What is Data Engineering
Where Does Data Come From & Tools for Data Engineering

Data Engineering

What is Data Engineering?

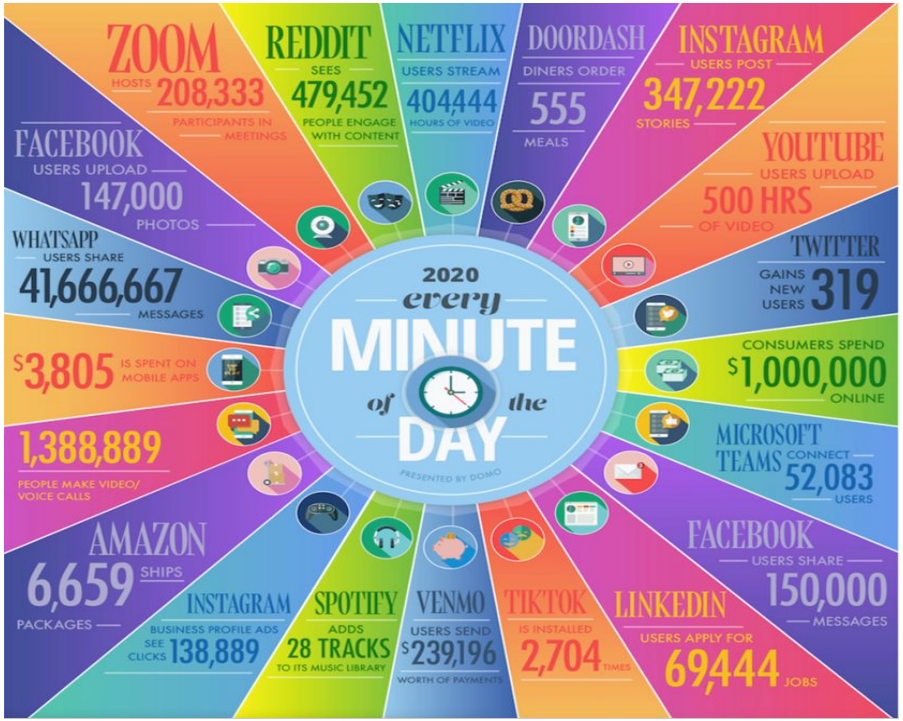
- “Data” engineers design and build pipelines that transform and transport data into a format wherein, by the time it reaches the Data Scientists or other end users, it is in a highly usable state. ref
- data engineers are concerned with the production readiness of that data and all that comes with it: formats, scaling, resilience, security, and more. ref
- they build pipelines that transform that data into formats that data scientists can use. ref

What is Data Engineering?

- “Data” engineers design and build pipelines that transform and transport data into a format wherein, by the time other end users, it
- data engi uction readiness with it: formats, d more. ref
- they build pipelines that transform that data into formats that data scientists can use. ref

General definition - Building pipelines to make data useful for data scientists and analysts

