# predictions
y_pred_train = clf.predict(X_train)
y_pred_test = clf.predict(X_test)
y_pred_outliers = clf.predict(X_outliers)

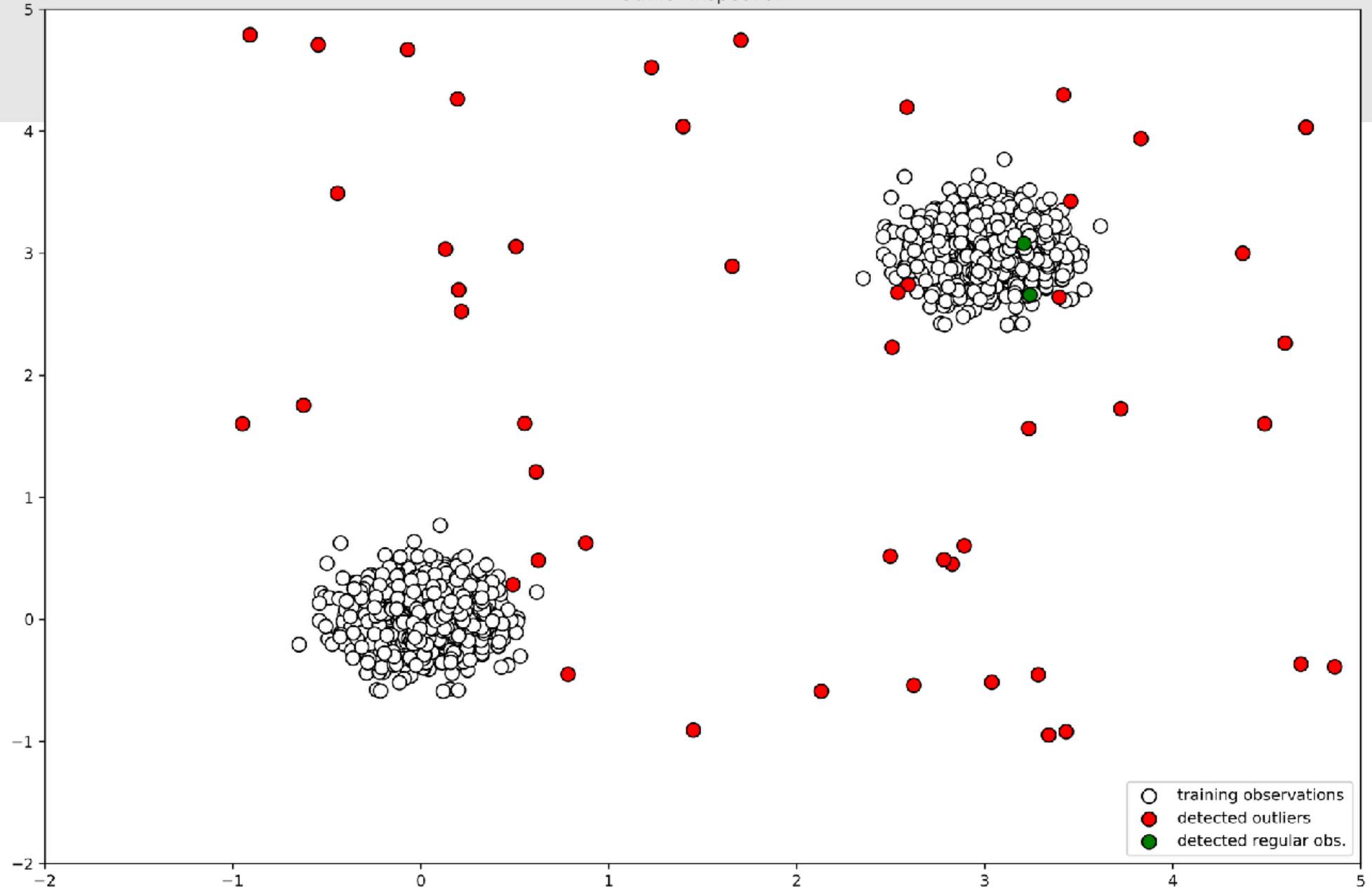
# new, 'normal' observations
print("Accuracy:", list(y_pred_test).count(1)/y_pred_test.shape[0])
Accuracy: 0.93

# outliers
print("Accuracy:", list(y_pred_outliers).count(-1)/y_pred_outliers.shape[0])
Accuracy: 0.96
```

Specifies the percentage of observations we believe to be outliers

Result

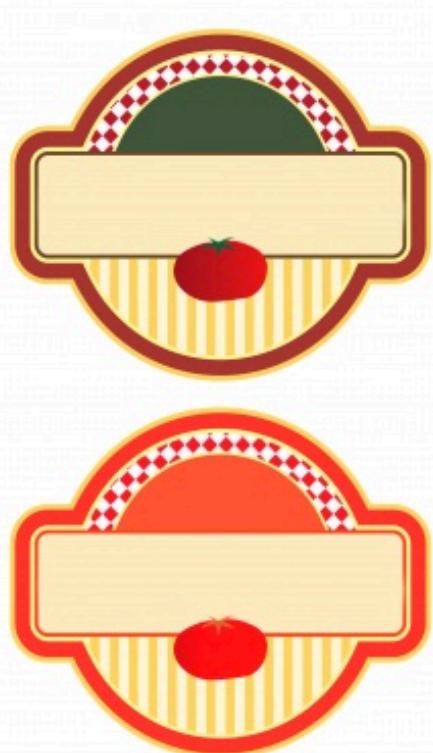
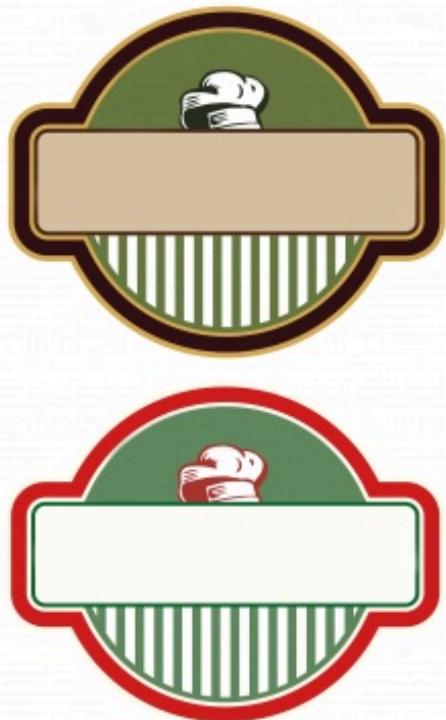
Outlier Inspection



Summary

- Isolation Forest is an outlier detection technique that identifies anomalies instead of normal observations
- Similarly to Random Forest it is built on an ensemble of binary (isolation) trees
- It can be scaled up to handle large, high-dimensional datasets

Labels



- Having good labels is essential for
 - Supervised learning
 - Quality assurance
- But where do we get our labels from?
- How to control the quality?

Where do labels come from?



Other databases

Users

```
ideas = []
for (var i = 0; i < 5; i++) {
    idea = mturk.prompt(
        "What's fun to see in New York City?
