CS150: Database & Datamining

Lecture 18: Analytics & Machine Learning I

Xuming He

Spring 2019

Acknowledgement: Slides are adopted from the Berkeley course CS186 by Joey Gonzalez and Joe Hellerstein, Stanford CS145 by Peter Bailis.

Transaction Processing vs Analytics

Online Transaction Processing (OLTP)

- Many small queries:
  - Freq. use of indexes
  - Many writes
  - Concurrency and Logging
- Managing the "Now"
  - Source of truth
- Fairly simple queries with few predicates and relations

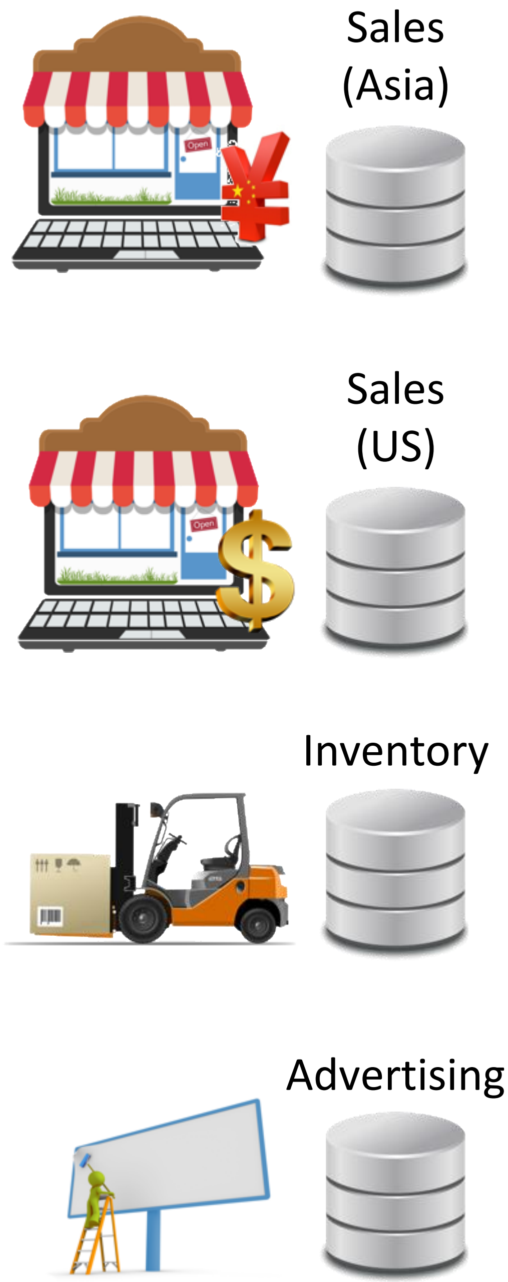
Online Analytics Processing (OLAP) & Data Mining/ML

- Exploratory Full Table Queries
  - e.g., Agg. Sales Per Market
  - Infrequent (but bulk) writes
  - Limited transaction processing
- Recording the history
  - What was our inventory at the end of last two quarters
- Complex queries with many predicates and many relations

Analytics & ML queries:

- What was our total sales by market last quarter?
  - Summarization
- What is our predicted sales for next quarter?
  - Forecasting
- Which users will likely leave our service?
  - Churn prediction
- If a user buys X what else are they likely to buy?
  - Collaborative filtering & Recommender Systems

You embark on the journey of a data scientist …

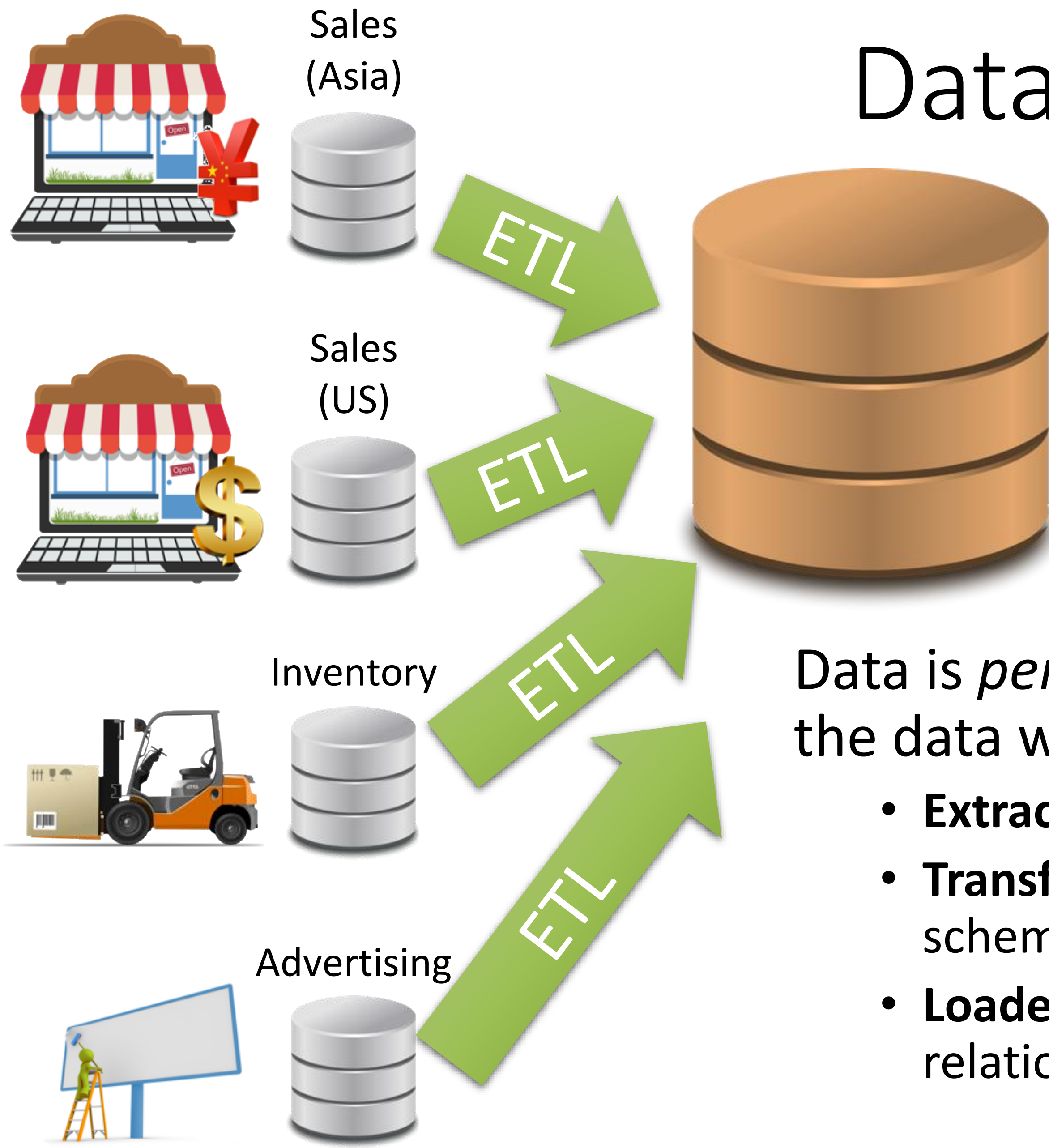


Data Everywhere

- ➢ Stored Across Multiple Operational OLTP Systems
  - Different formats (e.g., currency)
    - Different schemas (acquisitions ...)
  - Mission critical
    - Serving live sales traffic
    - Managing inventory
    - ... Be careful!
- ➢ Often limited historical data

We would like a consolidated, cleaned, historical snapshot of the data.

Data Warehouse



Collects and organizes historical data from multiple sources

Data is *periodically* **ETL**ed into the data warehouse:

- **Extracted** from remote sources
- **Transformed** to standard schemas
- **Loaded** into the (typically) relational system

Extracting Data from Sources

- Need to collect data from multiples sources
  - Various RDBMS vendors
  - Structured files JSON, XML

- Often done using SQL interfaces

- Validate extracted data
  - Flag corrupted records …

Transforming “Cleaning” Data

- Additional data validation and filtering

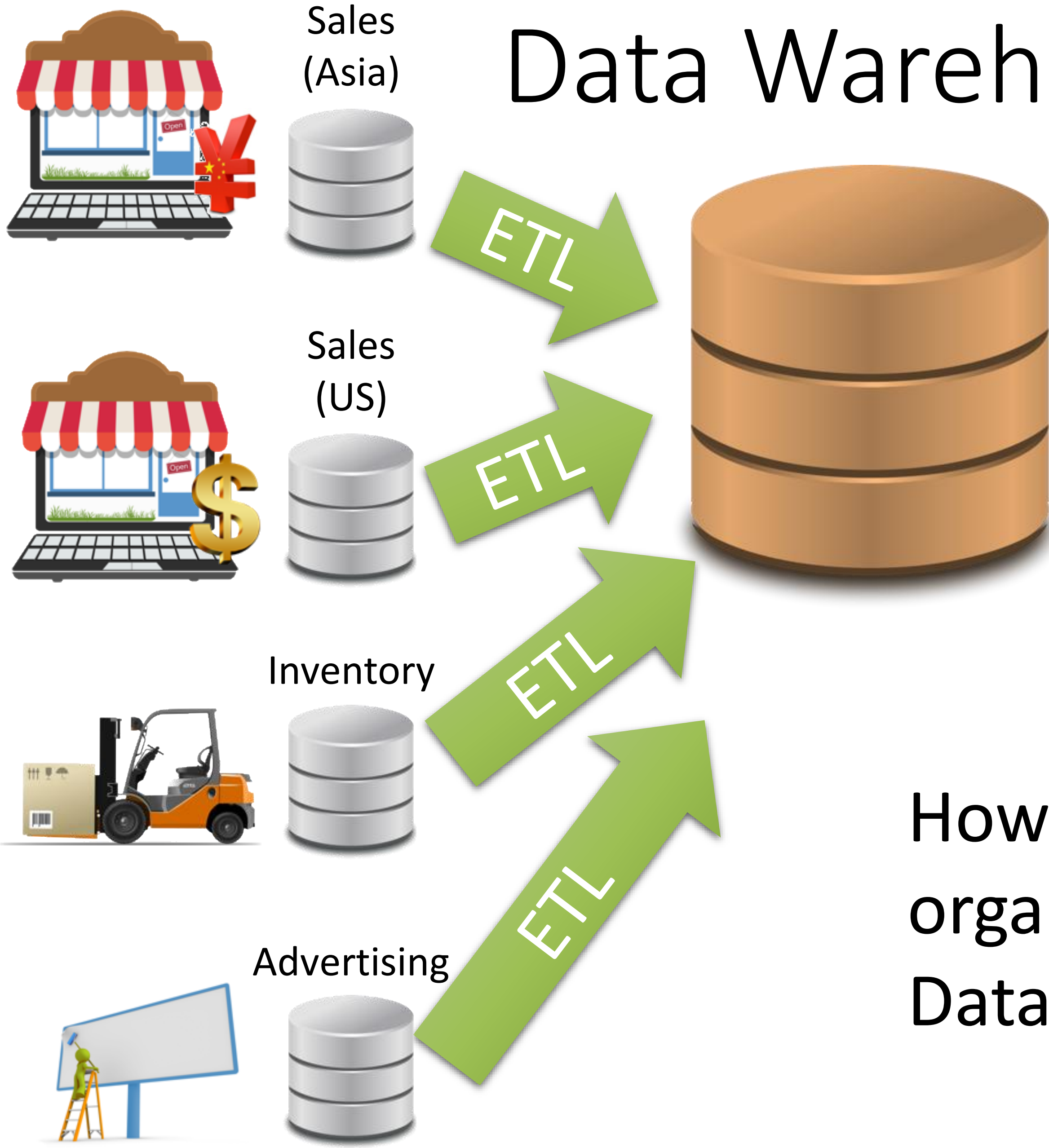
- Schema manipulation
  - Extract key fields
  - Encoding text
  - Verifying and enforcing constraints

- Data normalization (time zones, currency)

Loading Data

- Data is bulk loaded into large relations
  - Fact tables … (more on this later)
- Update:
  - Indexes
  - Metadata tables: Data about the data
    - When and how was it collected
    - Meaning of fields
  - Updating materialized views … (more on this later)
- Occasionally move older data to archival storage
  - Data aging

Data Warehouse



How is data organized in the Data Warehouse?

Example Sales Data:

pname	category	price	qty	date	day	city	state	country
Corn	Food	25	25	3/30/16	Wed.	Omaha	NE	USA
Corn	Food	25	8	3/31/16	Thu.	Omaha	NE	USA
Corn	Food	25	15	4/1/16	Fri.	Omaha	NE	USA
Galaxy 1	Phones	18	30	1/30/16	Wed.	Omaha	NE	USA
Galaxy 1	Phones	18	20	3/31/16	Thu.	Omaha	NE	USA
Galaxy 1	Phones	18	50	4/1/16	Fri.	Omaha	NE	USA
Galaxy 1	Phones	18	8	1/30/16	Wed.	Omaha	NE	USA
Peanuts	Food	2	45	3/31/16	Thu.	Seoul		Korea
Galaxy 1	Phones	18	100	4/1/16	Fri.	Seoul		Korea

- **Big** table: many *columns* and *rows*
  - Substantial redundancy → expensive to store and access
- Could we organize the data a little better?

Multidimensional Data Model

Sales **Fact Table**

pid	timeid	locid	sales
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
12	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35
11	2	2	22
11	3	2	10
12	1	2	26

Locations

locid	city	state	country
1	Omaha	Nebraska	USA
2	Seoul		Korea
5	Richmond	Virginia	USA

Dimension Tables

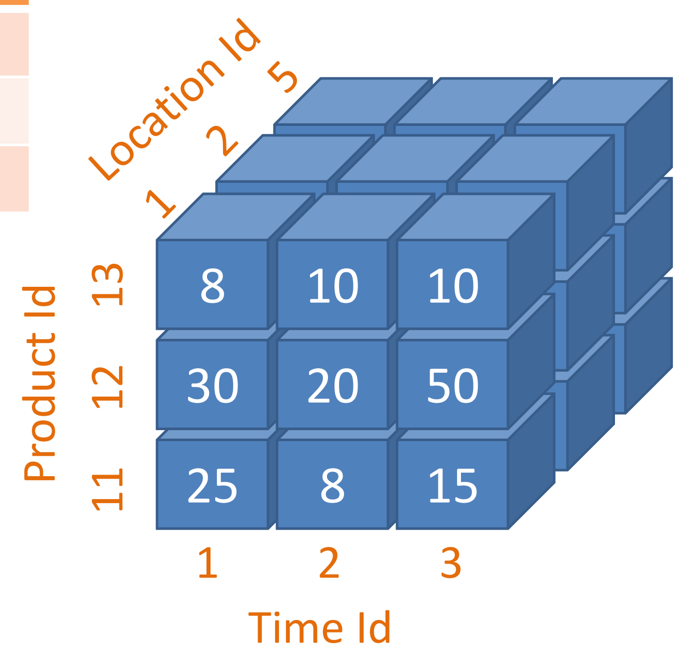
Products

pid	pname	category	price
11	Corn	Food	25
12	Galaxy 1	Phones	18
13	Peanuts	Food	2

Time

timeid	Date	Day
1	3/30/16	Wed.
2	3/31/16	Thu.
3	4/1/16	Fri.