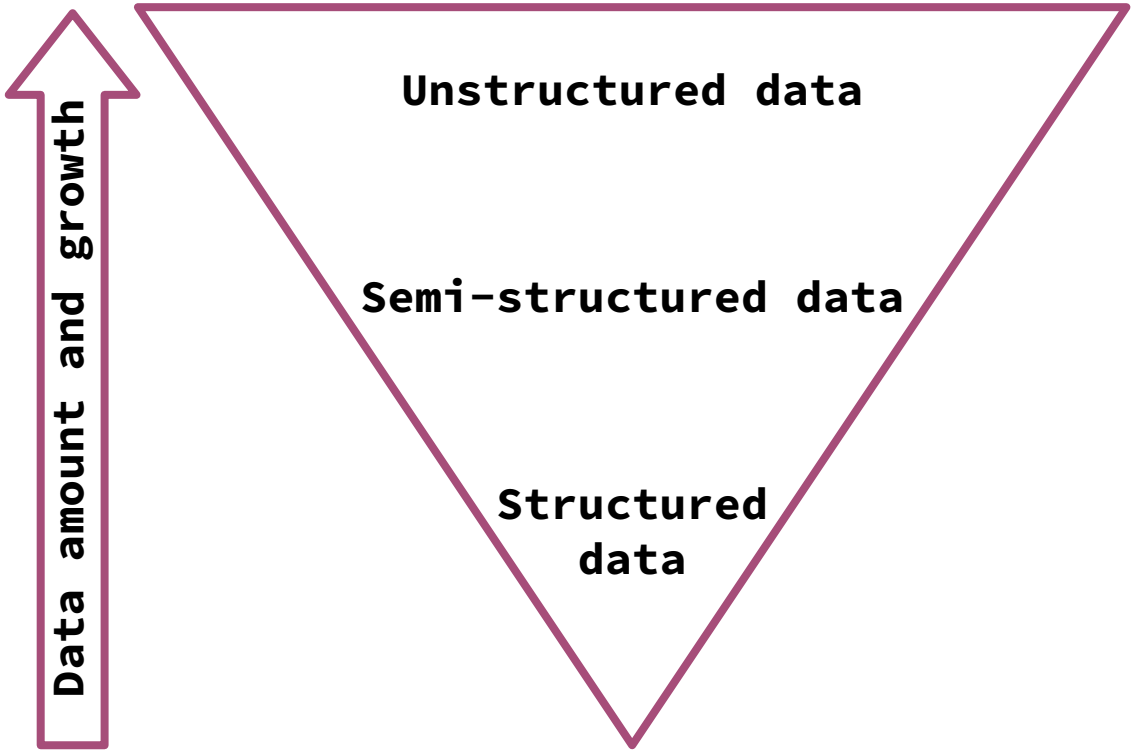
Every minute on the Internet



Source: domo.com

Pyramid of data organization

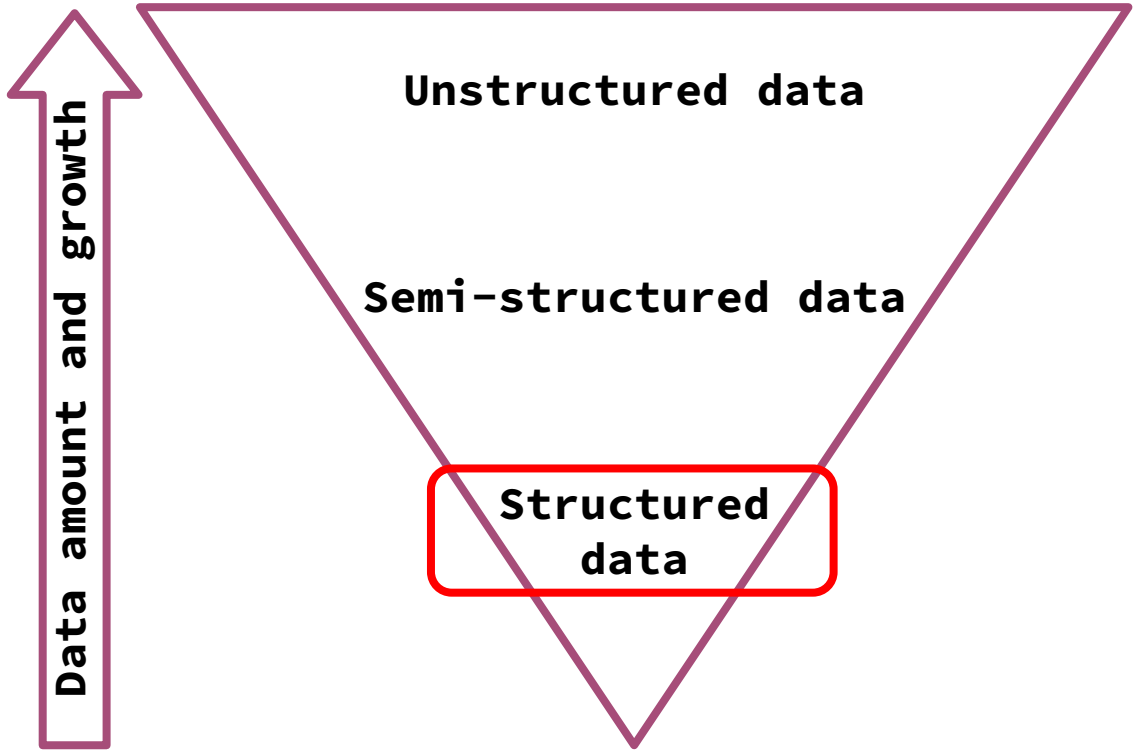
- Better to think of structure as being continuous
- Less and less data is **Structured**
- More and more is **Unstructured**
- Still room in between



Pyramid of data organization

Structured data

- Fits into a database nicely
- ‘square’ or ‘cube’ formats
- Don’t need to do work to analyze or query



Pyramid of data organization

Structured data

- SQL DB with two tables
- Simple query to get total sales for AZ stores
- No data processing needed to do this

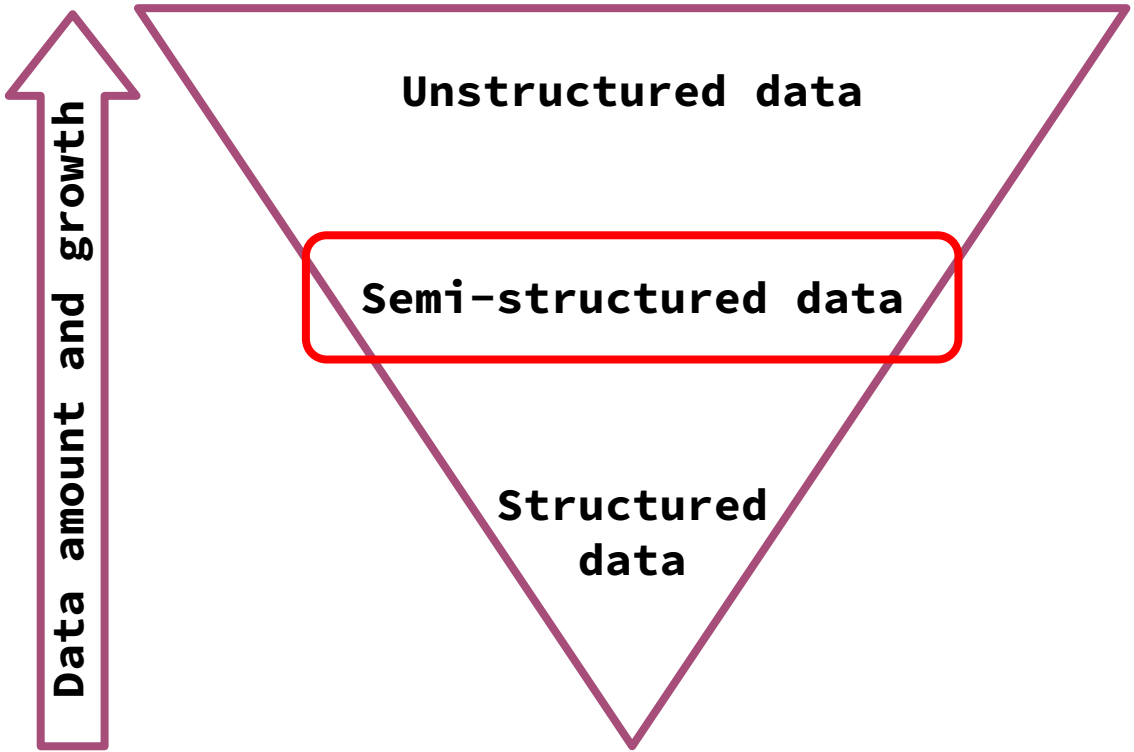
TABLE ID: STORE		
store_id	**store_state**	**country**
il_23	IL	USA
az_45	AZ	USA
ca_12	CA	USA
to_39	Ontario	Canada

TABLE ID: TRANSACTIONS			
transact_id	**store_id**	**UPC**	**price**
x88943	il_23	49914	2.57
x88943	il_23	99371	1.99
a85921	to_39	95831	8.99
a85921	to_39	99492	5.49
a85921	to_39	27482	4.49
z88930	az_45	33491	0.99

Pyramid of data organization

Semi-structured data

- JSON, XML, csv, tsv
- Has some consistent format
- Minimal work to get into useable format



Pyramid of data organization

Semi-structured data

- Excel, csv, tsv
- May need to clean
- Need to join a bunch
- Aggregate
- May take time, but relatively simple