        Ideas so far: " + ideas.join(", "))
    ideas.push(idea)
}

ideas.sort(function (a, b) {
    v = mturk.vote("Which is better?", [a, b])
    return v == a ? -1 : 1
})
```

Crowdsourcing

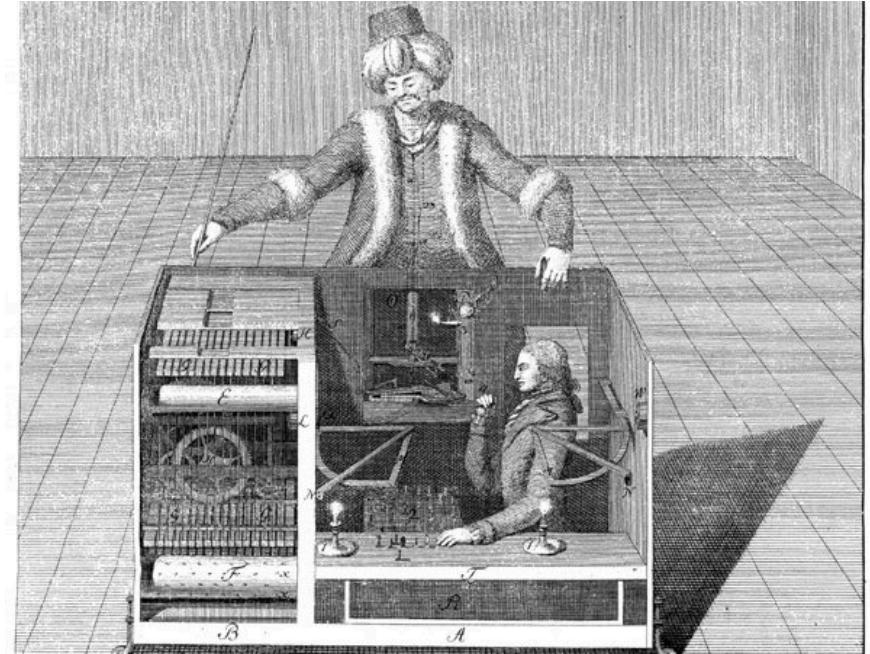
Von Ahn, Luis, et al. "recaptcha: Human-based character recognition via web security measures." *Science* 321.5895 (2008): 1465-1468.

Paid crowdsourcing

- Jeff Howe created the term for his article in the Wired magazine "The Rise of Crowdsourcing" (2006)
- Small scale work by people from a crowd or a community (an online audience)
- Mostly fee-based systems
- Some systems:
 - Amazon Mechanical Turk
 - Prolific Academic (prolific.ac)
 - Daemo (crowdresearch.stanford.edu)
 - microworkers.com
 - ClickWorker

Amazon Mechanical Turk

- Amazon Mechanical Turk (MTurk) is a crowdsourcing Internet marketplace
- Started as a service that Amazon itself needed for cleaning up individual product pages
- The name Mechanical Turk is a historical reference to an 18th century chess-playing device (according to legend, Jeff Bezos had thought about the name)



<https://www.quora.com/What-is-the-story-behind-the-creation-of-Amazons-Mechanical-Turk>

How Mechanical Turk works

- **Requesters** are able to post jobs known as Human Intelligence Tasks (HITs)
- **Workers** (also known as Turkers) can then decide to take them or not
- Workers and requesters have reputation scores
- Requesters can accept or reject the work (which affects the requester reputation). They can also decide to give a bonus.

The screenshot shows the Amazon Mechanical Turk homepage. At the top, there's a navigation bar with links for 'Your Account', 'HITs', 'Qualifications', 'Introduction', 'Dashboard', 'Status', and 'Account Settings'. A link to sign in as a 'Worker' or 'Requester' is also present. The main content area has two main sections: 'Make Money by working on HITs' for workers and 'Get Results from Mechanical Turk Workers' for requesters. Both sections include descriptive text, icons, and a 'Get Started' button.

Make Money by working on HITs

HITs - Human Intelligence Tasks - are individual tasks that you work on. [Find HITs now.](#)

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work

Find an interesting task → **Work** → **Earn money**

[Find HITs Now](#)

or [learn more about being a Worker](#)

Get Results from Mechanical Turk Workers

Ask workers to complete HITs - Human Intelligence Tasks - and get results using Mechanical Turk. [Get Started.](#)

As a Mechanical Turk Requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results

Fund your account → **Load your tasks** → **Get results**

[Get Started](#)

Submitting a HIT

amazon mechanical turk beta REQUESTER

Home Create Manage Developer Help

New Project New Batch with an Existing Project Create HITs individually

Start a New Project

Categorization
Data Collection
Moderation of an Image
Sentiment
Survey
Survey Link
Tagging of an Image
Transcription from A/V
Transcription from an image
Writing
Other

Example of Tagging of an Image

Provide 3 tags for this image.

Instructions:

- You must provide 3 tags for this image.
- Each tag must be a single word
- No tag can be longer than 25 characters
- The tags must describe the image, the contents of the image, or some relevant context.

Image:



Tag 1:

Tag 2:

Tag 3:

Create Project »



Mechanical Turk Project

If you're using the turk, Be sure to copy the text back into the HIT page so that you can be credited.

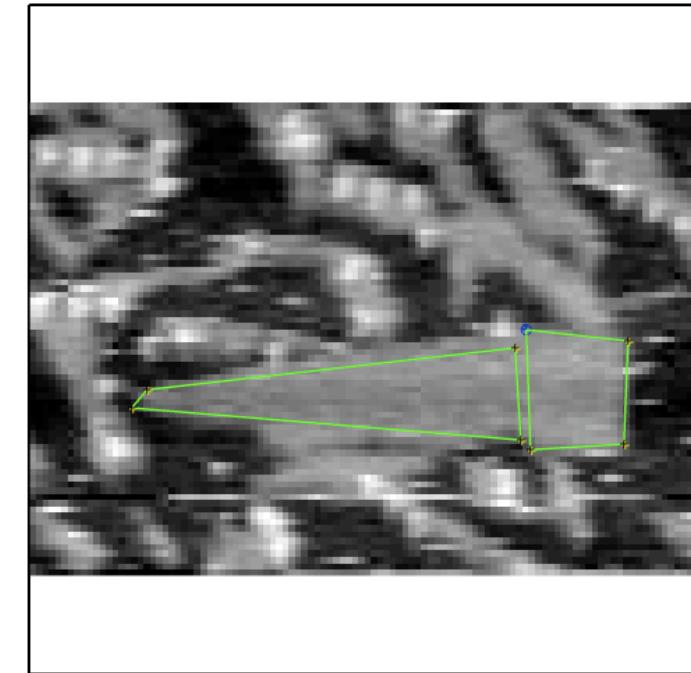
- Photo should be rotated 90 degrees left (counter-clockwise)
- Photo should be rotated 90 degrees right (clockwise)
- Photo should be turned upside down
- Photo is oriented properly

Please describe the picture in the box using 10 words or more:

shells

[Submit Turk](#) | [Skip / Load a different photo](#)

The submit button MUST be clicked!



Existing objects:

anvil_2

[UNDO](#)

[Done](#)

[Delete](#)

[Add anvil](#)

[Add ribbon](#)

Outline the second outcropping of the anvil (if available)

[Submit results](#)

Please select shape.



Who are the Turkers?