- Multidimensional "Cube" of data



Multidimensional Data Model

Sales Fact Table

pid	timeid	locid	sales
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
12	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35
11	2	2	22
11	3	2	10
12	1	2	26

Locations

locid	city	state	country
1	Omaha	Nebraska	USA
2	Seoul		Korea
5	Richmond	Virginia	USA

Dimension Tables

Products

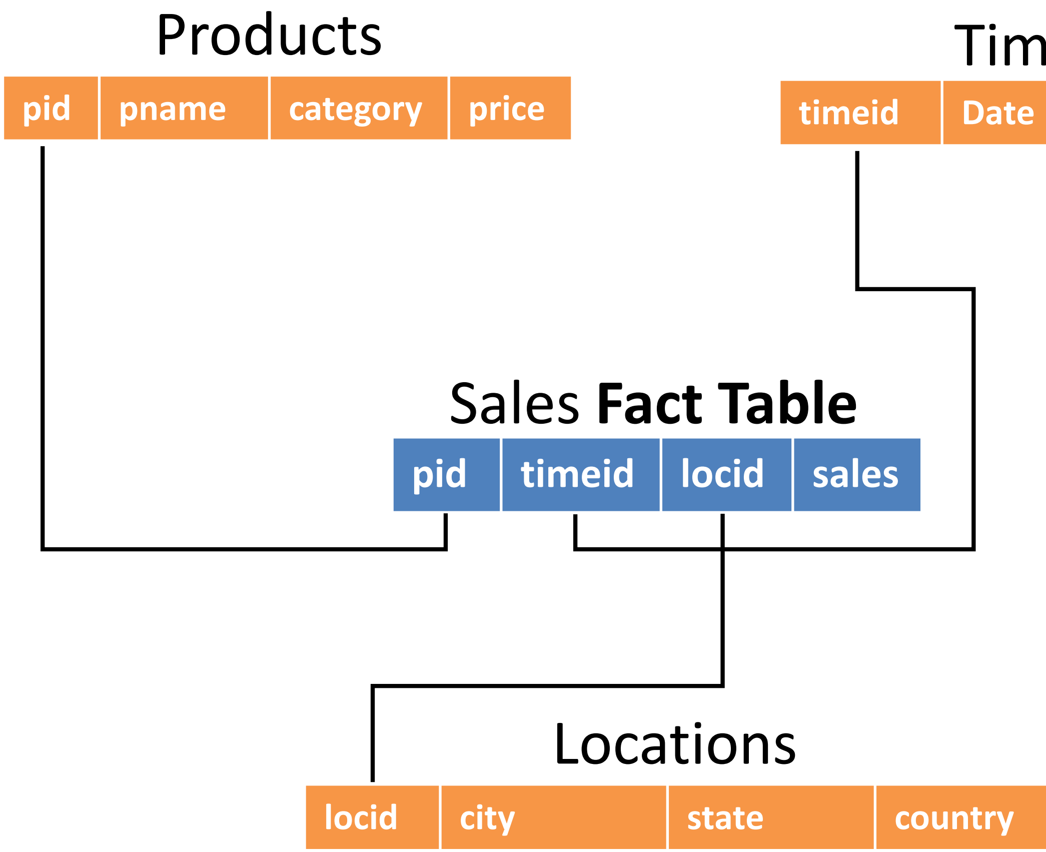
pid	pname	category	price
11	Corn	Food	25
12	Galaxy 1	Phones	18
13	Peanuts	Food	2

Time

timeid	Date	Day
1	3/30/16	Wed.
2	3/31/16	Thu.
3	4/1/16	Fri.

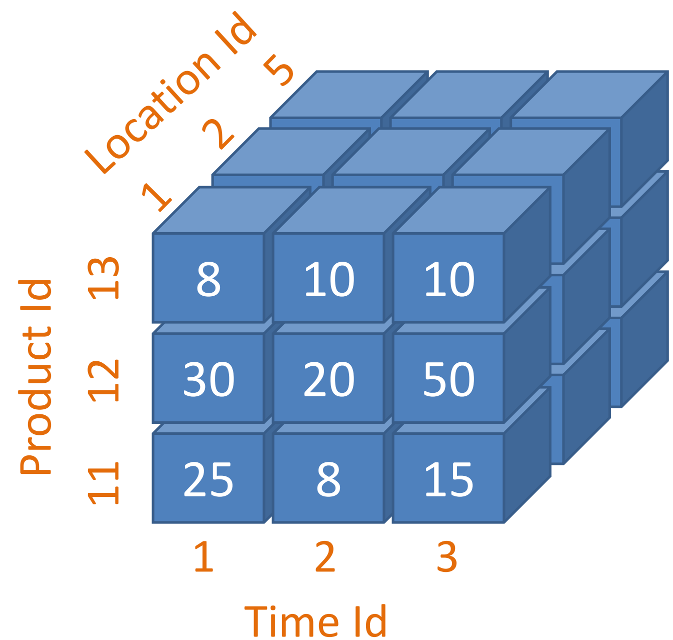
- Sales Fact Table
  - Contains only foreign keys → Efficient
- Easy to manage Dimensions
  - Galaxy1 → Phablet: no need to update **Fact Table**
- Normalization
  - Minimizing redundancy
  - More on this later …

Multidimensional Data: **Star Schema**

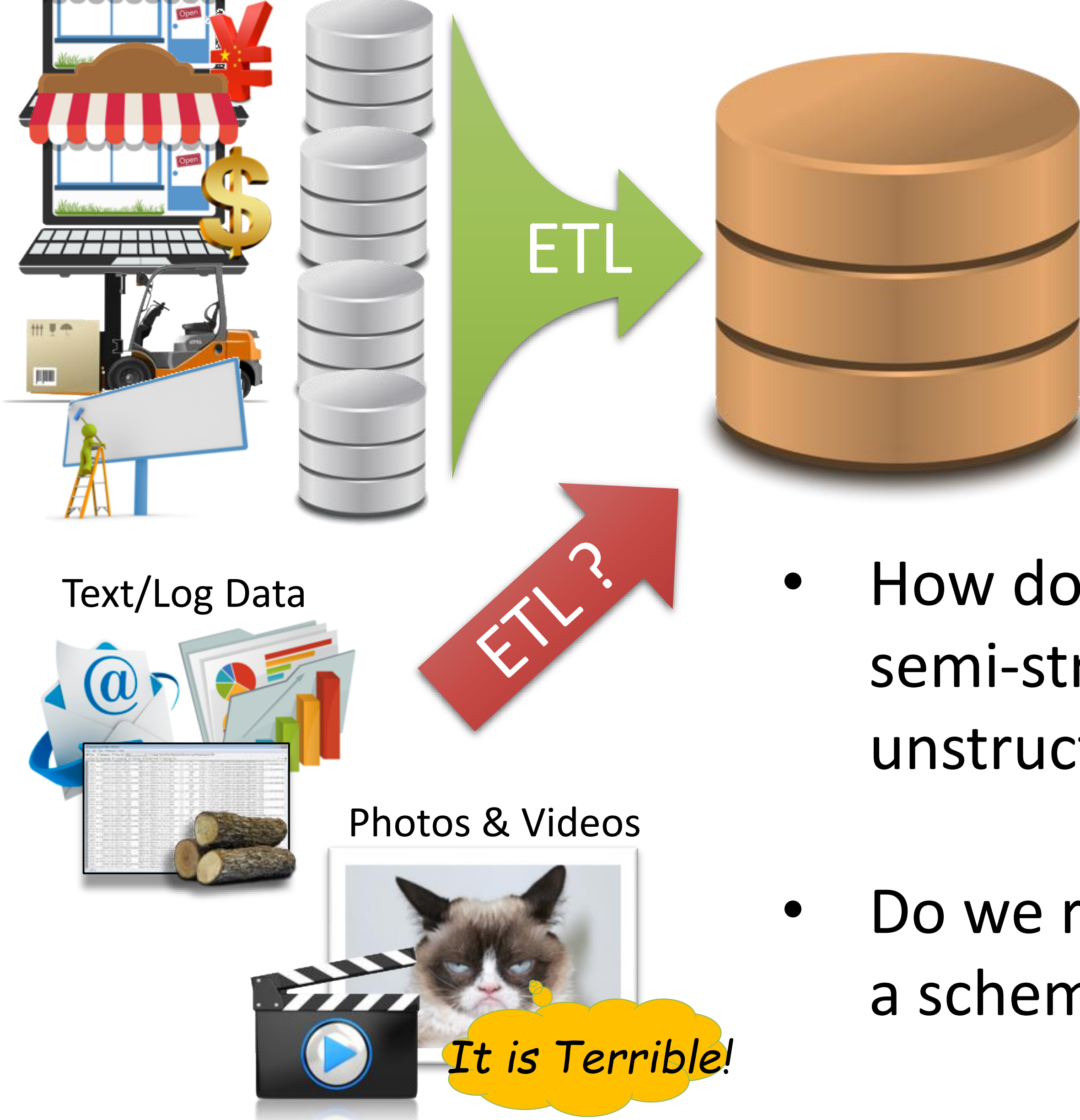


Dimension Tables

← This looks like a star …

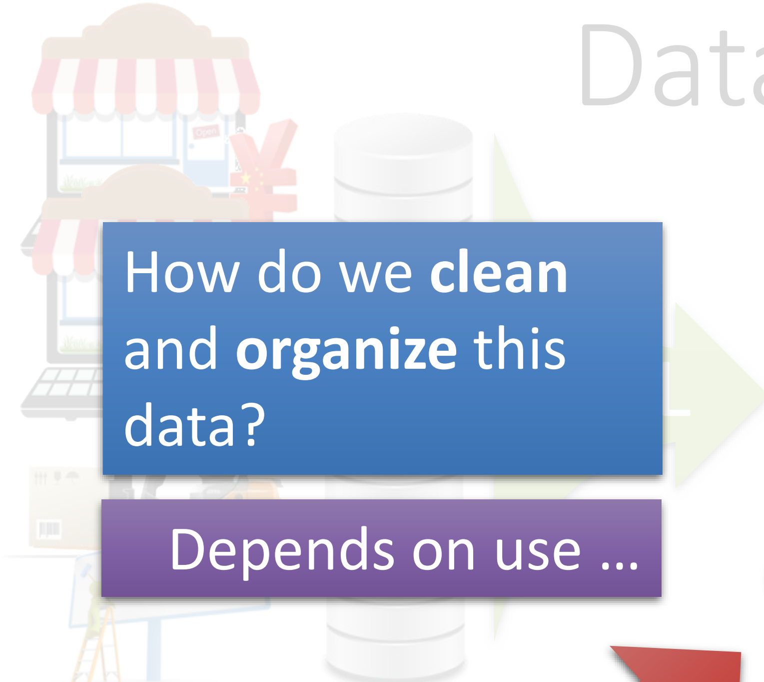


Data Warehouse



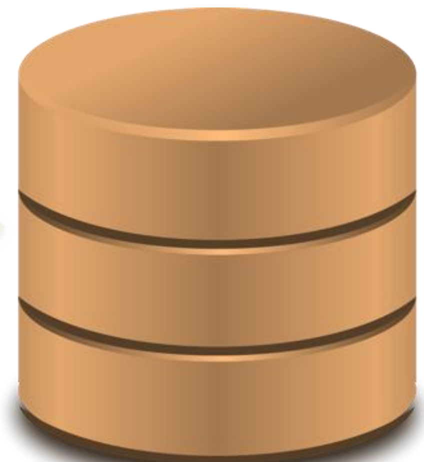
- How do we deal with semi-structured and unstructured data?
- Do we really want to force a schema on load?

Data Warehouse



How do we **clean** and **organize** this data?

Depends on use …



Text/Log Data

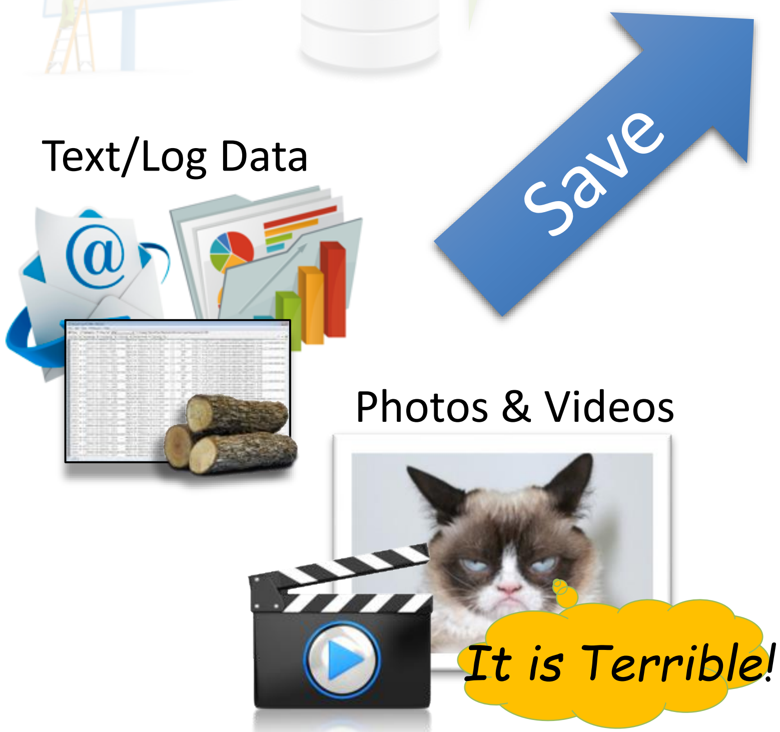
Photos & Videos

How do we **load** and **process** this data in a relation system?

Depends on use …
Can be difficult ...
Requires thought ...

Data Lake*

*Still being defined...
[Buzzword Disclaimer]

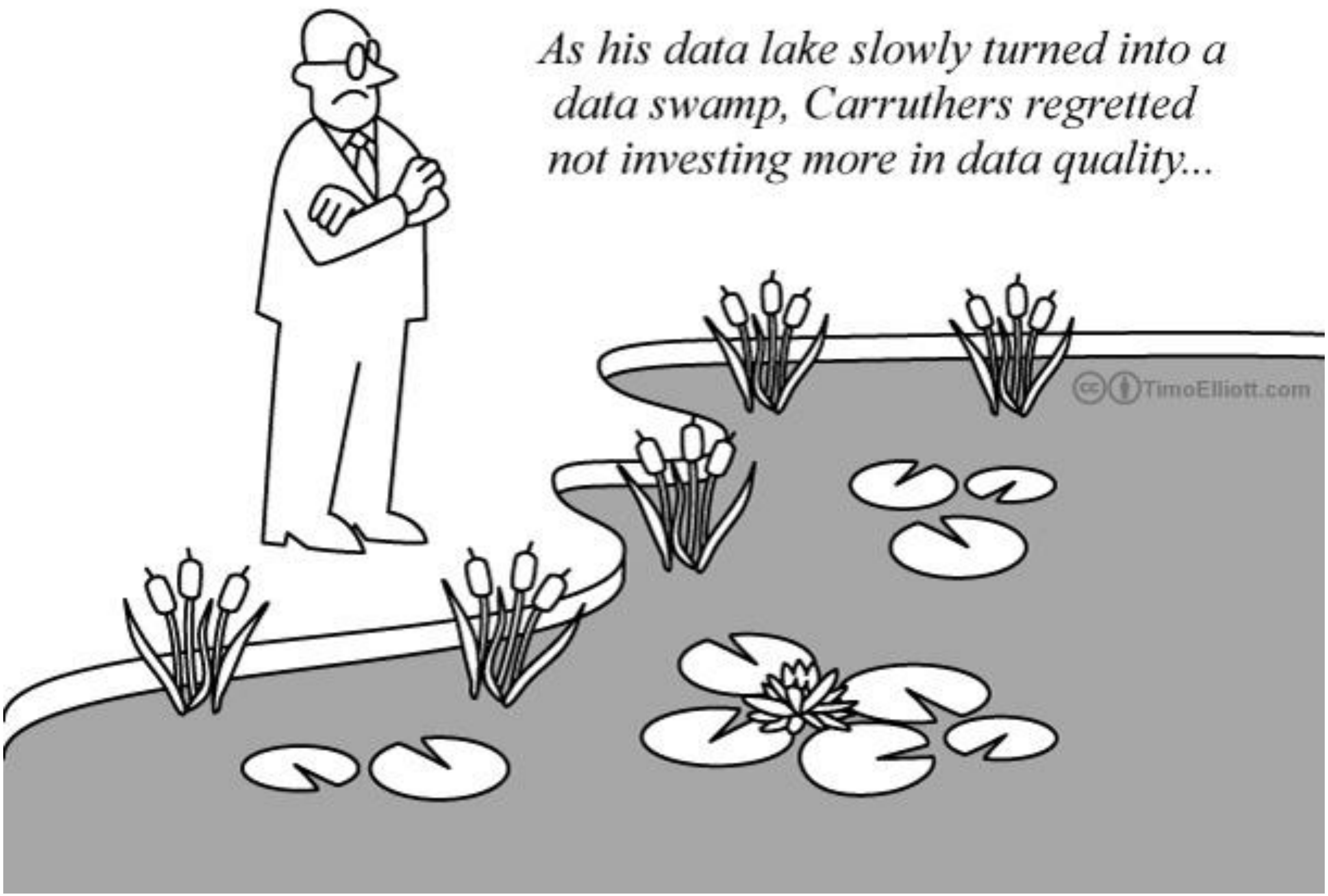


Big Idea:
Maintain a copy of all the data in one place and *free** data consumers to choose how to transform and use it.