Sample AirBNB data from CSV file

A	B	C	D	E	F	G	H	I	J
id	name	host_id	host_name	neighbourhood_g	neighbourhood	latitude	longitude	room_type	price
2539	Clean & quiet apt	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149
2595	Skylit Midtown Ca	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225
3647	THE VILLAGE O	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.9419	Private room	150
3831	Cozy Entire Floor	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89
5022	Entire Apt: Spaci	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80
5099	Large Cozy 1 BR	7322	Chris	Manhattan	Murray Hill	40.74767	-73.975	Entire home/apt	200
5121	BlissArtsSpace!	7356	Garon	Brooklyn	Bedford-Stuyvesa	40.68688	-73.95596	Private room	60
5178	Large Furnished	8967	Shunichi	Manhattan	Hell's Kitchen	40.76489	-73.98493	Private room	79
5203	Cozy Clean Gues	7490	MaryEllen	Manhattan	Upper West Side	40.80178	-73.96723	Private room	79
5238	Cute & Cozy Low	7549	Ben	Manhattan	Chinatown	40.71344	-73.99037	Entire home/apt	150
5295	Beautiful 1br on U	7702	Lena	Manhattan	Upper West Side	40.80316	-73.96545	Entire home/apt	135
5441	Central Manhatta	7989	Kate	Manhattan	Hell's Kitchen	40.76076	-73.98867	Private room	85
5803	Lovely Room 1, C	9744	Laurie	Brooklyn	South Slope	40.66829	-73.98779	Private room	89
6021	Wonderful Guest	11528	Claudio	Manhattan	Upper West Side	40.79826	-73.96113	Private room	85
6090	West Village Nes	11975	Alina	Manhattan	West Village	40.7353	-74.00525	Entire home/apt	120
6848	Only 2 stops to M	15991	Allen & Irina	Brooklyn	Williamsburg	40.70837	-73.95352	Entire home/apt	140
7097	Perfect for Your P	17571	Jane	Brooklyn	Fort Greene	40.69169	-73.97185	Entire home/apt	215
7322	Chelsea Perfect	18946	Doti	Manhattan	Chelsea	40.74192	-73.99501	Private room	140
7726	Hip Historic Brow	20950	Adam And Charil	Brooklyn	Crown Heights	40.67592	-73.94694	Entire home/apt	99
7750	Huge 2 BR Upper	17985	Sing	Manhattan	East Harlem	40.79685	-73.94872	Entire home/apt	190
7801	Sweet and Spaci	21207	Chaya	Brooklyn	Williamsburg	40.71842	-73.95718	Entire home/apt	299
8024	CBG CtyBGd He	22486	Lisel	Brooklyn	Park Slope	40.68069	-73.97706	Private room	130
8025	CBG Helps Haiti	22486	Lisel	Brooklyn	Park Slope	40.67989	-73.97798	Private room	80
8110	CBG Helps Haiti	22486	Lisel	Brooklyn	Park Slope	40.68001	-73.97865	Private room	110

Pyramid of data organization

Semi-structured data

- JSON, xml
- Has an overall schema/organization
- Need to parse and organize to make useable
- May take time, but relatively simple

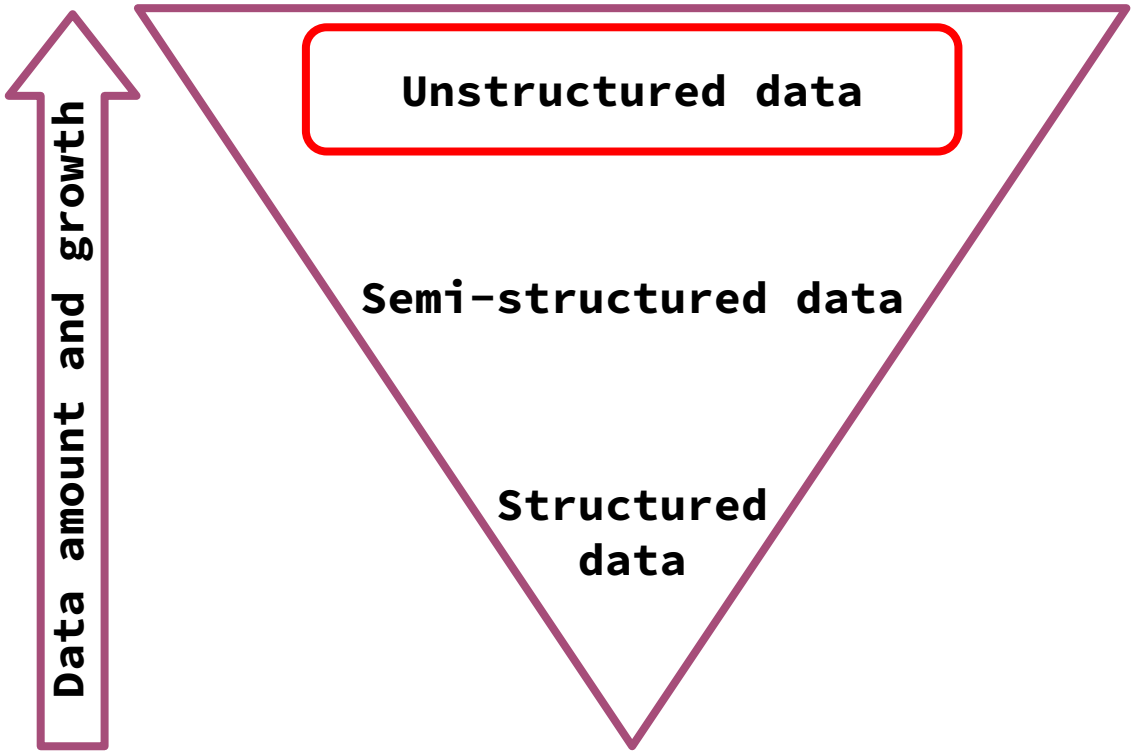
Sample Twitter data in JSON format


```
{
    "contributors": null,
    "coordinates": null,
    "created_at": "Fri Jun 28 07:31:35 +0000 2019",
    "display_text_range": [
        0,
        1
    ],
    "entities": {
        "hashtags": [],
        "symbols": [],
        "urls": [
            {
                "display_url": "twitter.com/polhomeeditor/\u2026",
                "expanded_url": "https://twitter.com/polhomeeditor/status/1144289510739587073",
                "indices": [
                    2,
                    25
                ],
                "url": "https://t.co/0fgkUFjCaB"
            }
        ],
        "user_mentions": []
    },
    "favorite_count": 3,
    "favorited": false,
    "full_text": "? https://t.co/0fgkUFjCaB",
    "geo": null,
    "id": 1144508626738044929,
    "id_str": "1144508626738044929",
```

Pyramid of data organization

Unstructured data

- Text, pdf, images, video
- Zero structure
- Lots of work to extract useful data from



Pyramid of data organization

Unstructured data

- Raw text - tweets, facebook, reviews
- How do you get something useful?
- Lots of processing
- Need to understand language
- Specific to use case