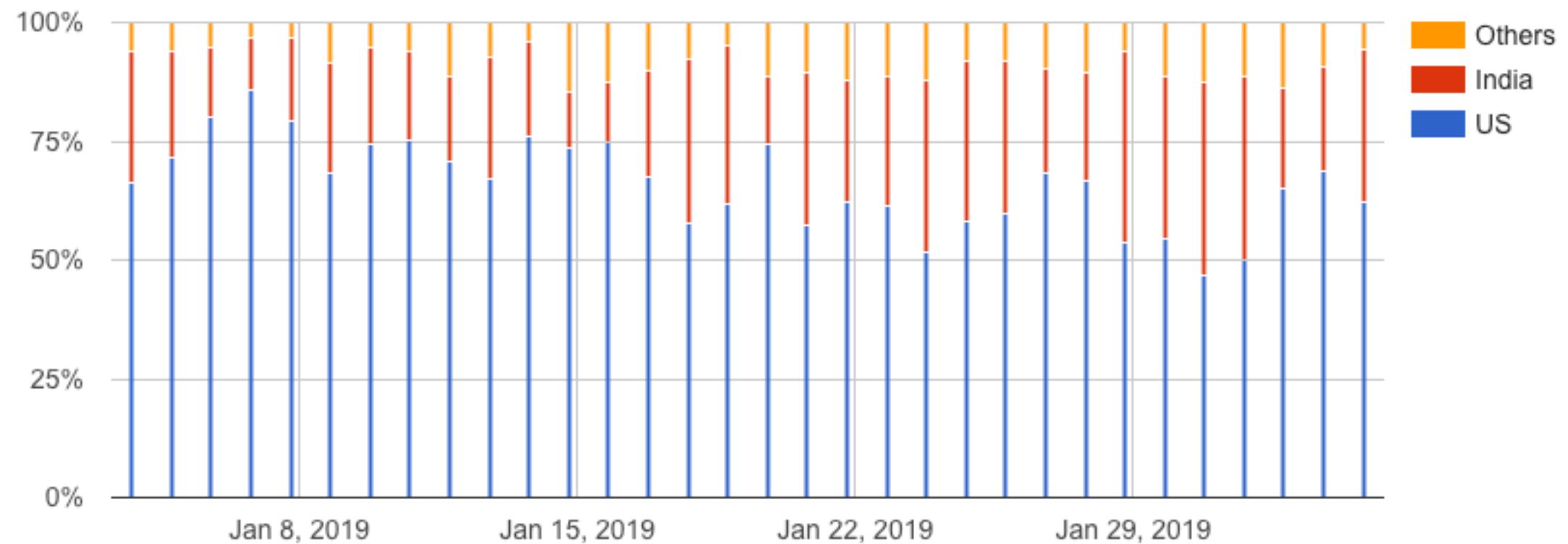
- Around 180K distinct workers (Difallah et al., 2018)
- About 10-20% of all workers do 80% of the work

https://waxy.org/2008/11/the_faces_of_mechanical_turk/



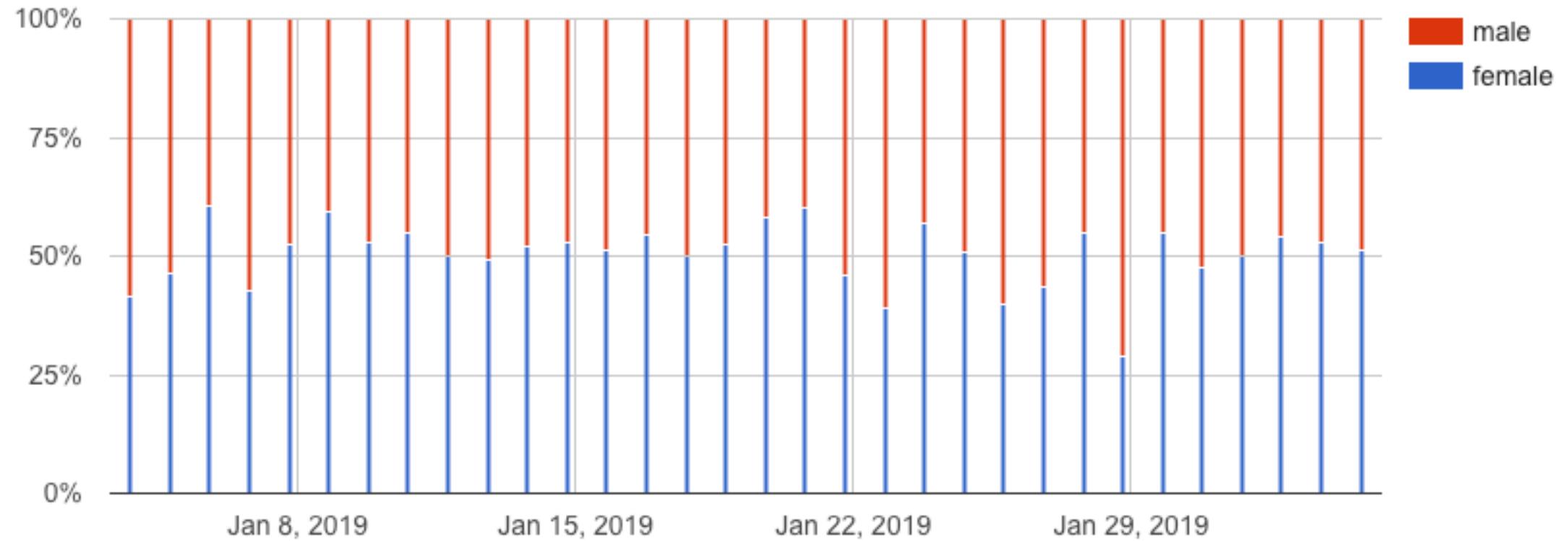
- Chandler, J., Mueller, P. A., & Paolacci, G. (2014). Nonnaïveté among Amazon Mechanical Turk workers: consequences and solutions for behavioral researchers. *Behavior Research Methods*, 46, 112–130.
- Difallah, Djellel, Elena Filatova, and Panos Ipeirotis. "Demographics and dynamics of mechanical turk workers." *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 2018.

Countries

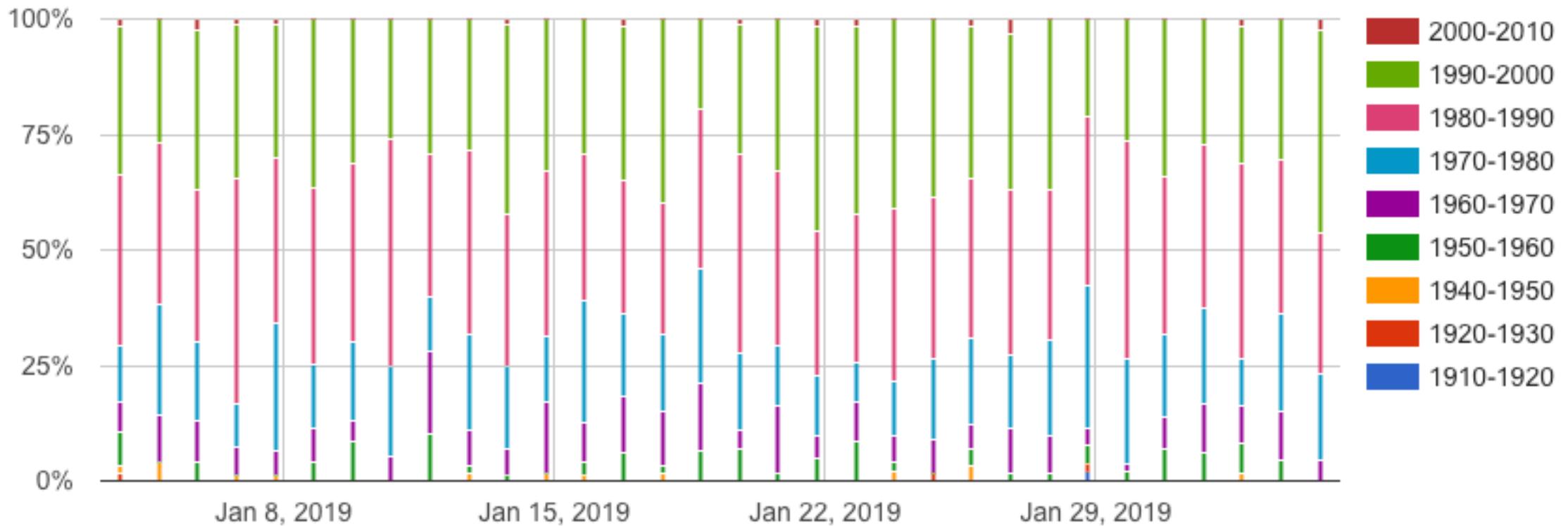


Analyzing the Amazon Mechanical Turk Marketplace, P. Ipeirotis, ACM XRDS, Vol 17, Issue 2, Winter 2010, pp 16-21.