*free to solve all the problems themselves

Data Lake

- Store unstructured data in **raw form**
  - Schema-on-**Read**: *determine the best organization **when data is used***
  - **Contrast:** Data Warehouses are Schema-on-Load (ET***L***)
    - Plan ahead (Fact tables and Dimensions)
- Often much **larger** than data warehouses
- Technologies
  - **Storage:** Large distributed file systems (e.g., HDFS)
    - Semi-structured formats (JSON, Parquet)
  - **Computation:** Map-Reduce
    - Recent trend to add SQL (or SQL like) functionality
- More Agile (?):
  - Don't worry about schema & verification when loading
  - Disaggregated compute and storage → BYOF
    - bring your own compute frameworks …
- **What could go wrong?**



http://timoelliott.com/blog/2014/12/from-data-lakes-to-data-swamps.html

Data Lake → Data Swamp

- Cultural shift: *Curate* → Save **Everything**!
  - Signal to Noise ratio drops …
- Limited data governance → more agile →
  hdfs://important/joey_big_file3.csv_with_json
  - **What** does it contain? **What** are all the **"fields"**
  - **When** and **how** and **from where** was it created
- Without cleaning and verification we begin to collect a rich history of **dirty data**
- Limited compatible with traditional tools

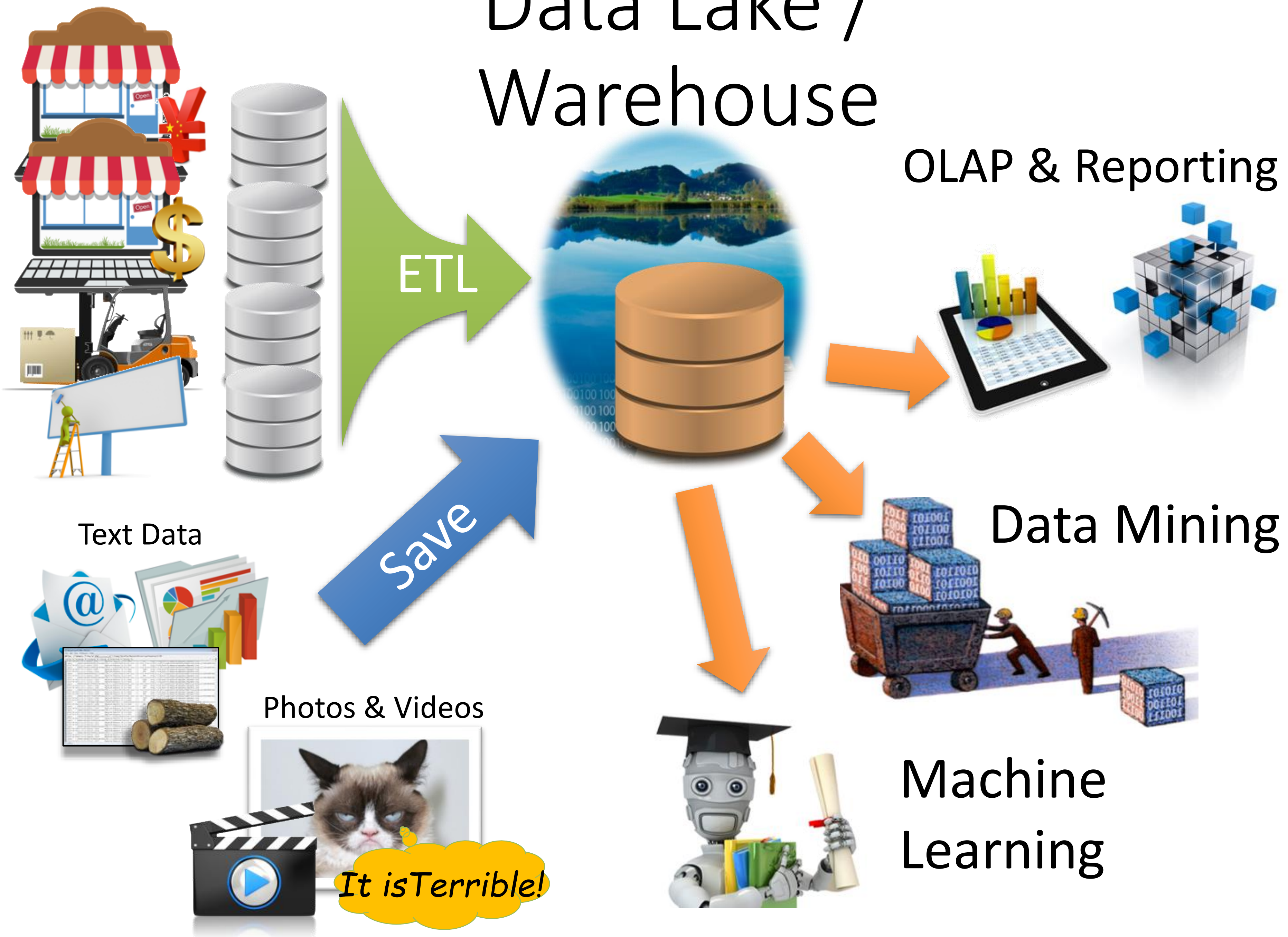
Data Lakes *Appear* to be Maturing

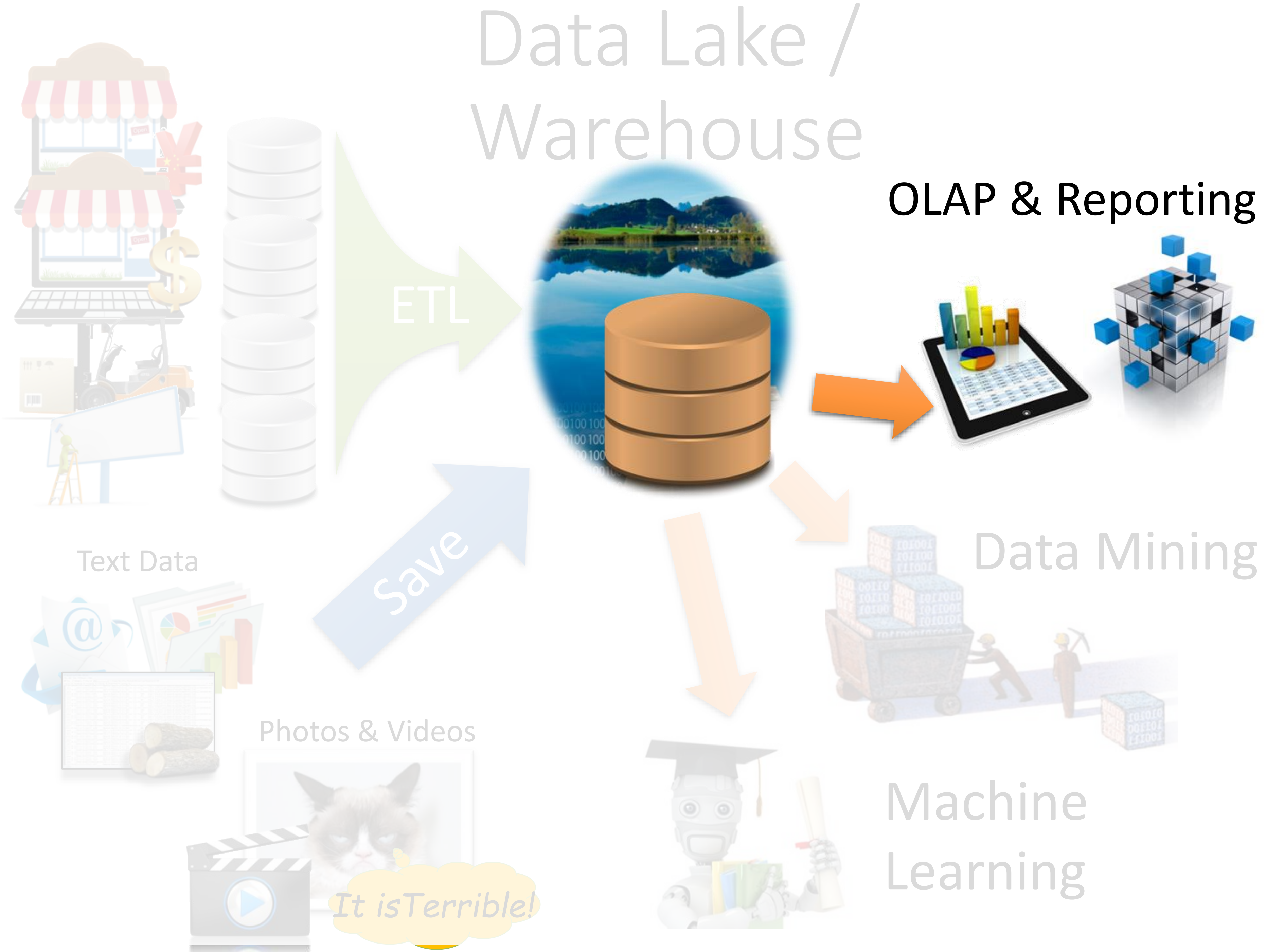
- Relational data-models + SQL:
  - **Hive:** SQL on top of Hadoop Map-Reduce
  - **SparkSQL:** SQL on top of Spark
- Tools are Improving:
  - Better data cleaning
  - Catalog Managers
  - Improved semi-structured “raw” data formats
- Improved data governance
  - Organization are recognizing the issues

Data Lake / Warehouse

What do we do with all this data?

Data Lake / Warehouse




Data Lake /
Warehouse
OLAP & Reporting
ETL
Save
Data Mining
Text Data
Photos & Videos
It isTerrible!
Machine
Learning

Online Analytics Processing (OLAP)

Users interact with multidimensional data:

- ➢Constructing ad-hoc and often complex SQL queries
- ➢Using graphical tools that to construct queries
- ➢Sharing views that summarize data across important dimensions

Cross Tabulation (Pivot Tables)

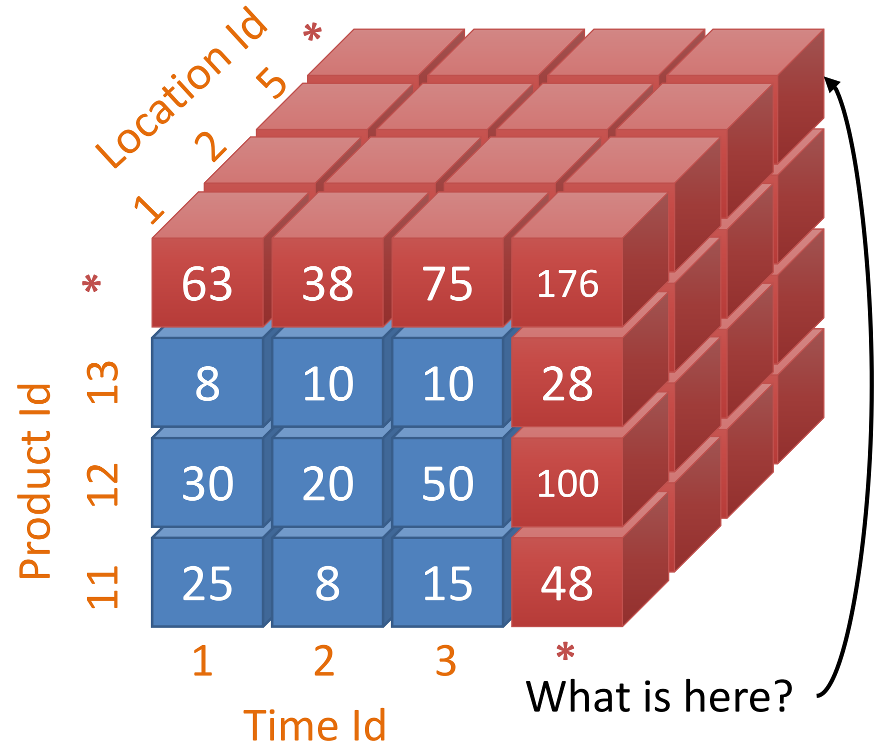
Item	Color	Quantity
Desk	Blue	2
Desk	Red	3
Sofa	Blue	4
Sofa	Red	5

		Item		
		Desk	Sofa	***Sum***
Color	Blue	2	4	***6***
	Red	3	5	***8***
	Sum	***5***	***9***	***14***

- Aggregate data across pairs of dimensions
  - **Pivot Tables:** *graphical interface* to select dimensions and aggregation function (e.g., SUM, MAX, MEAN)
  - **GROUP BY** queries
- Related to contingency tables and marginalization in stats.
- What about many dimensions?

Cube Operator

- Generalizes cross-tabulation to higher dimensions.



- In SQL:

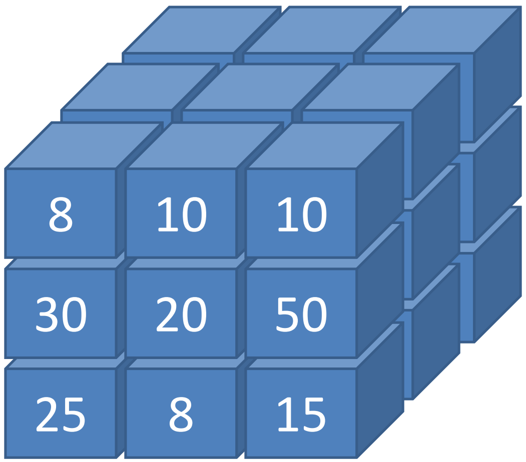
SELECT Item, Color, **SUM**(Quantity) **AS** QtySum
FROM Furniture
GROUP BY ***<u>CUBE</u>*** (Item, Color);

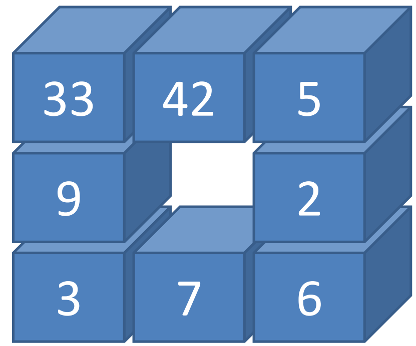
Item	Color	Quantity
Desk	Blue	2
Desk	Red	3
Sofa	Blue	4
Sofa	Red	5

Item	Color	QtySum
Desk	Blue	2
Desk	Red	3
Desk	*	5
Sofa	Blue	4
Sofa	Red	5
Sofa	*	9
*	*	14
*	Blue	6
*	Red	8

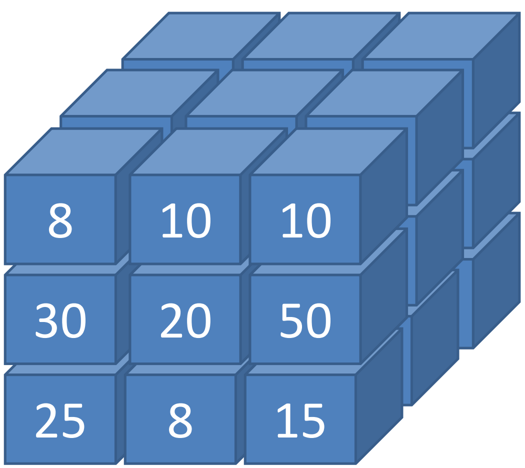
OLAP Queries

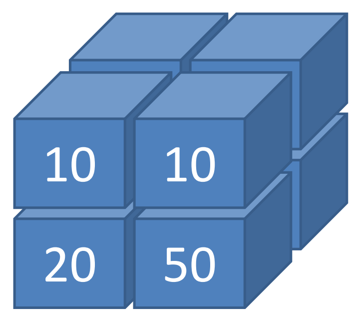
- **Slicing:** *selecting a value for a dimension*