Sample of Twitter text

```
## [1] "If you are feeling the impacts of climate change in your daily
life, you're not alone. What do you want to protect from #climatechange?
Tell us below. #AdaptOurWorld https://t.co/RFyNFRLHGB"
## [2] "Tree 🌲 planting is great but it cannot become a PR machine for
fake climate initiatives by the governments. We will not push
#climatechange back by planting trees. Governments must listen and act.
Real-meaningful actions. Not PR!"
## [3] "Cigarette smokers &amp;amp; vapers are exhaling additional CO2 into
the air. I call on all politicians to stop smoking and ban cigarettes and
vaping for the earth. If not, then shut up about #climatechange and
CO2.\n\nLet’s see how much politicians care about the things they talk
about. https://t.co/0v60pNd38q"
## [4] "@ThemeParkReview @Starbucks Make your own coffee. You can even
buy the Starbucks brand in a grocery store, if it’s that important. This
really is a ridiculous tantrum over something with many solutions. There
are better things to worry about like the #ClimateChange that keeps
causing these hurricanes."
## [5] "@GaryCMeleJr @DebraMessing The two party system isn’t working
for the people &amp;amp; #Democrats need to do better because my independent
vote is on loan to them. But it’s the #GOP breaking constitutional norms,
refusing to protect our elections, shoving church into state, denying
#ClimateChange, caging kids etc"
## [6] "Why are #hurricanes getting bigger and moving slower?
🌪️\n\n#HurricaneDorian \n#ClimateChange #ClimateChangeIsReal
\n#ThereIsNoPlanetB 🌎 https://t.co/2z1JlpVnUg"
```

Pyramid of data organization

Unstructured data

- pdf files
- Very long format
- Might want to synthesize 1000's of medical papers to determine effect
- Formats vary across journals

Sample of pdf

2019 IEEE 5th International Conference on Big Data Intelligence and Computing (DATACOM)

Distributed-Memory Vertex-Centric Network Embedding for Large-Scale Graphs*

Sara Riazi
Department of Computer and Information Science
University of Oregon
Eugene, OR 97403, USA
riazi@uoregon.edu

Boyana Norris
Department of Computer and Information Science
University of Oregon
Eugene, OR 97403, USA
bnorris2@uoregon.edu

***Abstract*—Network embedding is an important step in many different computations based on graph data. However, existing approaches are limited to small or middle size graphs with fewer than a million edges. In practice, web or social network graphs are orders of magnitude larger, thus making most current methods impractical for very large graphs. To address this problem, we introduce a new distributed-memory parallel network embedding method based on Apache Spark and GraphX. We demonstrate the scalability of our method as well as its ability to generate meaningful embeddings for vertex classification and link prediction on both real-world and synthetic graphs.**

I. Introduction

Network embedding is an important step in solving many graph problems including link prediction, vertex classification, and clustering. Network embedding aims to learn a low dimensional vector representation for vertices of a graph. However, existing approaches do not scale to very large graphs with billions of vertices and edges. One solution is to use distributed-memory systems and out-of-core computation.

Among distributed-memory systems, frameworks such as the Apache Spark-based GraphX [1] are of particular interest to us because they offer a map-reduce-based approach to expressing parallel algorithms for graph computations.

In order to take advantage of such distributed graph processing frameworks, we need to design new map-reduce [2] network embedding algorithms. In general, following the previous work for learning general network embedding [3], [4], [5], we use the structural properties of a network to train an embedding. A common assumption underlying existing methods and our new algorithm is that we expect that the embedding of a vertex is more similar to the embeddings of its neighbors rather than to the embedding of a random vertex outside of its neighborhood. We enforce this objective with approximate maximum likelihood training of the embedding in which the partition function is approximated using negative samples. This training requires lookup access to the embedding of vertices in a neighborhood, as well as vertices that lie outside of the neighborhood. However, lookup access in map-reduce frameworks is prohibitively expensive, which necessitates careful consideration in developing map-reduce based network embedding algorithms. In this paper, we introduce such an algorithm, and experimentally show that we can train network embeddings for very large graphs. We evaluate the new algorithm's accuracy and parallel scalability on a set of real-world networks.

Our **key contributions** include the following.

- A discussion of the limitations of GraphX for implementing existing network embedding algorithms.
- A new map-reduce-friendly message propagation model for learning vertex-centric network embeddings, which propagates the gradients instead of the embedding.
- The use of random graphs to construct negative sampling, which is necessary for approximate maximum likelihood training.
- A new GraphX based vertex-centric network embedding (VCNE) algorithm based on gradient propagation and random graphs that performs well on a range of real-world problems and synthetic graphs and can be applied to large problems that cannot be handled by current embedding approaches.

II. Parallel Graph Frameworks

Applying traditional graph algorithms to extremely large graphs requires distributed processing as well as out-of-core computation. Therefore, several parallel graph frameworks such as GraphX [1] and Giraph [6] have been developed on top of data-parallel systems, such as Apache Spark and Hadoop, respectively. As a result, they provide graph processing APIs using distributed data processing models such as map-reduce [2]. In map-reduce, data is converted to key-value pairs and then partitioned onto nodes. A map-reduce system consists of a set of workers that are coordinated by a master process. The master process assigns partitions to workers, and

This work was supported by the NSF CCF Award #1725585.

978-1-7281-4117-6/19/$31.00 ©2019 IEEE

DOI 10.1109/DataCom.2019.00029

Pyramid of data organization

Unstructured data

- Images
- Obviously we can determine elements in a picture
- Lots of work to get a computer to figure that out
- What items are in it
  - Ventilation Controls, steering wheel, Gas Pedal, Side Mirror, Road, Trees, etc.

Why is this talk about data structure relevant?

- Let's think about what a data analyst/scientist does
- **Data Analyst:** Processes and analyzes data with the goal of supporting decision making.
- **Data Science:** Extracting useful information from data using computational methods.
- Rough definitions, but both are using data to create understanding. Analysts do more whole data aggregation, summaries, basic stats. Data Scientists often delve deeper into stats and machine learning.
- Either way, let's consider the structure of a common model used by both.

Extracting info from data

- Let’s say you want to predict if someone shopping on amazon will buy a iPhone
- This is a common classification problem.
  - Target - Will they purchase
  - Features - Age, income level, browser, searched for iPhone, etc
- This is an easy model to fit in R/Python and created a prediction with.

```
P(buy_iPhone) ~ age + income_level + browser + search_iPhone
```

- But, what format do the data need to be in to run this model?