Gender



Age



Good and bad tasks

- Easy cognitive task
 - Good: Where is the car? (bounding box)
 - Good: How many cars are there? (3)
 - Bad: How many cars are there? (132)
- Well-defined task
 - Good: Locate corners of the eyes.
 - Bad: Label joint locations. (low resolution or close-up images)
- Concise definition
 - Good: 1-2 paragraphs, fixed for all tasks
 - Good: 1-2 unique sentences per task.
 - Bad: 300 pages annotation manual
- Low amount of input
 - Good: few clicks or a couple words
 - Bad: detailed outlines of all objects (100s of control points)

<http://vision.cs.uiuc.edu/annotation/>



The Online Hell of Amazon's Mechanical Turk

https://www.theatlantic.com/business/archive/2018/01/amazon-mechanical-turk/551192/

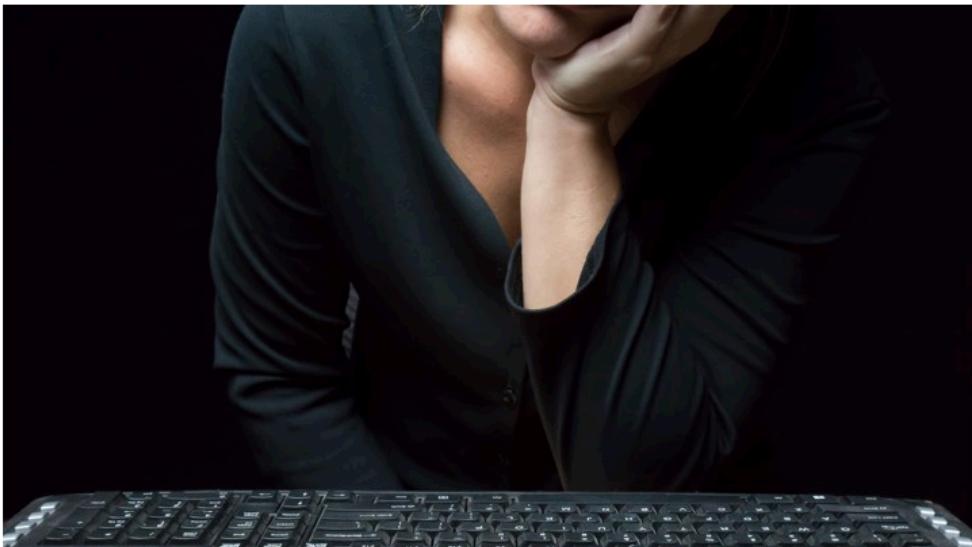
The Atlantic Popular Latest Sections Magazine More Subscribe

NEXT ECONOMY

The Internet Is Enabling a New Kind of Poorly Paid Hell

For some Americans, sub-minimum-wage online tasks are the only work available.

ALANA SEMUELS JAN 23, 2018



HARPAZO_HOPE / GETTY

MORE STORIES

A Small Town Kept Walmart Out. Now It Faces Amazon.
ALANA SEMUELS



This Is What Life Without Retirement Savings Looks Like
ALANA SEMUELS



Why Amazon Pays Some of Its Workers to Quit
ALANA SEMUELS



The Promise of Indoor, Hurricane-Proof 'Vertical' Farms
MEAGAN FLYNN



Technology has helped rid the American economy of many of the routine, physical,

How to be a good requester?

- Give your real identity
- Be available for workers
- Pay living wage
- Give context and be honest
- Allow for informed consent
- Don't get involved in wage theft
- Be careful when rejecting/blocking
- Keep Worker IDs anonymous



By Kristy Milland

Best practices

- Think about qualifications
 - Do not go below 98% qualifications
- Think about language and location
- Add quality assurance mechanisms

Worker requirements

Require that Workers be Masters to do your HITs ([Who are Mechanical Turk Masters?](#))

Yes No

Specify any additional qualifications Workers must meet to work on your HITs:
(Premium Qualifications incur additional fees, see [Pricing Details](#) to learn more)

-- Select -- Remove

System Qualifications

Location
HIT Approval Rate (%) for all Requesters' HITs
Number of HITs Approved

Premium Qualifications

Primary Mobile Device - iPhone
Primary Mobile Device - Android

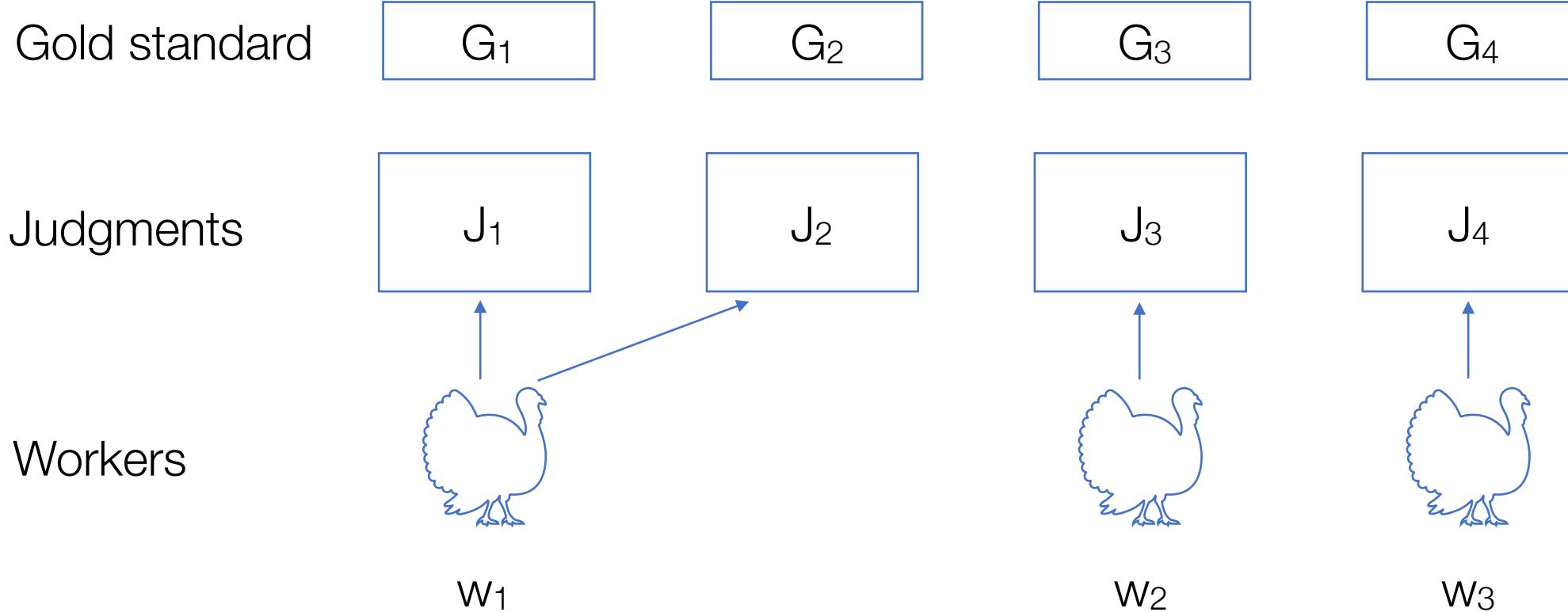
HIT Visibility ([What is HIT visibility?](#))

Public - All Workers can see and preview my HITs

Private - All Workers can see my HITs, but only Workers that meet all Qualification requirements can preview my HITs

Hidden - Only Workers that meet my HIT Qualification requirements can see and preview my HITs

Modeling judgments and quality



Defining quality

- Objective quality:
 - Whether judgments differ from a golden standard
- Consensus-based quality:
 - Inter-rater agreement: whether workers agree with each other

Definition 1: Distance from a gold standard

- Given a set of judgments ($J = j_1 \dots j_n$) about an object
- We assume that we have a gold standard: an oracle's decision ($G = G_1 \dots G_n$)