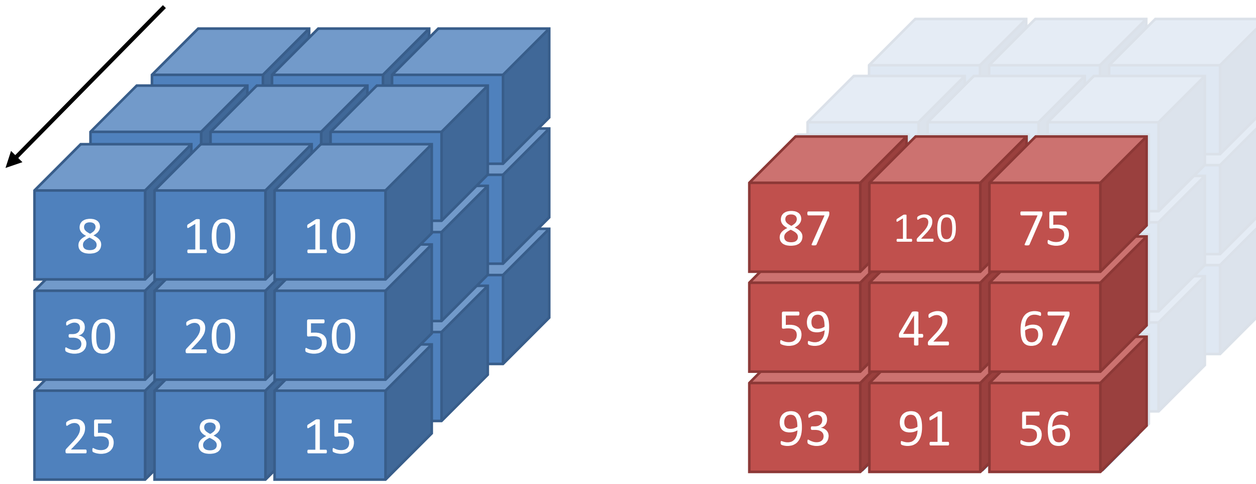
- **Dicing:** *selecting a range of values in multiple dimension*



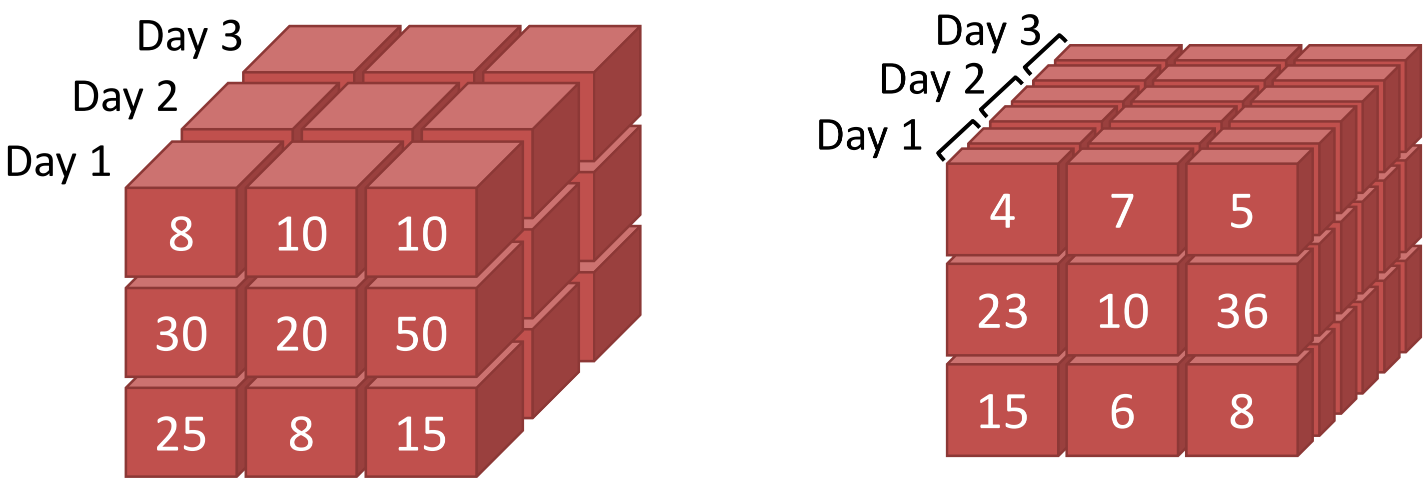


OLAP Queries

- **Rollup:** *Aggregating along a dimension*

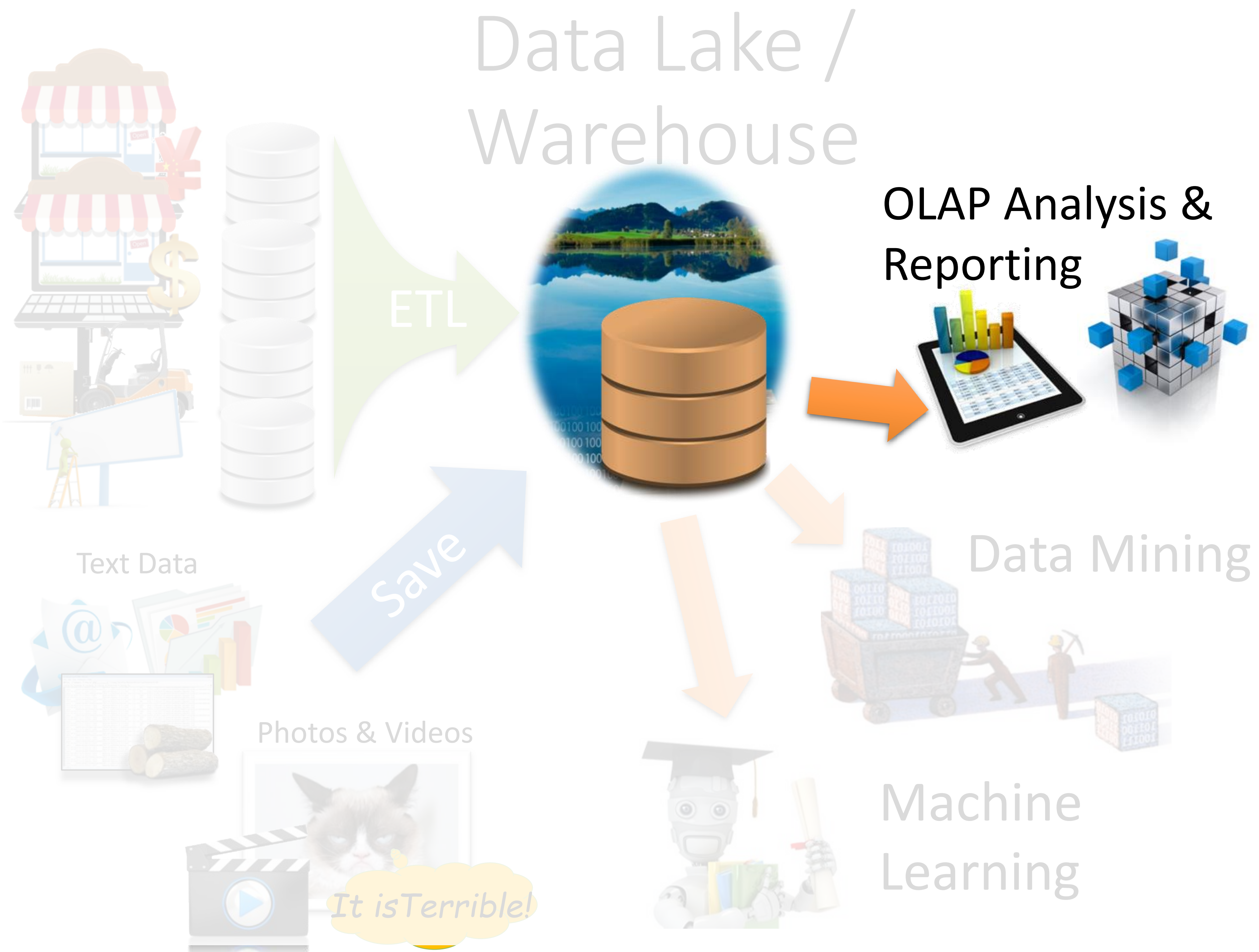


- **Drill-Down:** *de-aggregating along a dimension*

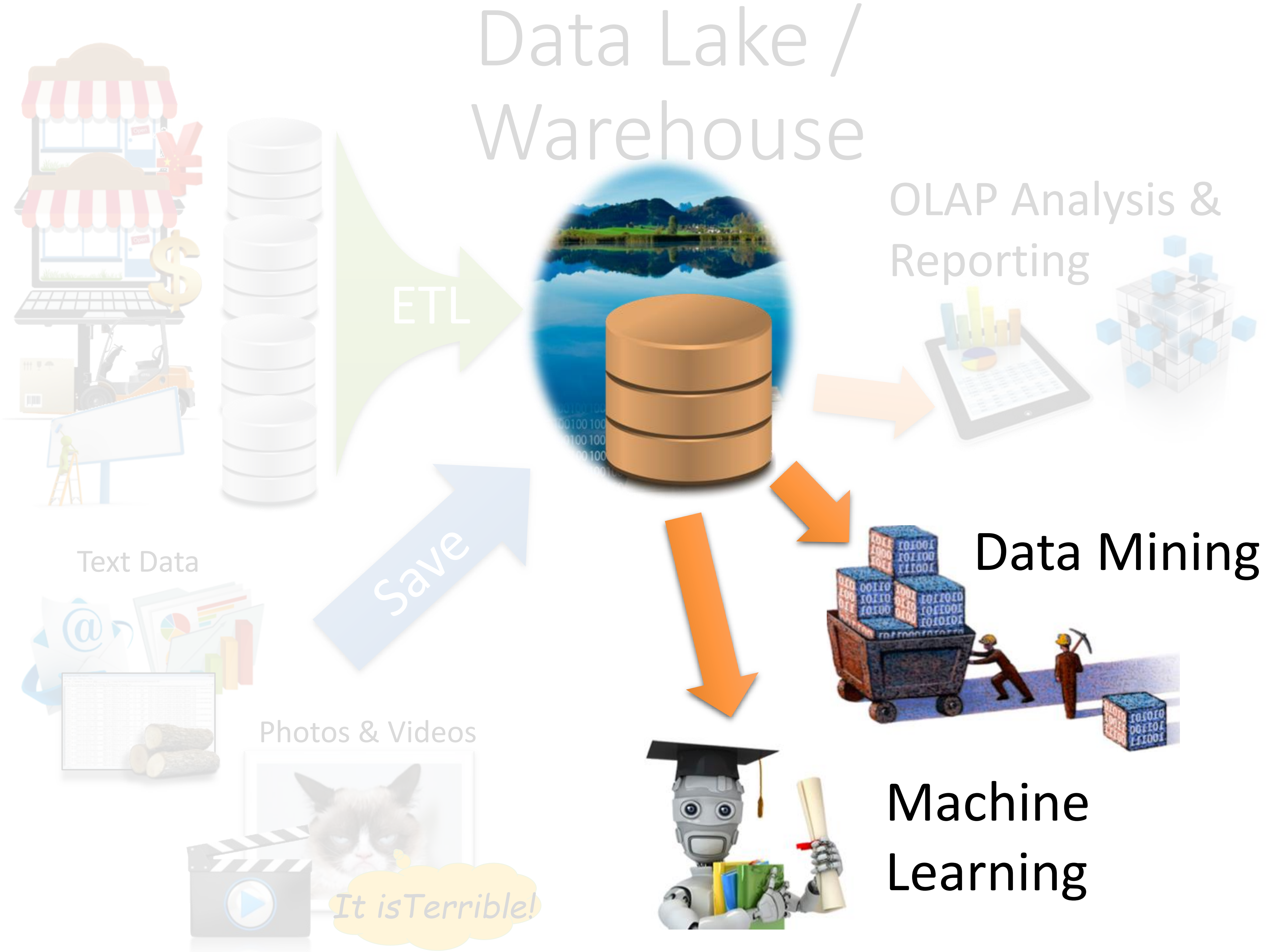


Reporting and Business Intelligence (BI)

- Use high-level tools to interact with their data:
  - Automatically generate SQL queries
    - Queries can get big!
- Common!



Data Lake /
Warehouse
OLAP Analysis &
Reporting
ETL
Save
Text Data
Photos & Videos
It isTerrible!
Data Mining
Machine
Learning

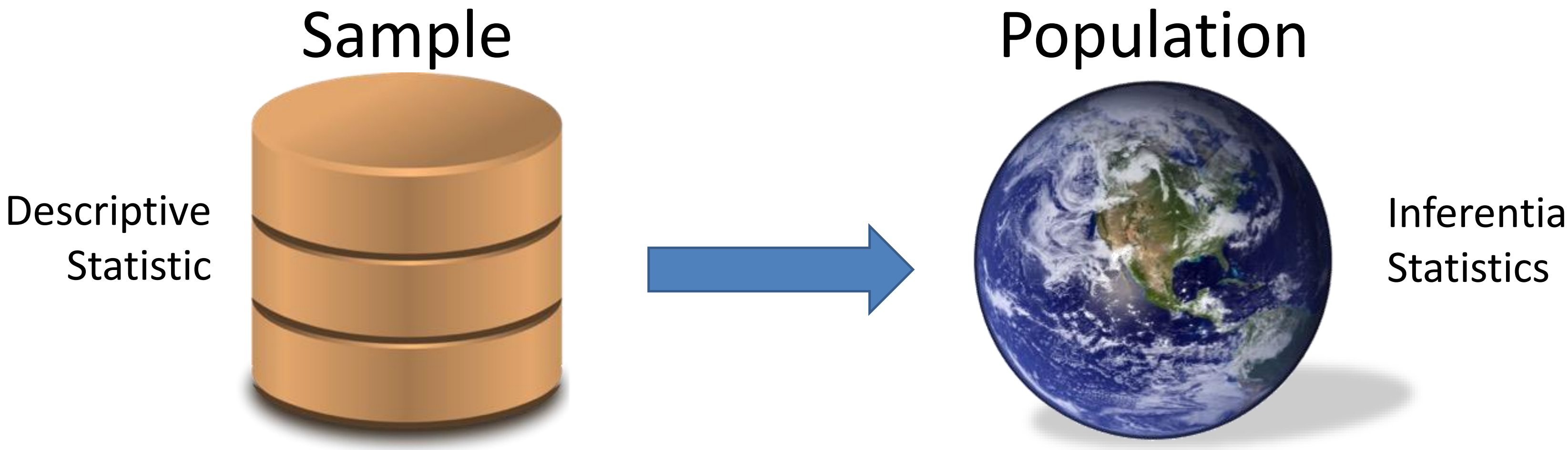


Data Lake /
Warehouse
ETL
OLAP Analysis &
Reporting
Save
Text Data
Photos & Videos
It isTerrible!
Data Mining
Machine
Learning

Knowledge Discovery in Databases (KDD)

- Process of extracting ***knowledge*** from a ***data***
  - What does this mean?