Extracting info from data

P(buy_iPhone) ~ age + income_level + browser + search_iPhone + apple_reviews + apple_sent

buy	age	income	browser	search_iPhone	apple_revs	apple_sentiment
yes	32	123000	safari	yes	3	4
no	56	56000	chrome	no	0	0
no	47	75000	firefox	no	1	2
yes	21	36000	safari	yes	5	5

- Not hard to get data in this format through SQL queries
- But what if these are scattered across messy databases
- Or say information is in JSON files
- Or you want features that are from unstructured data sources
- Or your data has lots of errors

Data science promised big things using these methods

- Predict things - supervised
  - Click an ad, is a purchase fraud, who you should friend, driving time, shipping time, what posts you should see, etc etc etc
- Uncover hidden insights - unsupervised
  - Customer segmentation, anomaly detection, market basket, etc
- Essentially, all things that would (and very much do) make money. Lots of money.
- Very cool models and tools to do all this
  - Regression, logistic regression, SVM, XGBoost, Decision Trees, naive Bayes, knn, k-means, dbscan, hierarchical clustering, PCA, t-SNE, etc
  - Tableau, PowerBI, Looker, Qlikview

Pyramid of needs

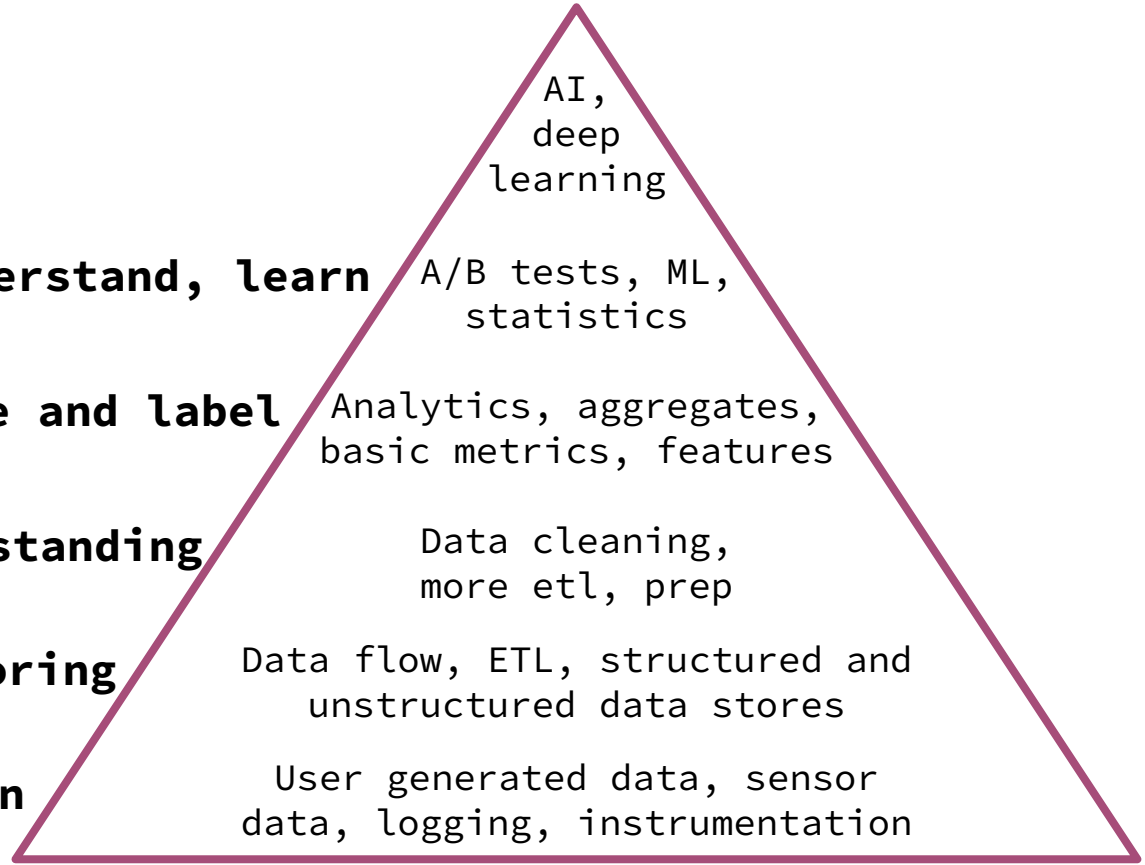


Fig adapted from - Ref

Pyramid of needs

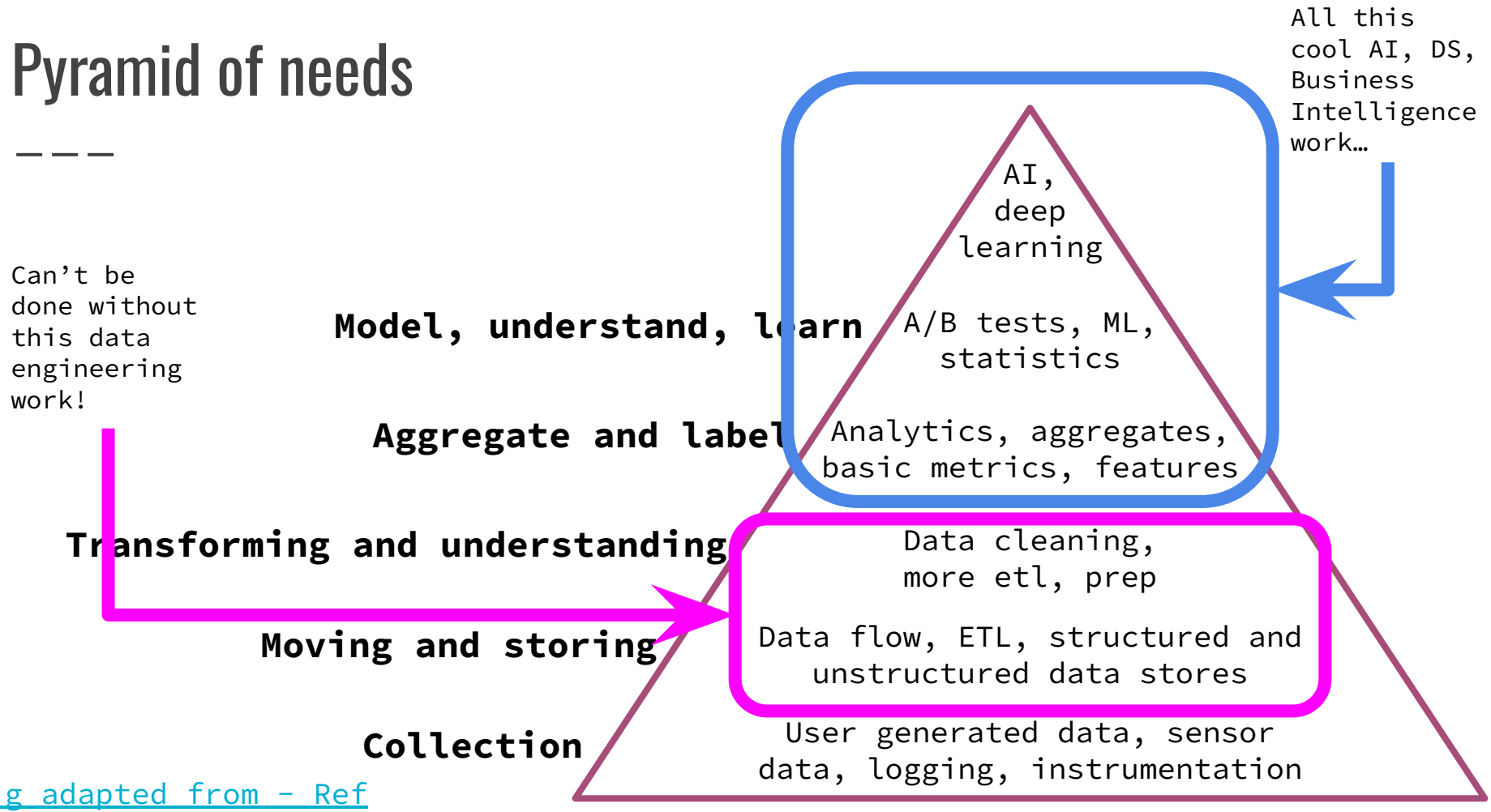


Fig adapted from - Ref

Data engineering is foundational to data science

- Data engineers take the raw data that's stored and coming in from all over the place and transform it to a useful format
- Data scientists take this data and make inference from it
- Ideally they work hand and hand
  - DS knows what the models need, DE knows how to get those data
- Industry hiring is following the pyramid

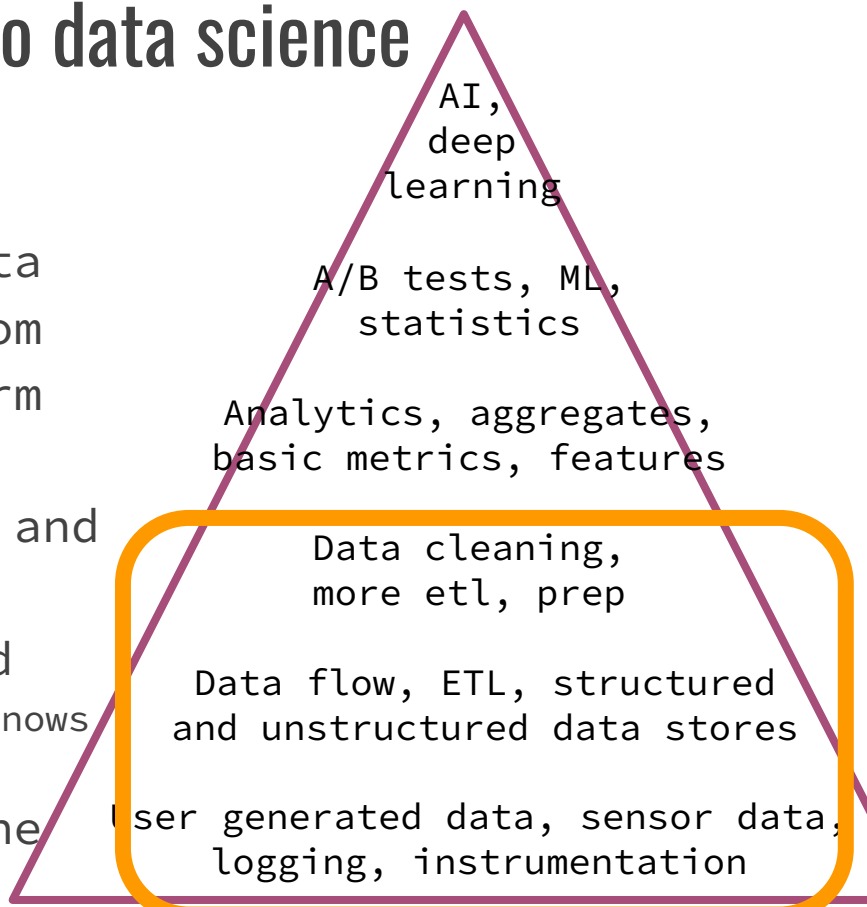
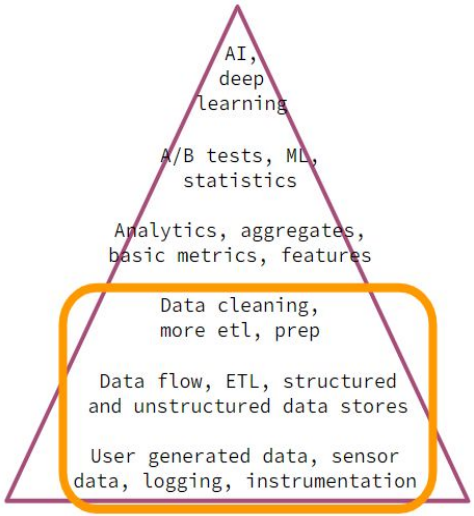


Fig adapted from - Ref

So what's a data engineering again?

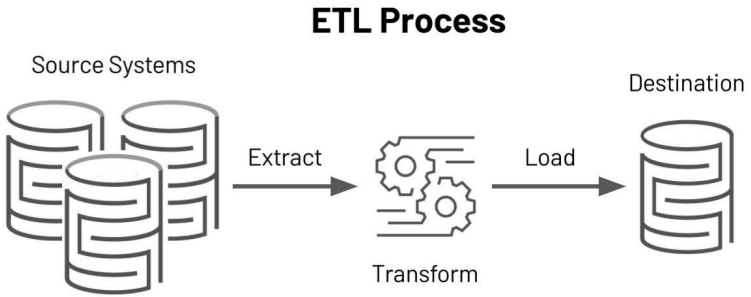
- Should be clear why we need data engineering
- In continue we dive deeper into
  - Where data comes from - bottom of the pyramid
  - What tasks DEs do to make it useful - next two levels of pyramid
- Read Hammerbacher 2009 - Rise of the Data Scientist
  - Great story of a DS/DE who helped build the early data infrastructure of Facebook
- Read Rogati 2017 - The AI Hierarchy of Needs
  - Short but good blog post on why DE is fundamental



Where are we going?