- The average distance is given by

$$\Delta(J, G) = \frac{\sum_{i=1}^n |g_i - j_i|}{n}$$

Cohen Kappa

- Cohen's kappa coefficient (Smeeton, 1985) is a simple statistic which measures inter-rater agreement for qualitative (categorical) items
- Each rater classify n items into C mutually exclusive categories
- p_o is the proportion of times that annotators agree and p_e is the proportion of times that agreement is expected by chance

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e}$$

Example

The data:

		B	
		Yes	No
A	Yes	a	b
	No	c	d

Raw data



		B	
		Yes	No
A	Yes	20	5
	No	10	15

Agreement table

To calculate p_e , we note that A says yes 25 times (50%) and B says yes 30 times (60%)

$$p_{\text{Yes}} = \frac{a + b}{a + b + c + d} \cdot \frac{a + c}{a + b + c + d} = 0.5 \times 0.6 = 0.3$$

$$p_{\text{No}} = \frac{c + d}{a + b + c + d} \cdot \frac{b + d}{a + b + c + d} = 0.5 \times 0.4 = 0.2$$

Calculating p_o - the relative observed agreement:

$$p_o = \frac{a + d}{a + b + c + d} = \frac{20 + 15}{50} = 0.7$$

Overall random agreement probability is the probability that they agreed on either Yes or No:

$$p_e = p_{\text{Yes}} + p_{\text{No}} = 0.3 + 0.2 = 0.5$$

$$\kappa = \frac{p_o - p_e}{1 - p_e} = \frac{0.7 - 0.5}{1 - 0.5} = 0.4$$

Methods for Improving quality

- Removing Low-Agreement Judges
- Removing Outlying Judgments
- Scaling Judgments

Denkowski, Michael, and Alon Lavie. "Exploring normalization techniques for human judgments of machine translation adequacy collected using Amazon Mechanical Turk." *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Association for Computational Linguistics, 2010.

Removing Low-Agreement Judges

- Calculate pairwise inter-annotator agreement (p_o) of each annotator with all others
- Removing judgments from annotators with p_o below some threshold
- The threshold can be set such that the highest overall agreement can be achieved while retaining at least one judgment for each translation

Removing Outlying Judgments

- For a given translation and human judgments ($j_1 \dots j_n$)
- Calculate the distance (δ) of each judgment from the mean (\bar{j}):

$$\delta(j_i) = |j_i - \bar{j}|$$

- We then remove outlying judgments with $\delta(j_i)$ exceeding some threshold.
- This threshold is also set such that the highest agreement is achieved while retaining at least one judgment per translation

Scaling Judgments

- To account for the notion that some annotators judge translations more harshly than others, apply per-annotator scaling to the adequacy judgments based on annotators' signed distance from gold standard judgments
- For judgments ($J = j_1 \dots j_n$) and gold standard ($G = g_1 \dots g_n$), an additive scaling factor is calculated:

$$\lambda_+(J, G) = \frac{\sum_{i=1}^n g_i - j_i}{n}$$

- Adding this scaling factor to each judgment has the effect of shifting the judgments' center of mass to match that of the gold standard

Summary

- Definitions of errors
- Removing Low-Agreement Judges
- Removing Outlying Judgments
- Scaling Judgments

Types of Data Sources

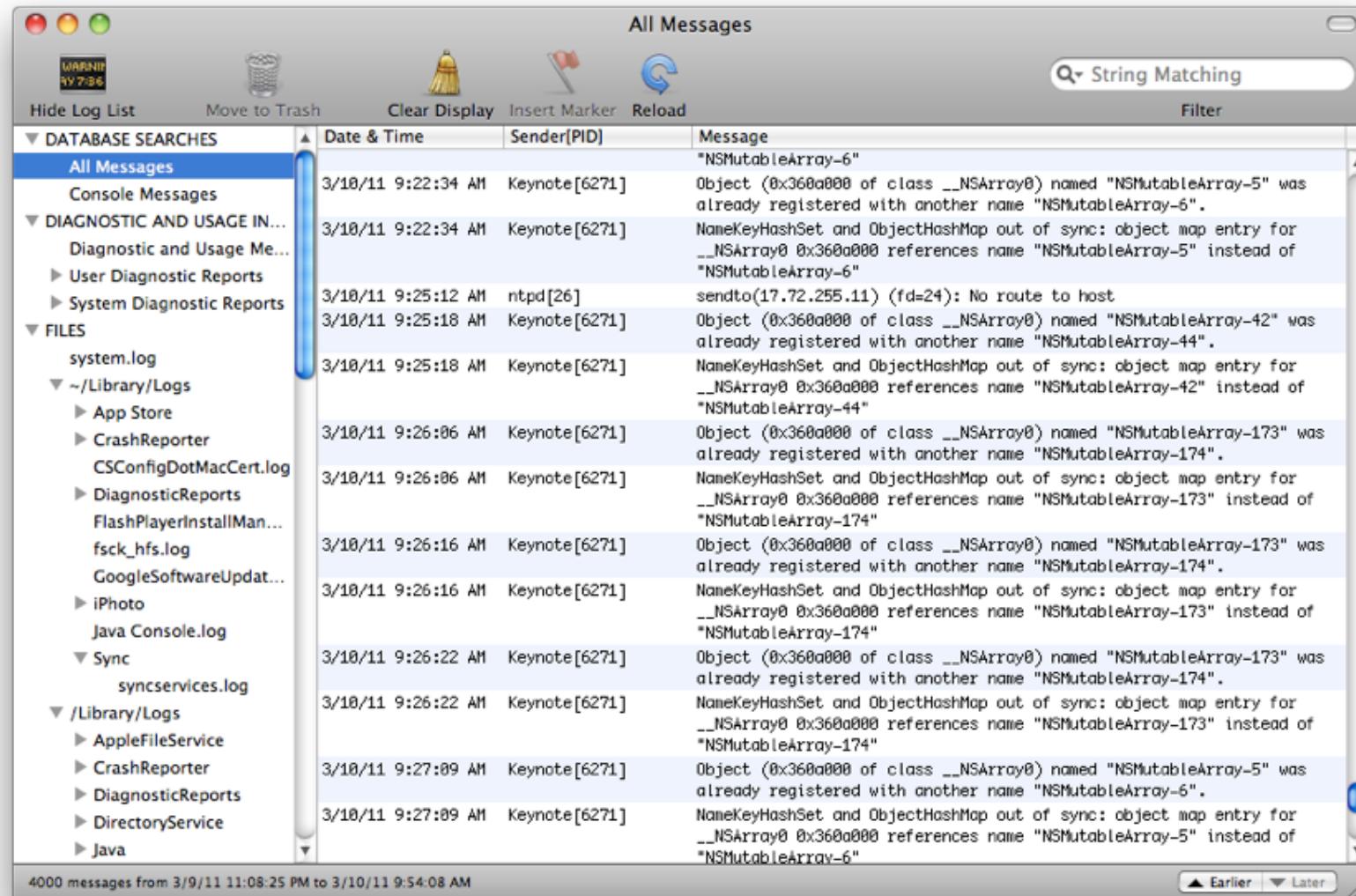
- Files
 - Flat files (csv...)