Descriptive vs. **Inferential** Statistics



- **Descriptive Statistics:** *describe* the sample data
  - <u>Example:</u> *Average* sales last quarter
  - Can be **measured directly** from the database
- **Inferential Statistics:** *estimate* the population
  - <u>Example:</u> *Expected* sales next quarter
  - May be **estimated** using descriptive statistics

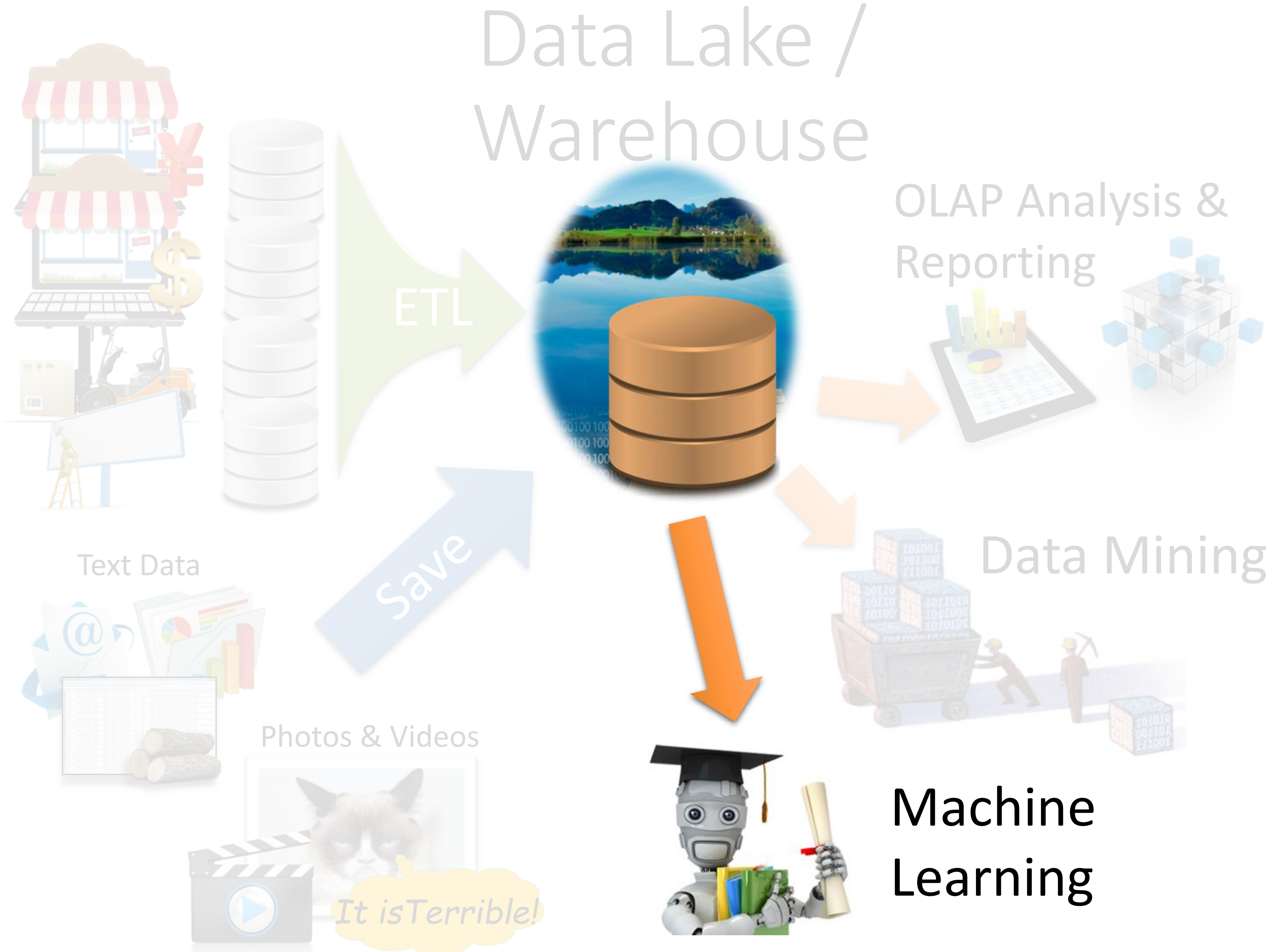
The Basic KDD Process

- **Data Selection:** *What data do I need for a given task?*
  - If data was already collected, how was the data collected?

- **Data Cleaning:** *Preparing the data for a given task*
  - Typically most challenging (time consuming) part.
  - Why might ETL not be enough?

- **Data Mining & ML:** *Running algorithms to infer patterns*
  - The fun part! Many tools, many options, complex tradeoffs.

- **Evaluation:** *Verifying that patterns are significant*
  - Algorithms will typically find patterns especially when none exist.


Data Lake /
Warehouse
ETL
OLAP Analysis &
Reporting
Data Mining
Text Data
Save
Photos & Videos
It isTerrible!
Machine
Learning

What is Machine Learning?

Study of algorithms that:

- That improve their **performance**
  - Ability to understand what you are saying
- at some **task**
  - Voice recognition
- through **experience**
  - Transcribed speech data

-- Prof. Tom Mitchell, *CMU*

*"Machine Learning is the **second best** solution to any problem. The **first best** is of course to **solve the problem** directly."*

-- Prof. Yaser S. Abu-Mostafa, *Caltech*

How would you write a program to recognize human speech?

You use ML every day!

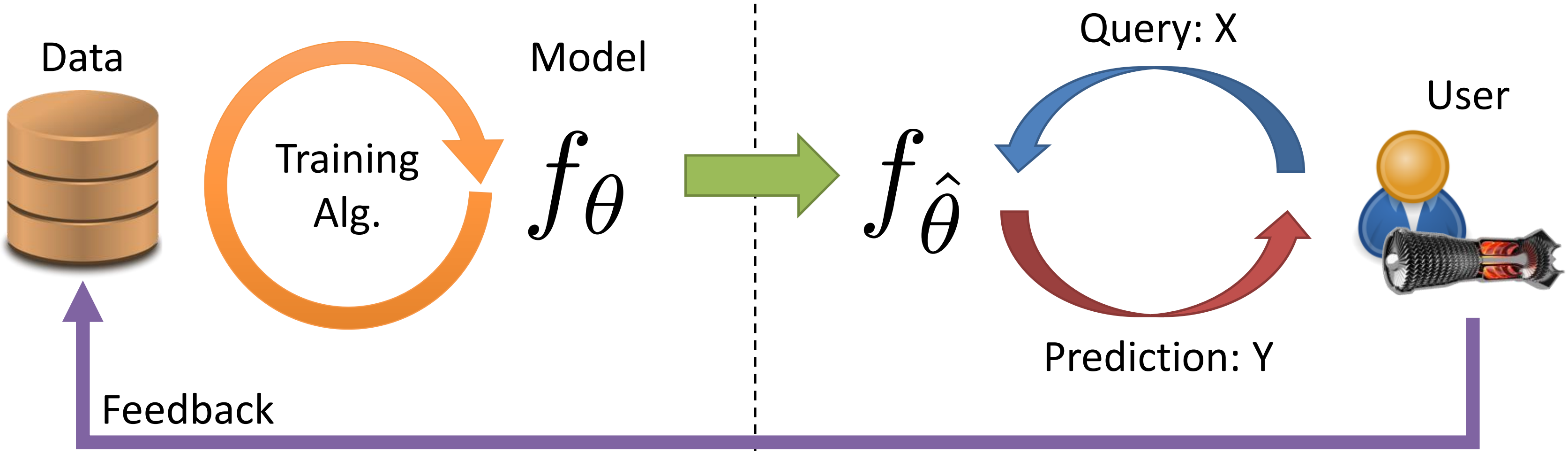
What machine learning do you use every day?

- Spam detection
- Voice recognition
- Face tagging on Facebook
- Ad Targeting
- Credit card fraud detection
- Others? …

Machine Learning Lifecycle

Learning

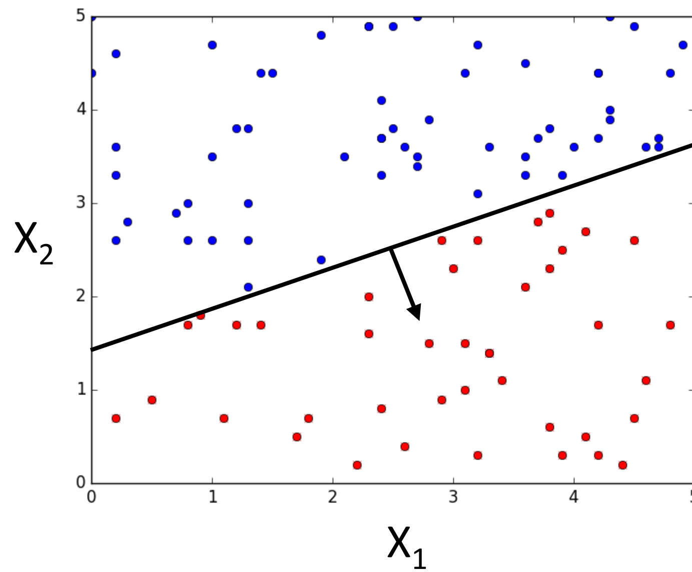
Inference



- Typically a time consuming iterative batch process
  - Feature engineering
  - Validation
- Focus is on making fast robust predictions
  - Monitoring and tracking feedback
  - Materialization + fast model inference

Learning: *Fitting the Model*

- **Training Data**
  - **X**: Features
  - **Y**: Label/Obs.
- Learn a function that **generalizes** the relationship between *X* and *Y*



X		Y
X_1	X_2	
3.9	3.1	T
0.2	0.1	F
4.2	0.9	F
1.1	2.9	T

Features

Function class / Model Family

$$f_{\theta}(X) \rightarrow Y$$

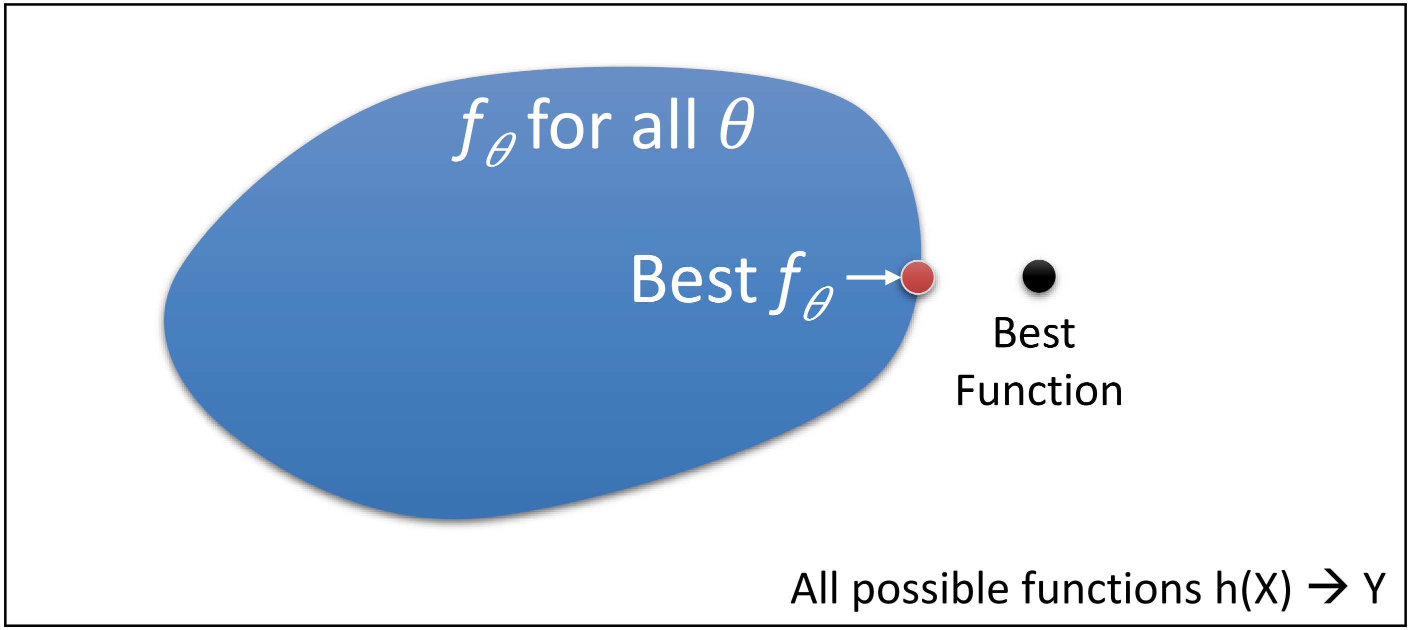
Labels / Observations

Model Parameters

Finding the Best Parameters

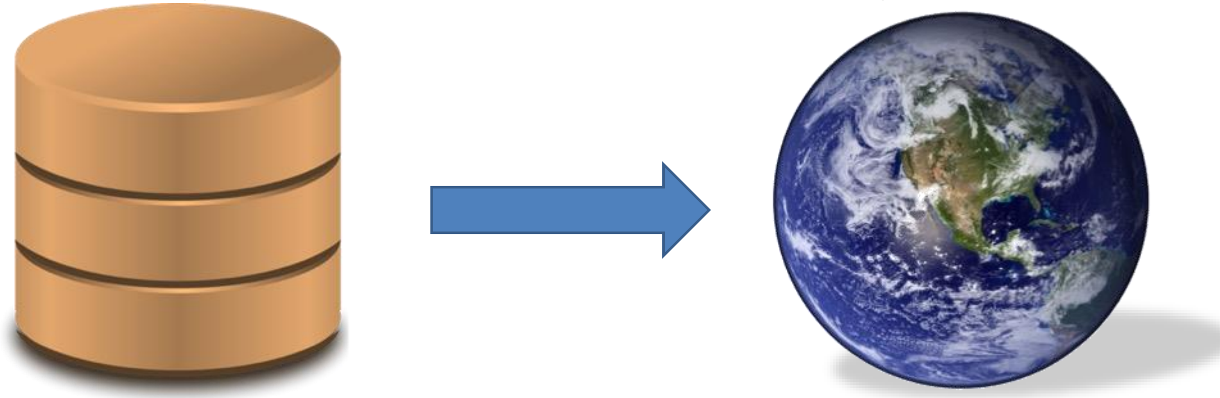
$$f_\theta(X) \rightarrow Y$$

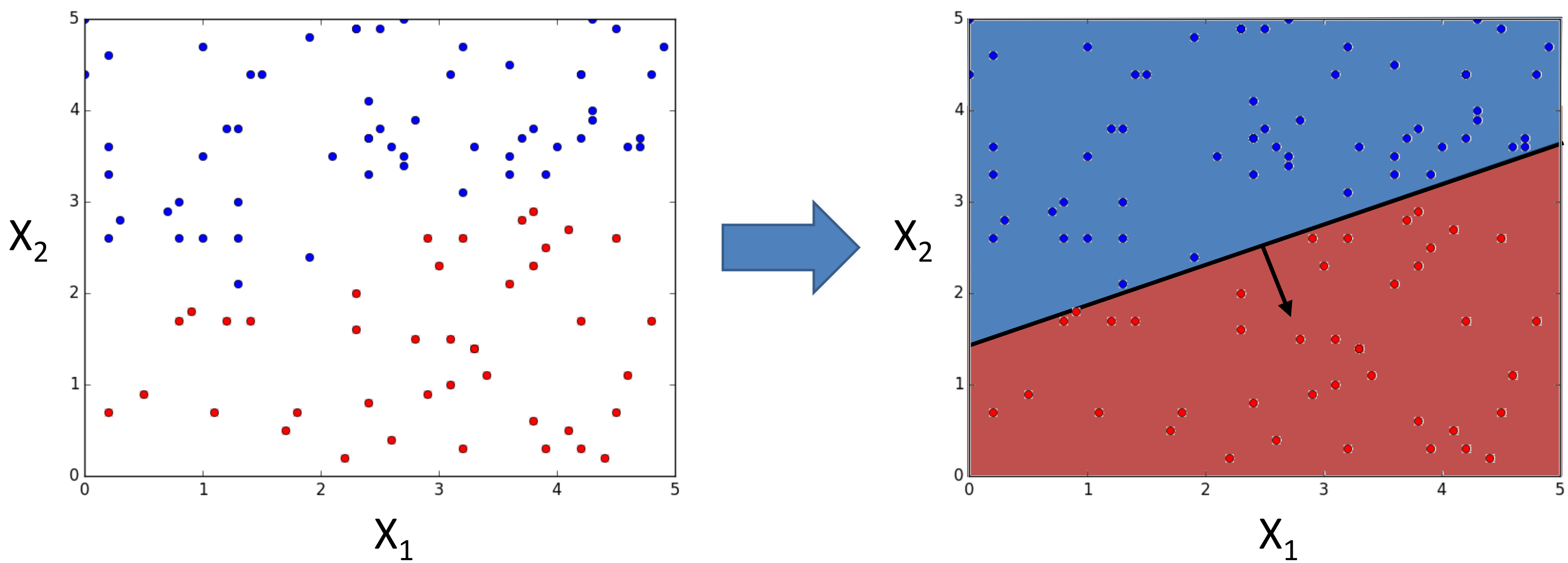
- Define some **objective** (e.g., prediction error)
- Search for best θ with respect to the objective



Generalization …

Sample Population





Inference: *Rendering Predictions*

- Evaluating the model on input queries:

$$f_{\hat{\theta}}(X) \to Y$$

- Online vs Offline:
  - Pre-computed **offline**: *movie rankings*
  - Computed **online** with each query: *speech recognition*
- May want to track confidence in prediction
- May require additional pre and post-processing
  - Feature lookup, content ranking, etc…

Feedback: *Incorporating New Data*

- After rendering a prediction we may get feedback on the results of the prediction:
  - **Explicit**: the *correct value* was "cat"
  - **Implicit**: the predicted animal was *incorrect*
  - Can be **noisy** …

- Watch out for **sample bias:**
  - Model affects the data is uses for training in the future
  - **Example**: only play top40 songs ...