- Talk about where all these data are coming from
- The (generally) main job of a DE - making *ETLs
- Technologies using in DE and what subset we'll use

*ETL: Extract, transform, load



https://databricks.com/glossary/extract-transform-load

Not all Unstructured

- Of course, not all data collected is structured like this
- Some is just stored in a database across multiple tables
  - Each transaction in a convenience store

TABLE ID: STORE		
store_id	**store_state**	**country**
il_23	IL	USA
az_45	AZ	USA
ca_12	CA	USA
to_39	Ontario	Canada

TABLE ID: TRANSACTIONS			
transact_id	**store_id**	**UPC**	**price**
x88943	il_23	49914	2.57
x88943	il_23	99371	1.99
a85921	to_39	95831	8.99
a85921	to_39	99492	5.49
a85921	to_39	27482	4.49
z88930	az_45	33491	0.99

Not all Structured

- Of course, not all data collected is structured like this
- Some is just stored in a database across multiple tables
  - Each transaction in a convenience store
  - And data collected might not be optimized

A	B	C	D	E	F	G	H	I	J
id	name	host_id	host_name	neighbourhood_g	neighbourhood	latitude	longitude	room_type	price
2539	Clean & quiet apt	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149
2595	Skylit Midtown Ca	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225
3647	THE VILLAGE O	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.9419	Private room	150
3831	Cozy Entire Floor	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89
5022	Entire Apt: Spaci	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80
7322	Chelsea Perfect	18946	Doti	Manhattan	Chelsea	40.74192	-73.99501	Private room	140
7726	Hip Historic Brow	20950	Adam And Charit	Brooklyn	Crown Heights	40.67592	-73.94694	Entire home/apt	99
7750	Huge 2 BR Uppe	17985	Sing	Manhattan	East Harlem	40.79685	-73.94872	Entire home/apt	190
7801	Sweet and Spaci	21207	Chaya	Brooklyn	Williamsburg	40.71842	-73.95718	Entire home/apt	299
8024	CBG CtyBGd He	22486	Lisel	Brooklyn	Park Slope	40.68069	-73.97706	Private room	130
8025	CBG Helps Haiti	22486	Lisel	Brooklyn	Park Slope	40.67989	-73.97798	Private room	80
8110	CBG Helps Haiti	22486	Lisel	Brooklyn	Park Slope	40.68001	-73.97865	Private room	110

So what does a DE do again?

- Collects data from these various databases that are recording events/transactions/information
- Reorganizes it in some way or another into a format that lets end user do analytics or data science tasks
- stores it in a database for end user to use
- This process has a general name - **ETL**
  - **Extract - Transform - Load**

ETL

- **ETLs are essentially the core of DE**
- That raw data in structured, semi-structured, or unstructured format is all stored in a **data lake**
- The transform step is going to remove errors, create features, scale values, aggregate data for metrics and whatever else is needed to support analytics and DS
- The transformed data is stored in a **data warehouse**

ETL

- From reading - Ch1 Data Mining Concepts and Techniques

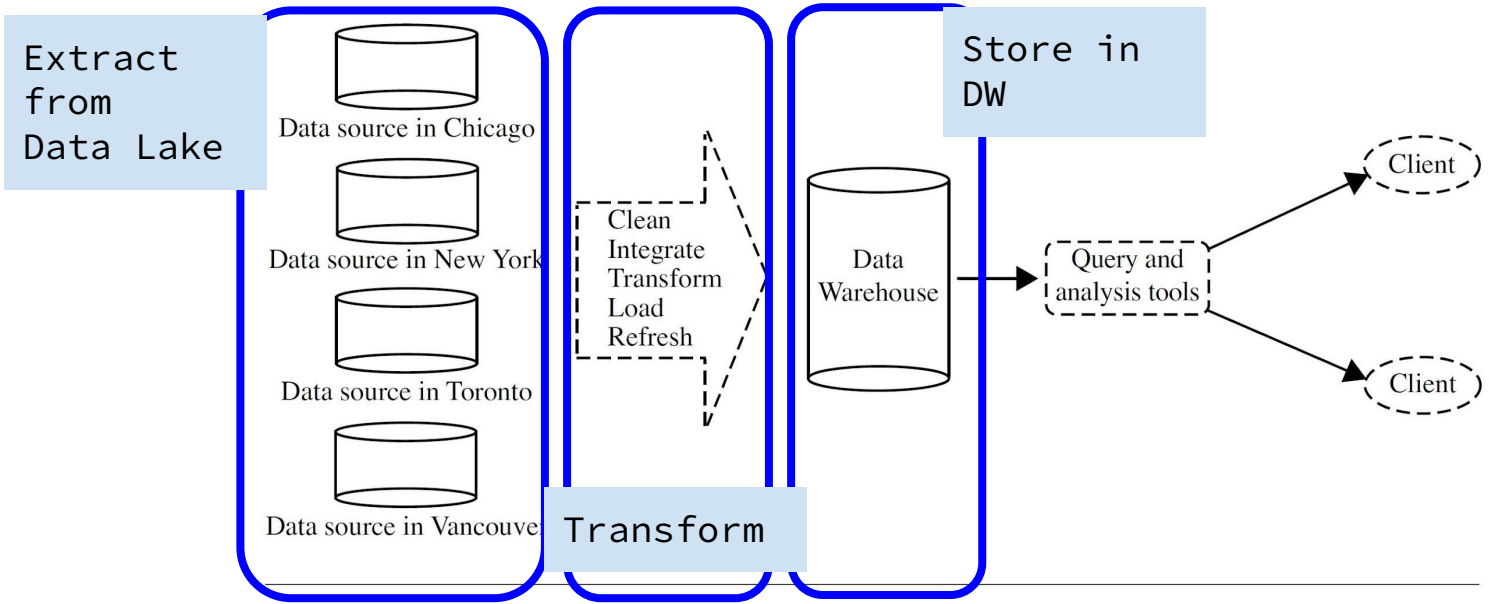
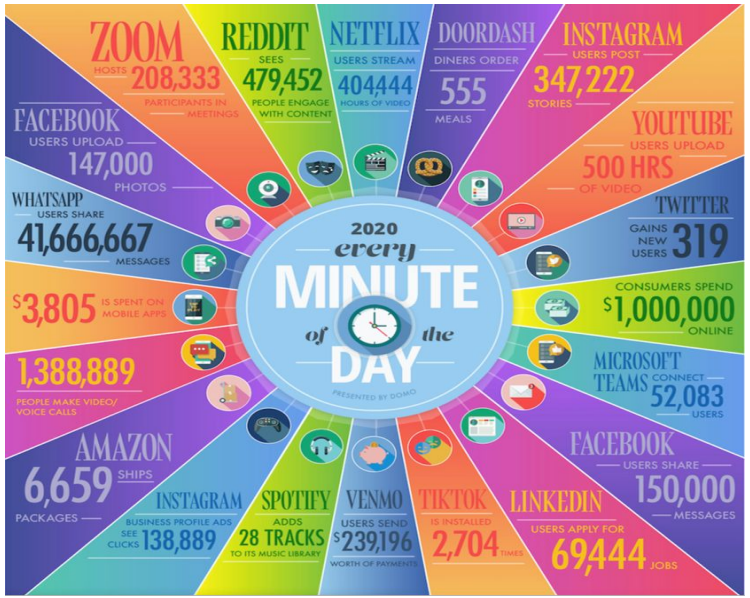