- Structured sources
 - Rational databases
 - XML / JSON

Data source 1: Flat files

- Flat files such as comma-separated values (CSV) files store numbers and text in plain text
- The CSV file format is not standardized, apart from commas between values and \n at the end of a record (and even those may change)

```
WI6nd1W1b1,_User$yx1fzkPK1D,2016-11-13T06:56:56.279Z,"[34.77328245,32.07458749]"
ZWrcA2NJeV,_User$R2wN32XXkE,2016-11-13T06:56:53.819Z,"[34.8134714,32.014789]"
F8uFlvaZuD,_User$Dc9xA04evy,2016-11-13T06:56:53.089Z,"[34.77381643,32.08176609]"
5afVZJaaui,_User$p5U4u5DXBx,2016-11-13T06:56:51.792Z,"[34.76782405913168,32.06603412054489]"
XV5KHZ4duz,_User$VOCydAgn51,2016-11-13T06:56:48.520Z,"[34.863347632312156,32.19136579571034]"
76B5M2E6Ul,_User$8LQLe63Jqq,2016-11-13T06:56:43.438Z,"[35.44087488,32.98058869]"
mvrIILpB83R,_User$wB5KVtfNEp,2016-11-13T06:56:19.242Z,"[34.78664151,31.42228791]"
CGc6r2cyl2,_User$Ealybaxr2A,2016-11-13T06:56:18.758Z,"[34.80443977,32.0269589]"
w26YPSJYks,_User$rfYUev7pD2,2016-11-13T06:56:16.431Z,"[34.7823733,32.0577361]"
```

Logs



The screenshot shows the OS X Console application window titled "All Messages". The left sidebar lists various log sources under categories like "DATABASE SEARCHES", "DIAGNOSTIC AND USAGE IN...", and "FILES". The main pane displays a table of log entries with columns for Date & Time, Sender(PID), and Message. The "Message" column contains repeated errors about mutable arrays. A blue vertical bar highlights the first error entry.

	Date & Time	Sender(PID)	Message
All Messages	3/10/11 9:22:34 AM	Keynote[6271]	"NSMutableArray-6" Object (0x360a000 of class __NSArray0) named "NSMutableArray-5" was already registered with another name "NSMutableArray-6".
Console Messages	3/10/11 9:22:34 AM	Keynote[6271]	NameKeyHashSet and ObjectHashMap out of sync: object map entry for __NSArray0 0x360a000 references name "NSMutableArray-5" instead of "NSMutableArray-6"
Diagnostic and Usage Me...	3/10/11 9:25:12 AM	ntpd[26]	sendto(17.72.255.11) (fd=24): No route to host
Diagnostic and Usage Me...	3/10/11 9:25:18 AM	Keynote[6271]	Object (0x360a000 of class __NSArray0) named "NSMutableArray-42" was already registered with another name "NSMutableArray-44".
User Diagnostic Reports	3/10/11 9:25:18 AM	Keynote[6271]	NameKeyHashSet and ObjectHashMap out of sync: object map entry for __NSArray0 0x360a000 references name "NSMutableArray-42" instead of "NSMutableArray-44"
System Diagnostic Reports	3/10/11 9:26:06 AM	Keynote[6271]	Object (0x360a000 of class __NSArray0) named "NSMutableArray-173" was already registered with another name "NSMutableArray-174".
system.log	3/10/11 9:26:06 AM	Keynote[6271]	NameKeyHashSet and ObjectHashMap out of sync: object map entry for __NSArray0 0x360a000 references name "NSMutableArray-173" instead of "NSMutableArray-174"
~/Library/Logs	3/10/11 9:26:16 AM	Keynote[6271]	Object (0x360a000 of class __NSArray0) named "NSMutableArray-173" was already registered with another name "NSMutableArray-174".
App Store	3/10/11 9:26:16 AM	Keynote[6271]	NameKeyHashSet and ObjectHashMap out of sync: object map entry for __NSArray0 0x360a000 references name "NSMutableArray-173" instead of "NSMutableArray-174"
CrashReporter	3/10/11 9:26:16 AM	Keynote[6271]	Object (0x360a000 of class __NSArray0) named "NSMutableArray-173" was already registered with another name "NSMutableArray-174".
CSConfigDotMacCert.log	3/10/11 9:26:16 AM	Keynote[6271]	NameKeyHashSet and ObjectHashMap out of sync: object map entry for __NSArray0 0x360a000 references name "NSMutableArray-173" instead of "NSMutableArray-174"
DiagnosticReports	3/10/11 9:26:16 AM	Keynote[6271]	Object (0x360a000 of class __NSArray0) named "NSMutableArray-173" was already registered with another name "NSMutableArray-174".
FlashPlayerInstallMan...	3/10/11 9:26:16 AM	Keynote[6271]	NameKeyHashSet and ObjectHashMap out of sync: object map entry for __NSArray0 0x360a000 references name "NSMutableArray-173" instead of "NSMutableArray-174"
fsck_hfs.log	3/10/11 9:26:16 AM	Keynote[6271]	Object (0x360a000 of class __NSArray0) named "NSMutableArray-173" was already registered with another name "NSMutableArray-174".
GoogleSoftwareUpdat...	3/10/11 9:26:16 AM	Keynote[6271]	NameKeyHashSet and ObjectHashMap out of sync: object map entry for __NSArray0 0x360a000 references name "NSMutableArray-173" instead of "NSMutableArray-174"
iPhoto	3/10/11 9:26:22 AM	Keynote[6271]	Object (0x360a000 of class __NSArray0) named "NSMutableArray-173" was already registered with another name "NSMutableArray-174".