Taxonomy
of Machine Learning

Labeled Data

Indirect (reward)

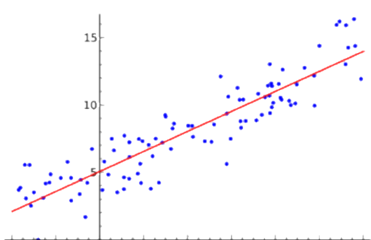
Unlabeled Data

Supervised Learning

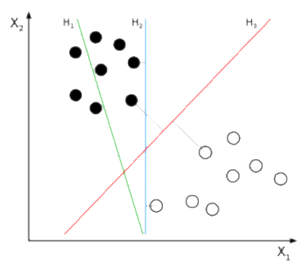
Reinforcement & Bandit Learning

Unsupervised Learning

Regression

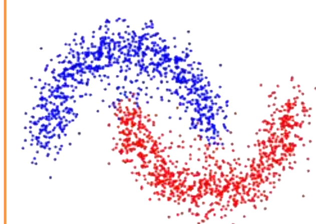


Classification



Dimensionality Reduction

Clustering



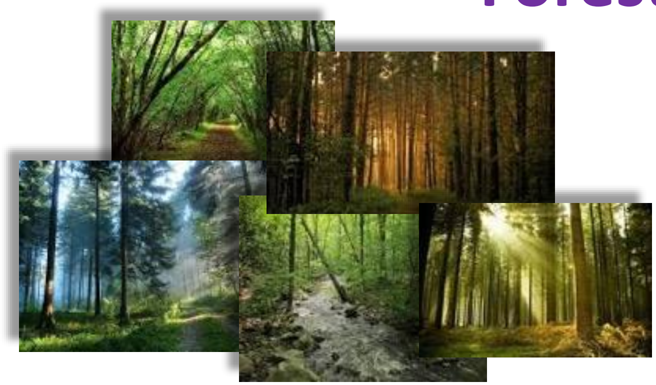
Clustering Images

➢ Given a collection of images cluster them into *meaningful groups*.

Clustering Images

➢ Given a collection of images cluster them into meaningful groups.

"Forest"



"Mountains"

"Beaches"

Clustering Images

- Given a collection of images cluster them into meaningful groups.

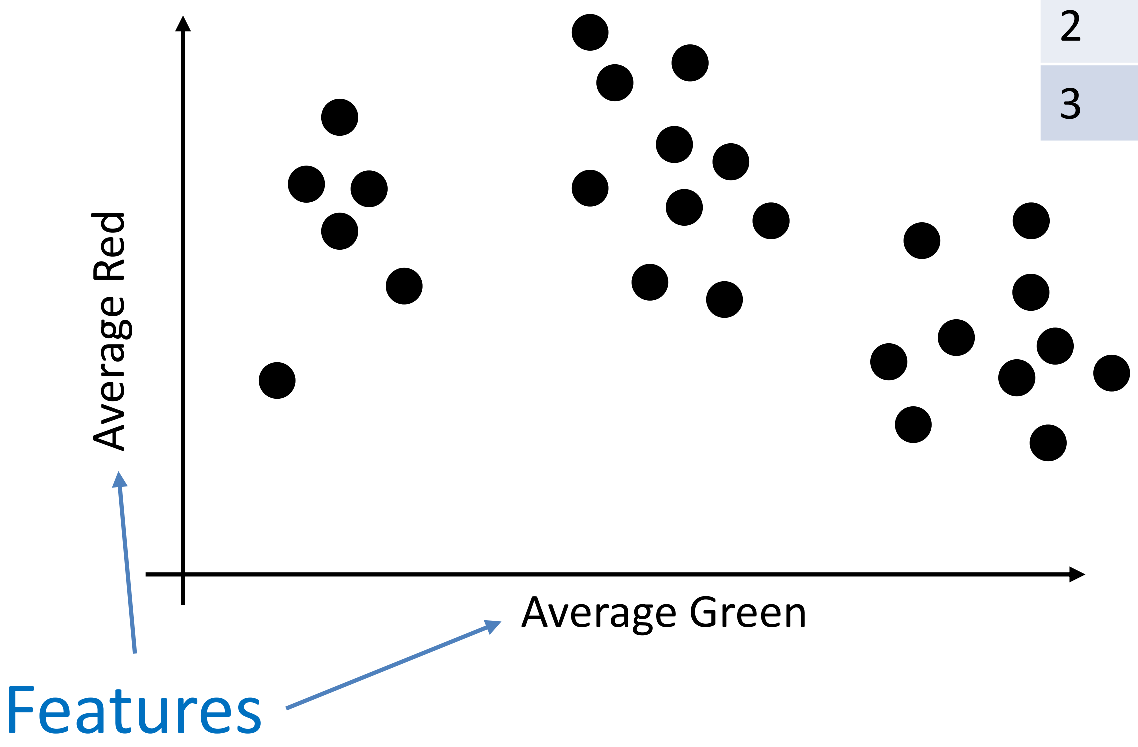
- **Unsupervised:** The labels of the groups are not given in the training data
- **Exploratory:** overlaps with data mining

Clustering Images

- Given a collection of images cluster them into meaningful groups.

Image Id	Average Red	Average Green
1	123	200
2	212	103
3	55	35

Simplified Illustration



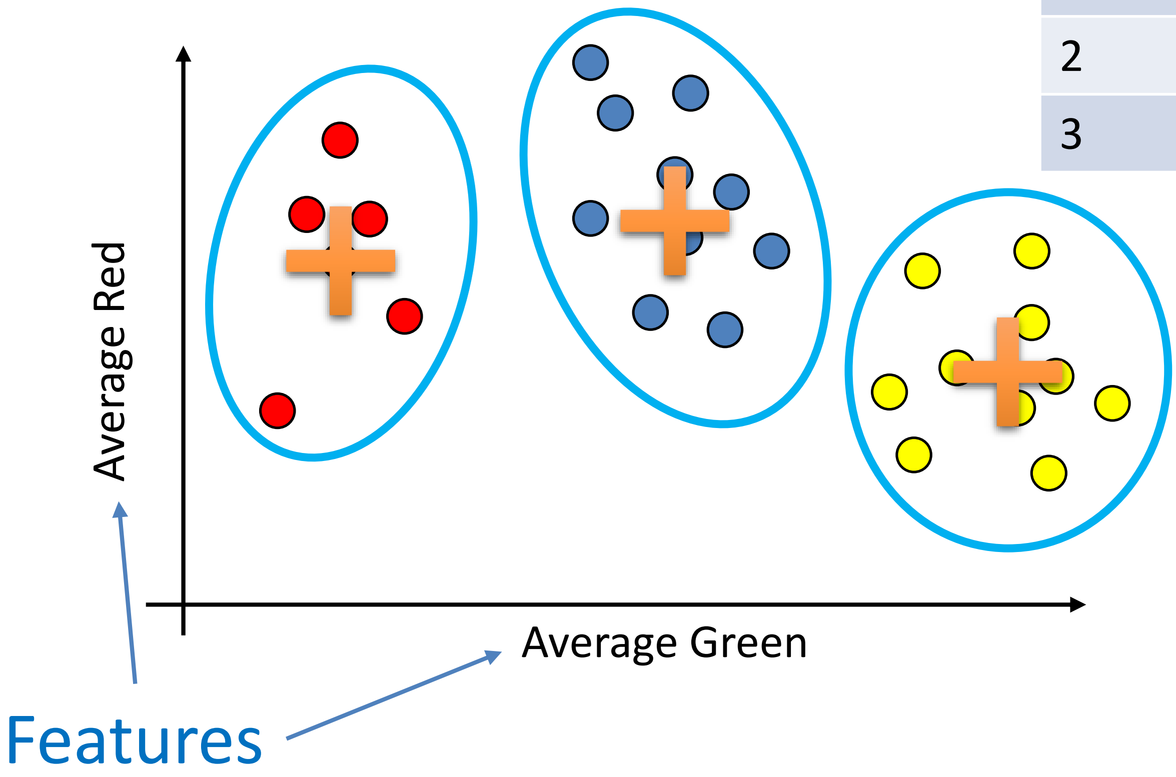
- How many clusters?
- Where are the clusters?

Clustering Images

- Given a collection of images cluster them into meaningful groups.

Image Id	Average Red	Average Green
1	123	200
2	212	103
3	55	35

Simplified Illustration

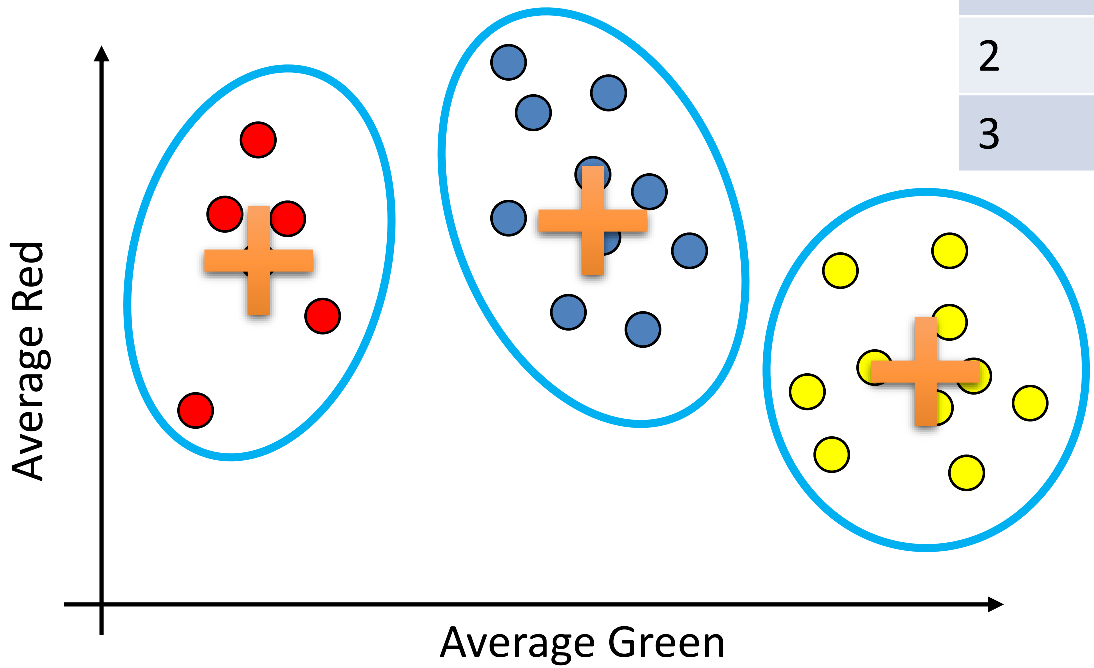


- Where are the clusters?
- How many clusters?

Clustering Images

- Given a collection of images cluster them into meaningful groups.

Image Id	Average Red	Average Green
1	123	200
2	212	103
3	55	35



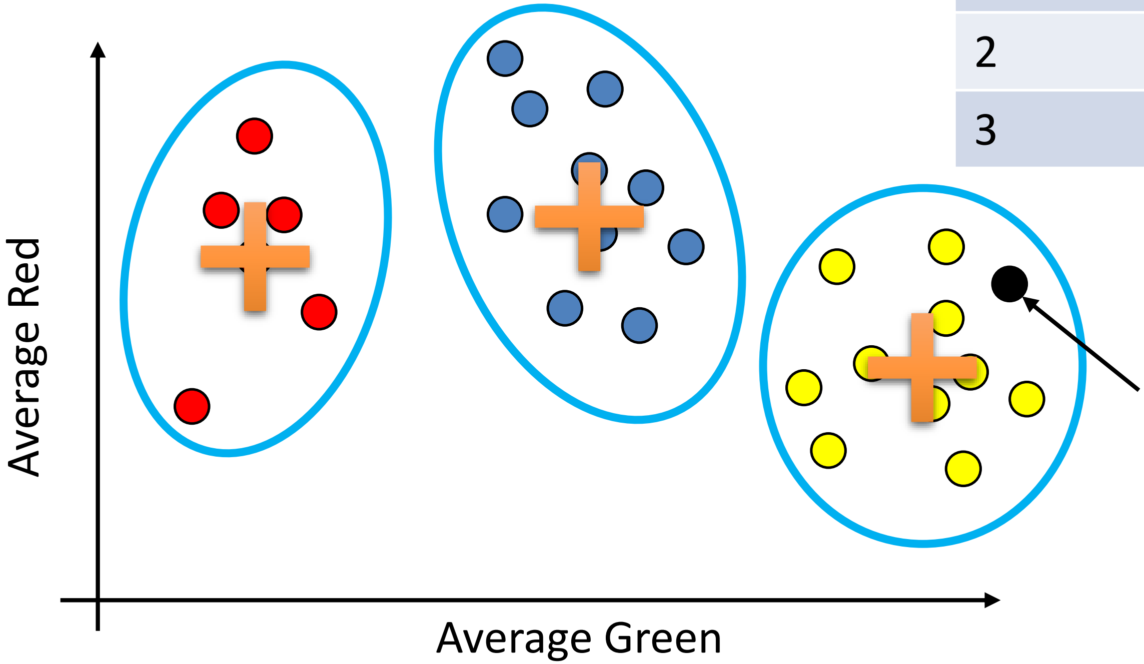
What makes a good clustering?

- All points are near the cluster center
- Spread between clusters > spread within clusters

Clustering Images

- Given a collection of images cluster them into meaningful groups.