Figure 1.6 Typical framework of a data warehouse for *AllElectronics*.

But how to deal with so many events?

- Our goal is to get the data into a useful format
- But we're dealing ***lots*** of data
- Average computer has say 16gb of memory
- Obviously this is the other challenge of DE
  - How to deal with massive volumes of data fast enough to be useful
  - Can't let it take hours/days/weeks to process on one machine



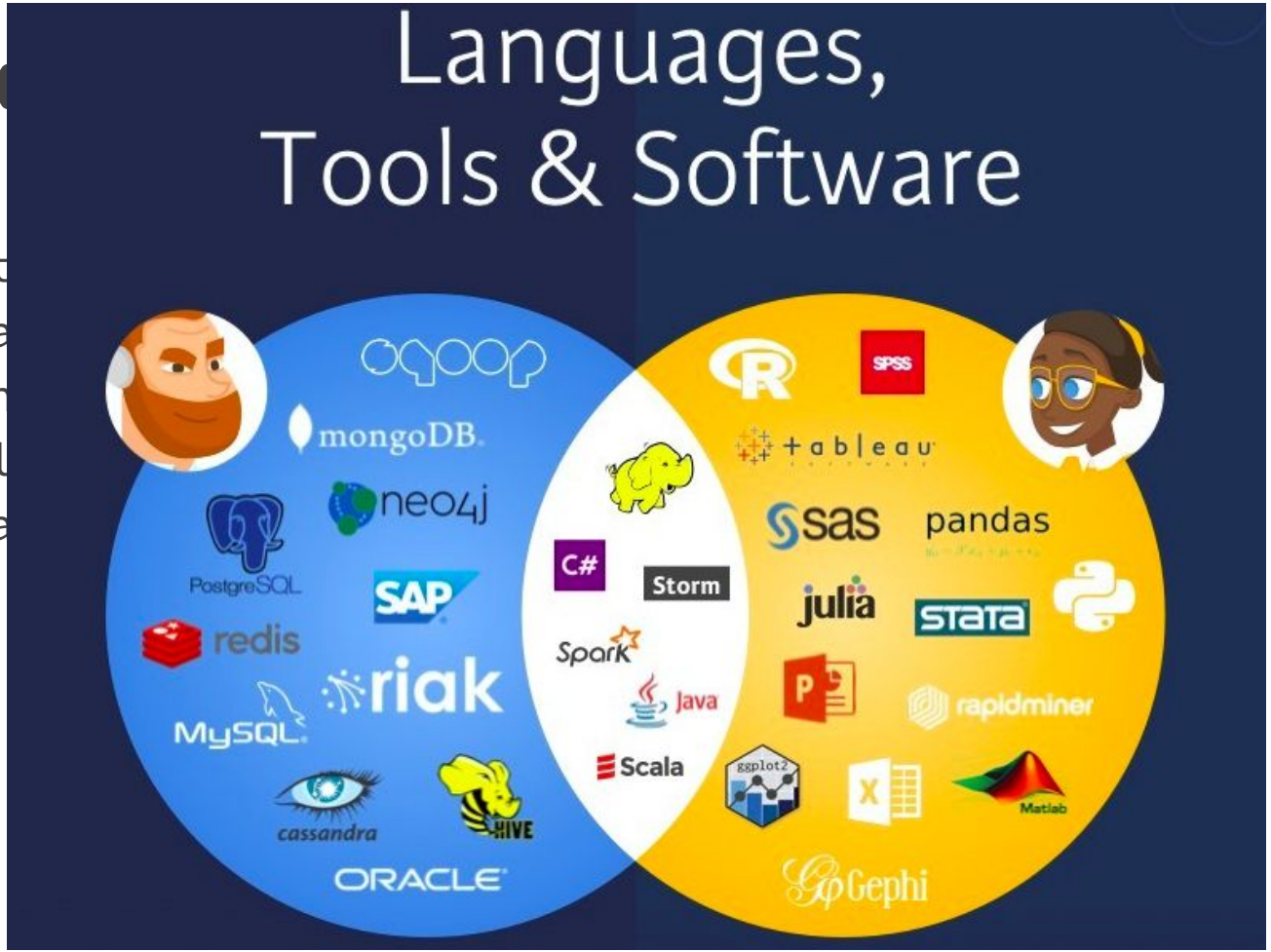
Source: domo.com

Enter big data technologies

- The other part of of being a DE is using big data processing frameworks that allow for much, much faster data processing
- Technologies like hadoop/mapreduce and Spark utilized clusters of machines to distribute framework and optimize speed

Enter big data tec

- The other part
  processing fra
  data processin
- Technologies l
  clusters of ma
  speed



Enter big data technologies

- The other part of of being a DE is using big data processing frameworks that allow for much, much faster data processing
- Technologies like hadoop/mapreduce and Spark utilized clusters of machines to distribute framework and optimize speed
- It’s a massive ecosystem of tools - We’re only going to learn some of the essential tools

A bit more about the technologies we're going to use

- Languages / technologies
  - Python and pandas
  - SQL - MySQL
  - Pyspark locally
  - Pyspark via Databricks
- Environments
  - We'll be working in Jupyter Notebooks
  - Use Google Colaboratory - Google cloud based Jupyter Notebook
    - You'll download a notebook, upload and open there
  - You're welcome to use a local install, but I won't be providing tutorials (I can't troubleshoot 40+ installs of all the libraries)
  - Databricks - Cloud notebook based analytics/DS platform