Java Console.log	3/10/11 9:26:22 AM	Keynote[6271]	NameKeyHashSet and ObjectHashMap out of sync: object map entry for __NSArray0 0x360a000 references name "NSMutableArray-173" instead of "NSMutableArray-174"
Sync	3/10/11 9:27:09 AM	Keynote[6271]	Object (0x360a000 of class __NSArray0) named "NSMutableArray-5" was already registered with another name "NSMutableArray-6".
syncservices.log	3/10/11 9:27:09 AM	Keynote[6271]	NameKeyHashSet and ObjectHashMap out of sync: object map entry for __NSArray0 0x360a000 references name "NSMutableArray-5" instead of "NSMutableArray-6"
/Library/Logs	3/10/11 9:27:09 AM	Keynote[6271]	
AppleFileService			
CrashReporter			
DiagnosticReports			
DirectoryService			
Java			

4000 messages from 3/9/11 11:08:25 PM to 3/10/11 9:54:08 AM

Earlier Later

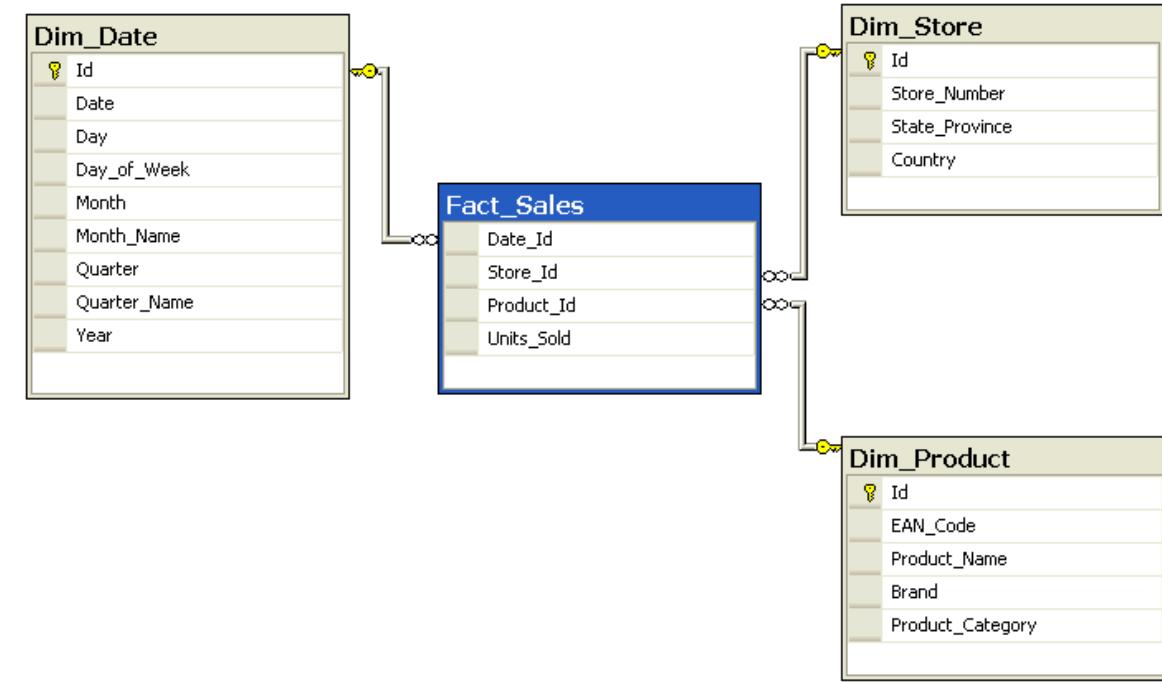


Characteristics

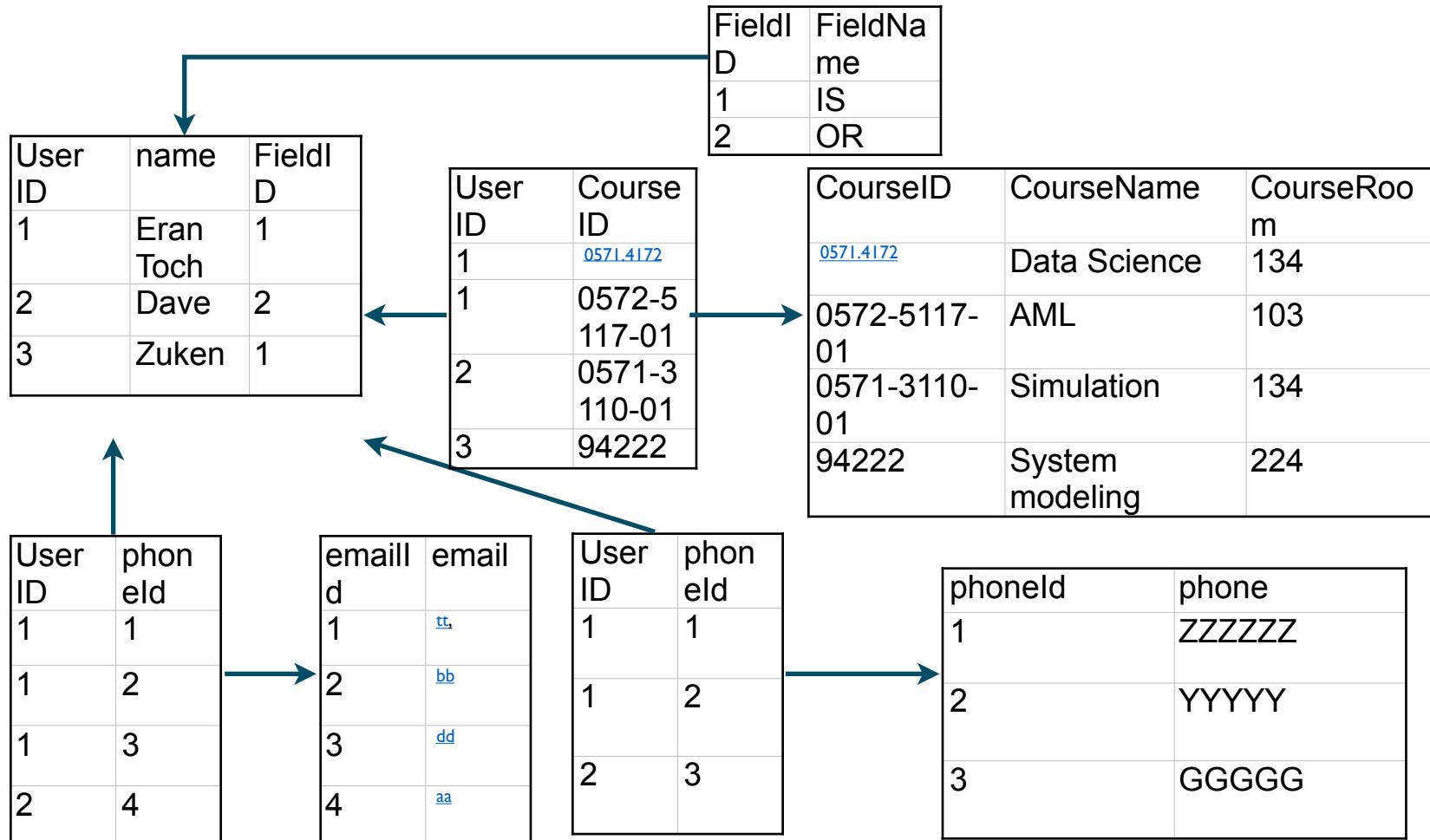
- Strong points
 - Simple (one file) structure
 - Timed data (in many cases)
- Weak points
 - No schema
 - No semantics

Data Source 2: Relational Databases

- Relational databases organize data into one or more tables (or "relations") of columns and rows
- Each row in a table has its own unique key
- Rows can be linked using foreign keys
- Inserting data or querying it requires to check the constraints of the schema, and in most cases using a standard language (SQL - structured query language)



Data Source 2: Relational Databases



Characteristics

- Strong points
 - Standard interface
 - Predictable structure
 - Schema is consistent and static
- Weak points
 - Normalized
 - Performance is muddy with joins
 - Might be non-timed

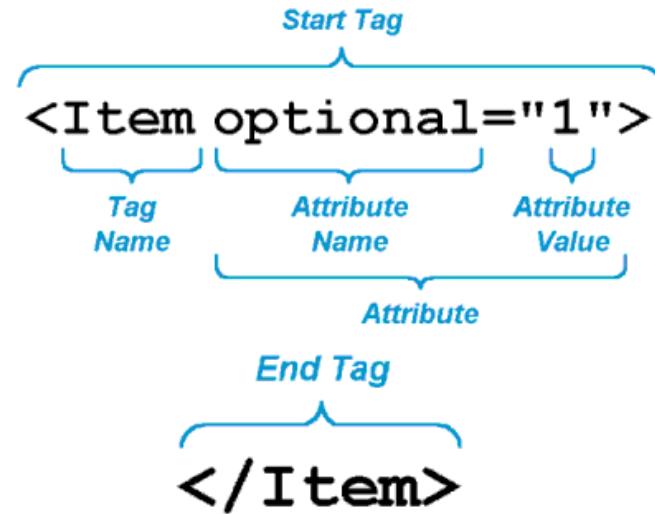
Data source 3: XML files

- XML stands for **E**xensible **M**arkup **L**anguage
- It is a text-based markup language derived from Standard Generalized Markup Language (SGML)
- XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags
- XML allows to create self-descriptive tags, or language

```
<note>
  <to>InfoSys</to>
  <from>Eran</from>
  <heading>Reminder</
heading>
  <body>Don't forget the
HW</body>
</note>
```

XML Elements

- XML files are made of tags
- Each tag may include a list of attributes
 - text
 - attributes
 - other elements
- The Item defined by the tag ends with the end tag
- The XML file is defined with the header:
`<?xml version="1.0"
encoding="UTF-8"?>`



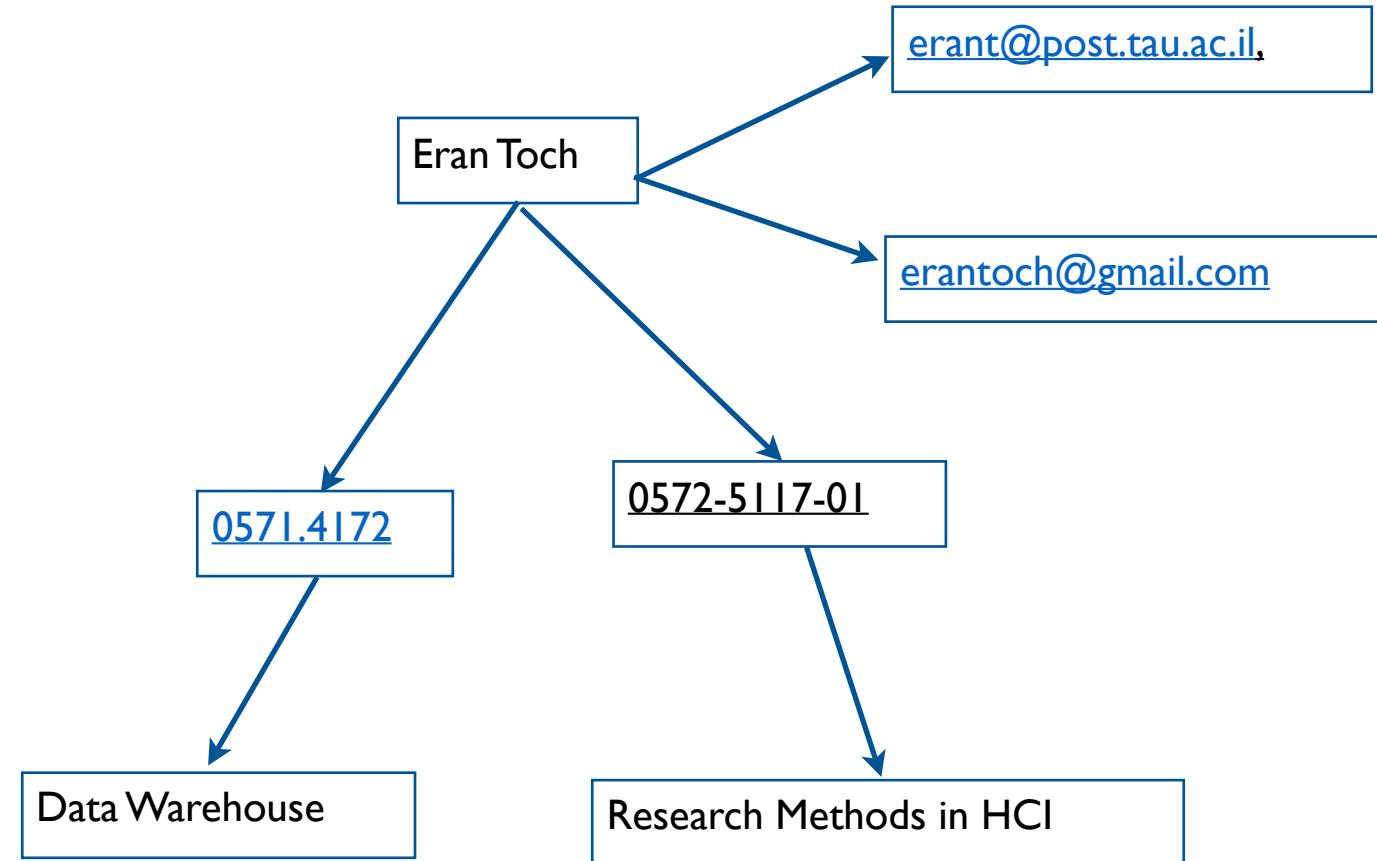
XML Tree Structure