Image Id	Average Red	Average Green
1	123	200
2	212	103
3	55	35

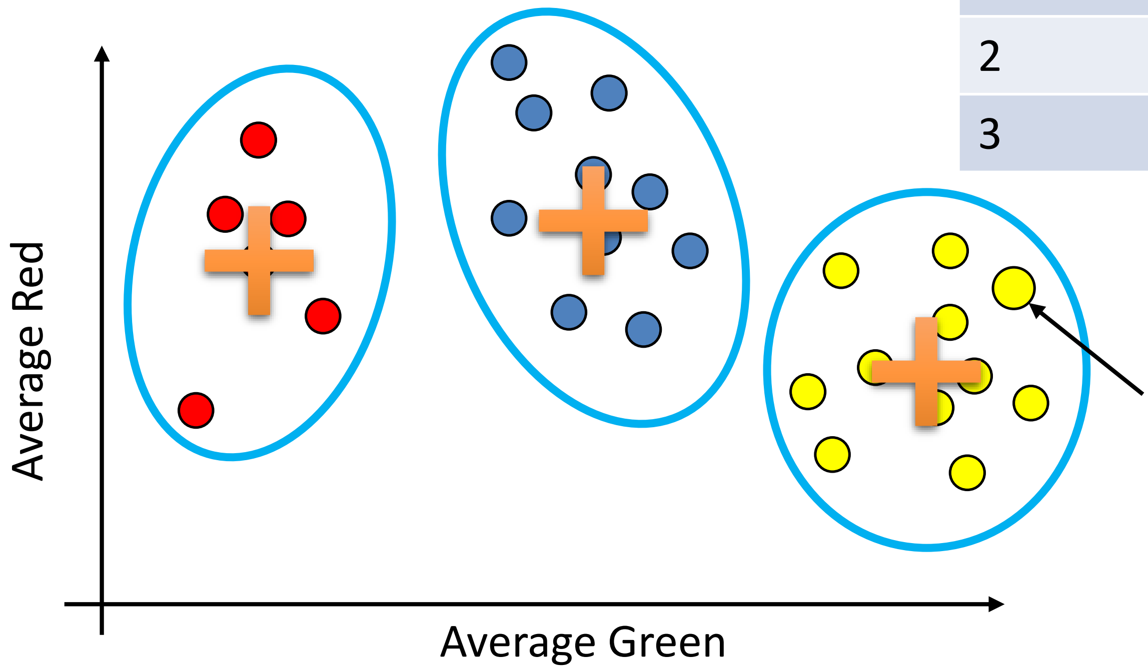


What happens when a new point arrives?

Clustering Images

- Given a collection of images cluster them into meaningful groups.

Image Id	Average Red	Average Green
1	123	200
2	212	103
3	55	35



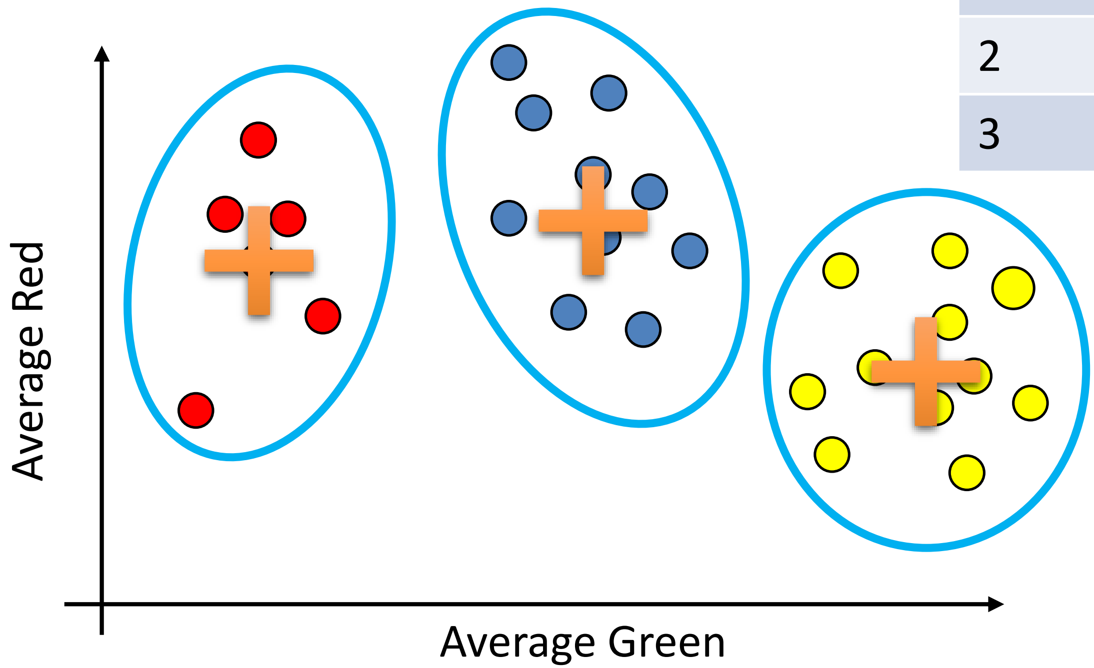
What happens when a new point arrives?

Predict "label" based on existing clusters (Yellow)

Clustering Images

- Given a collection of images cluster them into meaningful groups.

Image Id	Average Red	Average Green
1	123	200
2	212	103
3	55	35



How do we automatically cluster data?

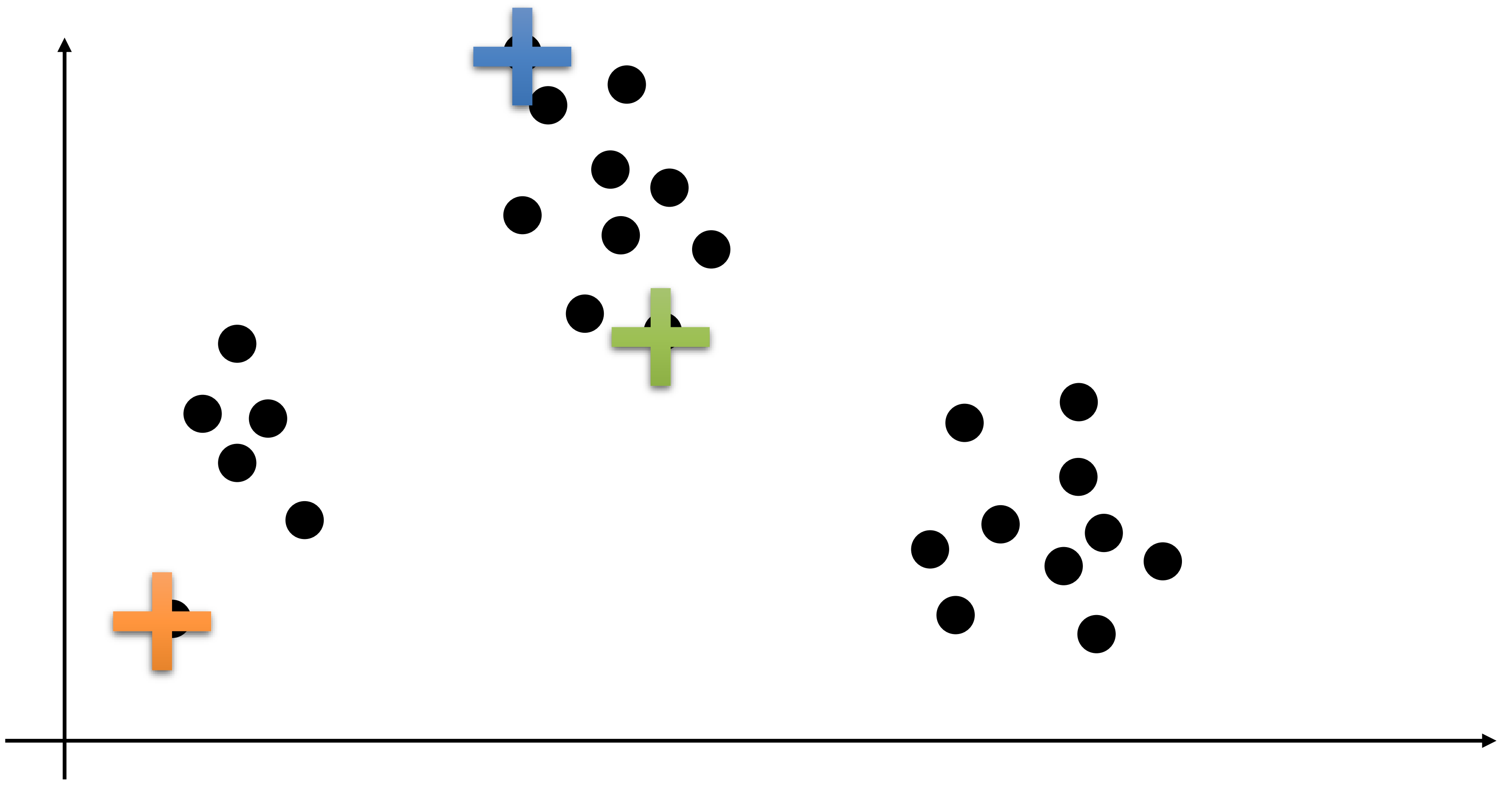
How do we Compute a Clustering?

Many different clustering models and algorithms:

- Feature Based Clustering: *Points in R^d*
  - **K-Means:** EM on Symmetric Gaussians ← We will learn this one
  - **Mixture Models:** Generalized k-means
  - …
- Spectral Methods: *Similarity Function Between Items*
  - **Similarity based clustering:** *A and B are co-purchased*
  - **Graph clustering:** *Cities based on road network*
  - *…*
- Hierarchical Clustering: *clustering nested items*
  - **Latent Dirichlet Allocation:** *Documents based on words*
    - *Developed at Berkeley and widely used!*
  - *…*

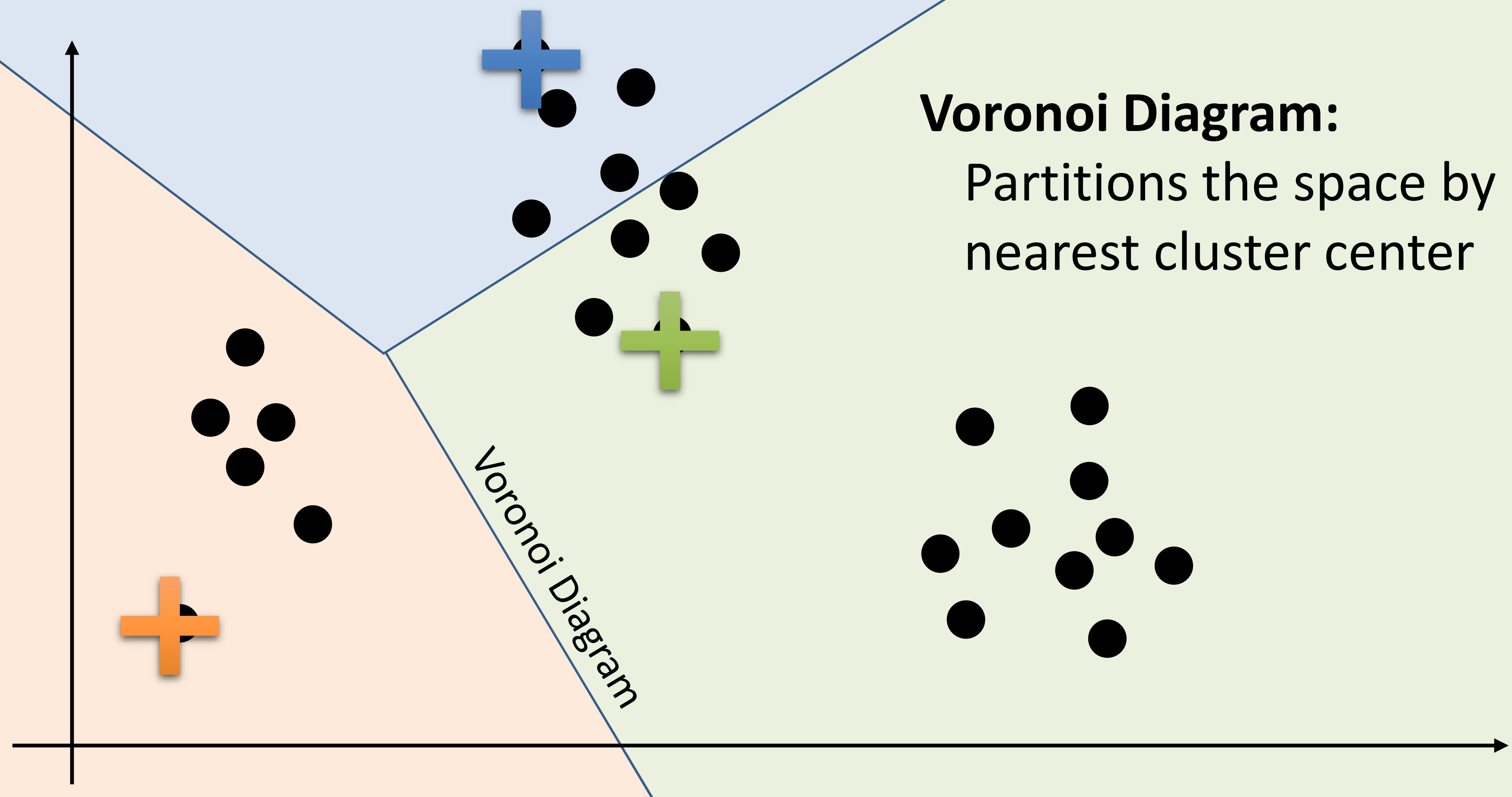
K-Means Clustering: *Intuition*

- Input K: The number of clusters to find
- Pick an initial set of points as cluster centers



K-Means Clustering: *Intuition*

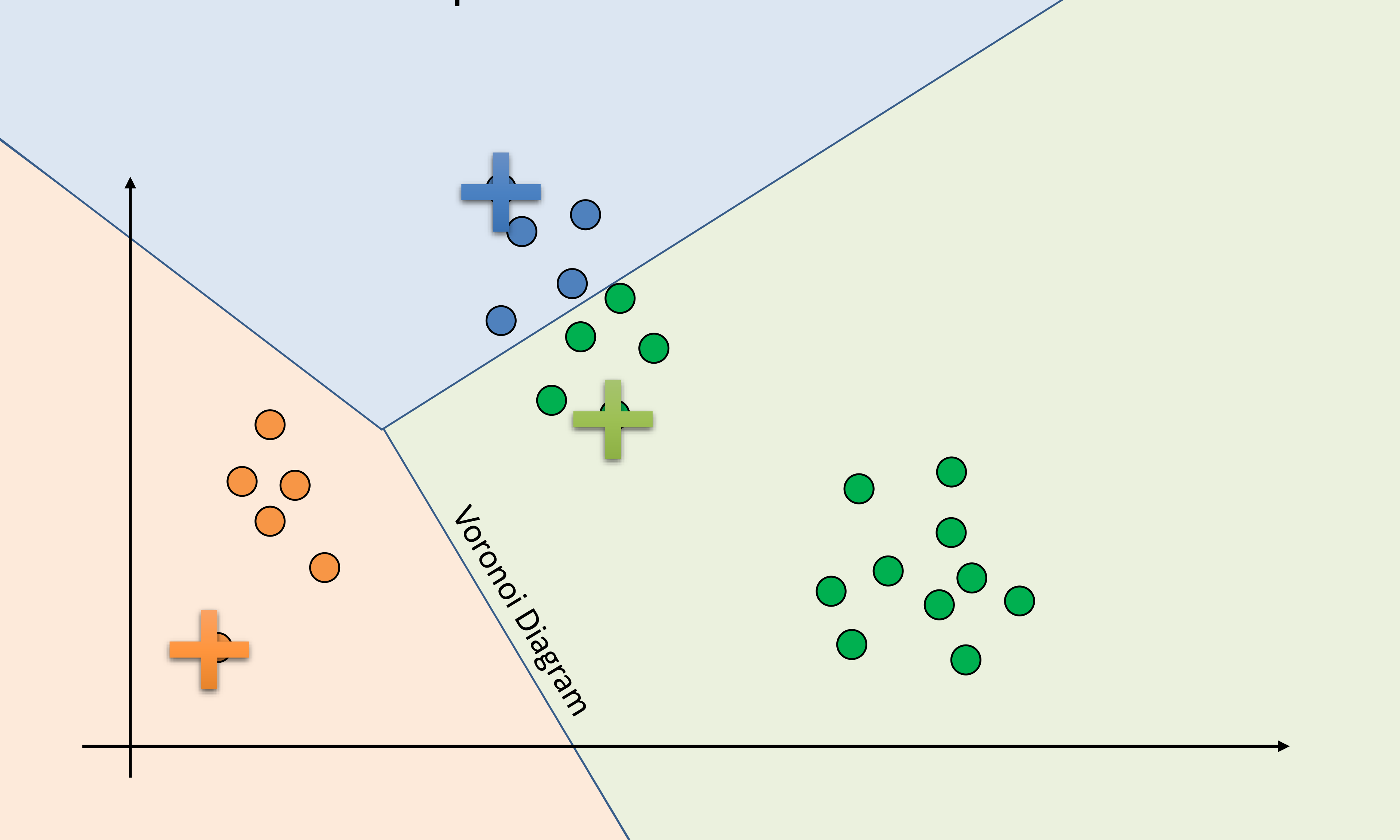
- For each data point find the cluster nearest center