XML documents form a tree structure that starts at the root and branches to the leaves

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

XML as Structured Schema Database

- Mainly graph-based
- Standard libraries to read and write to files



Syntax Rules

- No unclosed tags
 - An empty tag is defined with <item/>
- No overlapping tags
 - <Tomato>Let's call <Potato>the whole thing off</Tomato></Potato>
- Attribute values must be enclosed in quotes (<TABLE BORDER="1">)
- XML Tags are Case Sensitive
- <!-- This is a -- comment -->

Characteristics

- Strong points
 - Standard interface
 - Tree structure (fast joins)
 - Well-explained semantic structure
- Weak points
 - Weak keys
 - References
 - Non-timed

Data Source 4: JSON - JavaScript Object Notation

- JSON is a lightweight data-interchange format
- It provides most of the features of XML, but with less overhead
- Native to JavaScript

```
{  
  "book": [  
    {  
      "id": "01",  
      "language": "English",  
      "title": "Harry Potter",  
      "author": "J K. Rowling"  
    },  
    {  
      "id": "07",  
      "language": "English",  
      "title": "Harry Potter 2",  
      "author": "J K. Rowling"  
    }  
  ]  
}
```

JSON Objects

- An unordered set of name/value pairs
- Objects are enclosed in curly braces
that is, it starts with '{' and ends with '}'
- Each name is followed by ':'(colon)
and the key/value pairs are separated
by , (comma)
- The keys must be strings and should
be different from each other.

```
{  
  "id": "1234",  
  "language": "English",  
  "price": 500,  
}
```

JSON Values

- JSON Values can include:
 - number (integer or floating point)
 - string
 - boolean
 - array
 - object
 - null

```
var i = 1;  
var j = "harry";  
var k = null;  
var l = true;
```

JSON Arrays

- Arrays are an ordered collection of values
- These are enclosed in square brackets which means that array begins with [and ends with]

```
{  
  "books": [  
    { "language": "Java" , "edition": "second" },  
    { "language": "C++" , "lastName": "fifth" },  
    { "language": "C" , "lastName": "third" }  
  ]  
}
```

Characteristics

- Strong points
 - Minimal overhead
 - Standard interface
 - Tree structure (fast joins)
 - Well-explained semantic structure
- Weak points
 - Weak keys
 - References
 - Non-timed
 - Hard to read manually

Summary

- Flat files (csv...)
- Relational databases
- Tree-based sources
 - Rational databases
 - XML / JSON

Summary