Voronoi Diagram:
Partitions the space by nearest cluster center

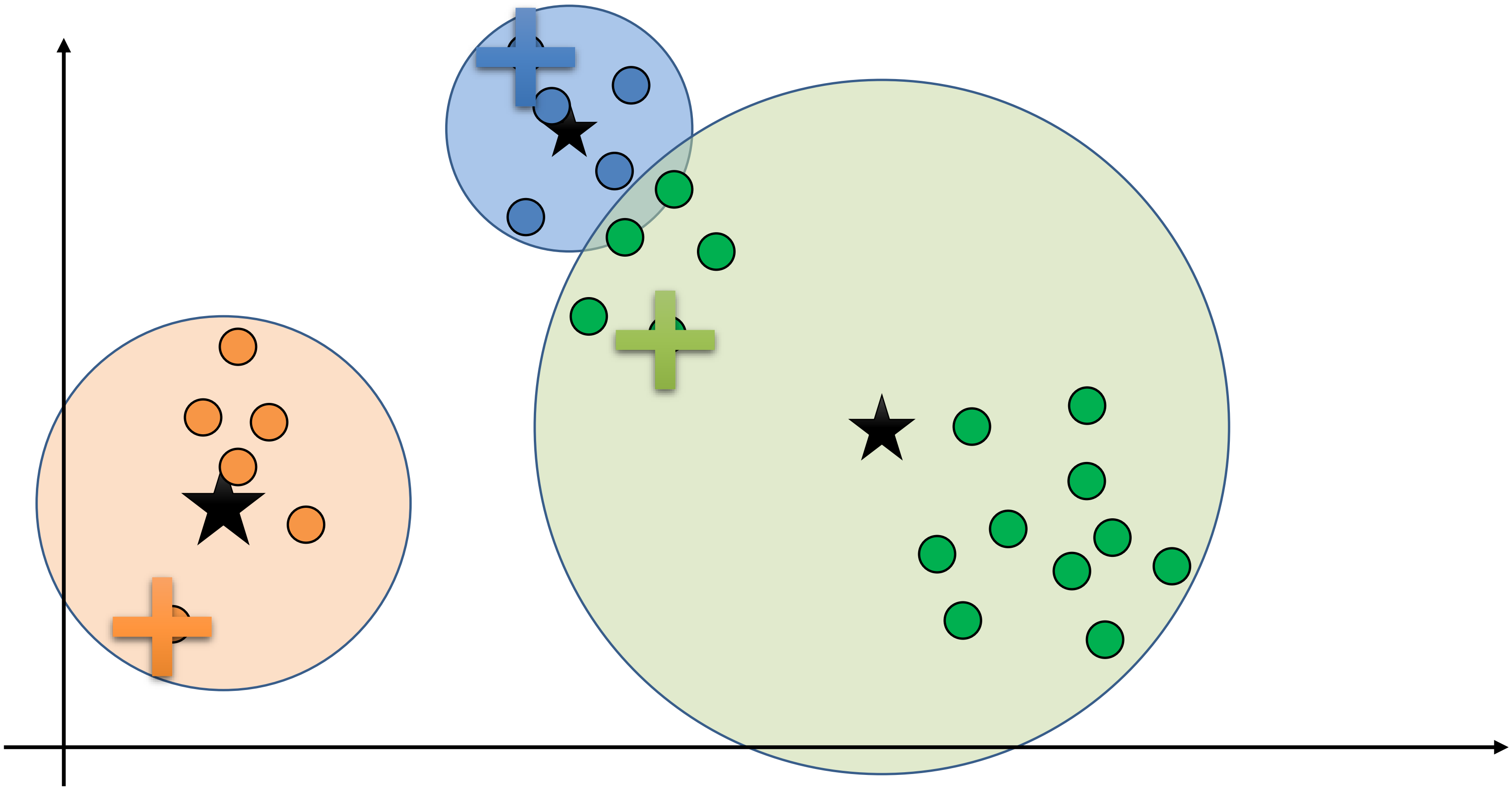
K-Means Clustering: *Intuition*

- For each data point find the cluster nearest center



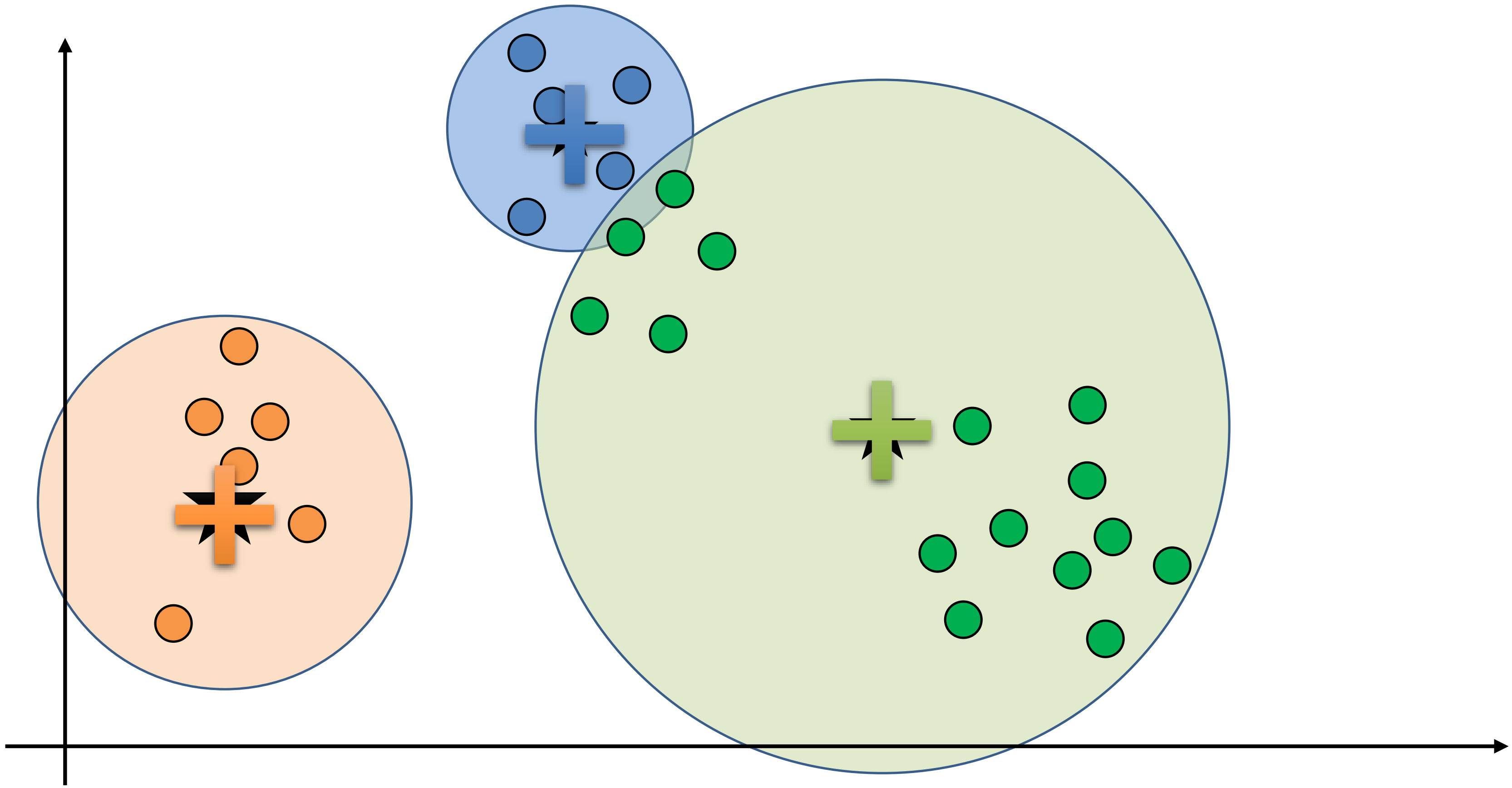
K-Means Clustering: *Intuition*

- Compute mean of points in each “cluster”



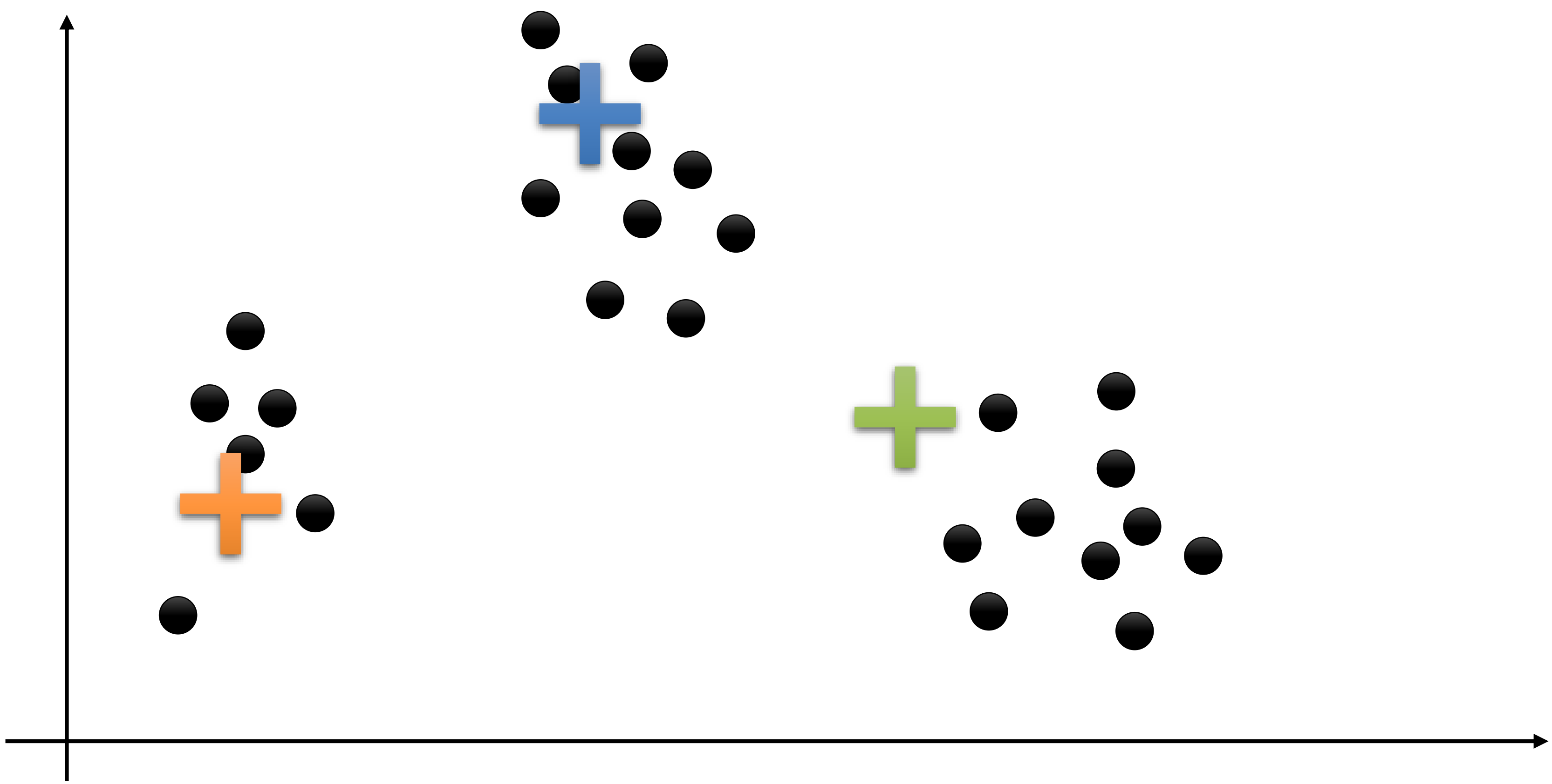
K-Means Clustering: *Intuition*

- Adjust cluster centers to be the mean of the cluster



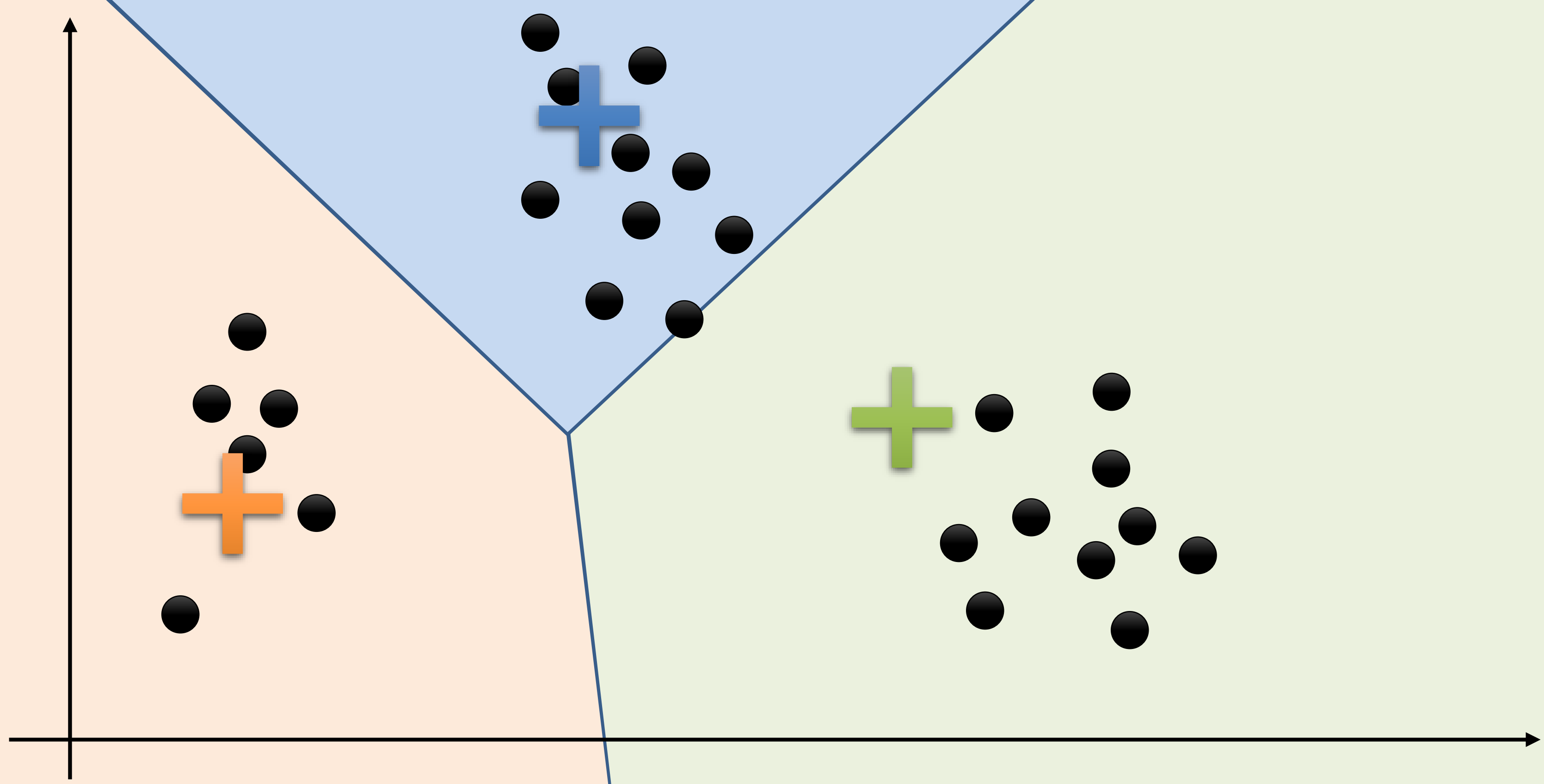
K-Means Clustering: *Intuition*

- Improved?
- Repeat



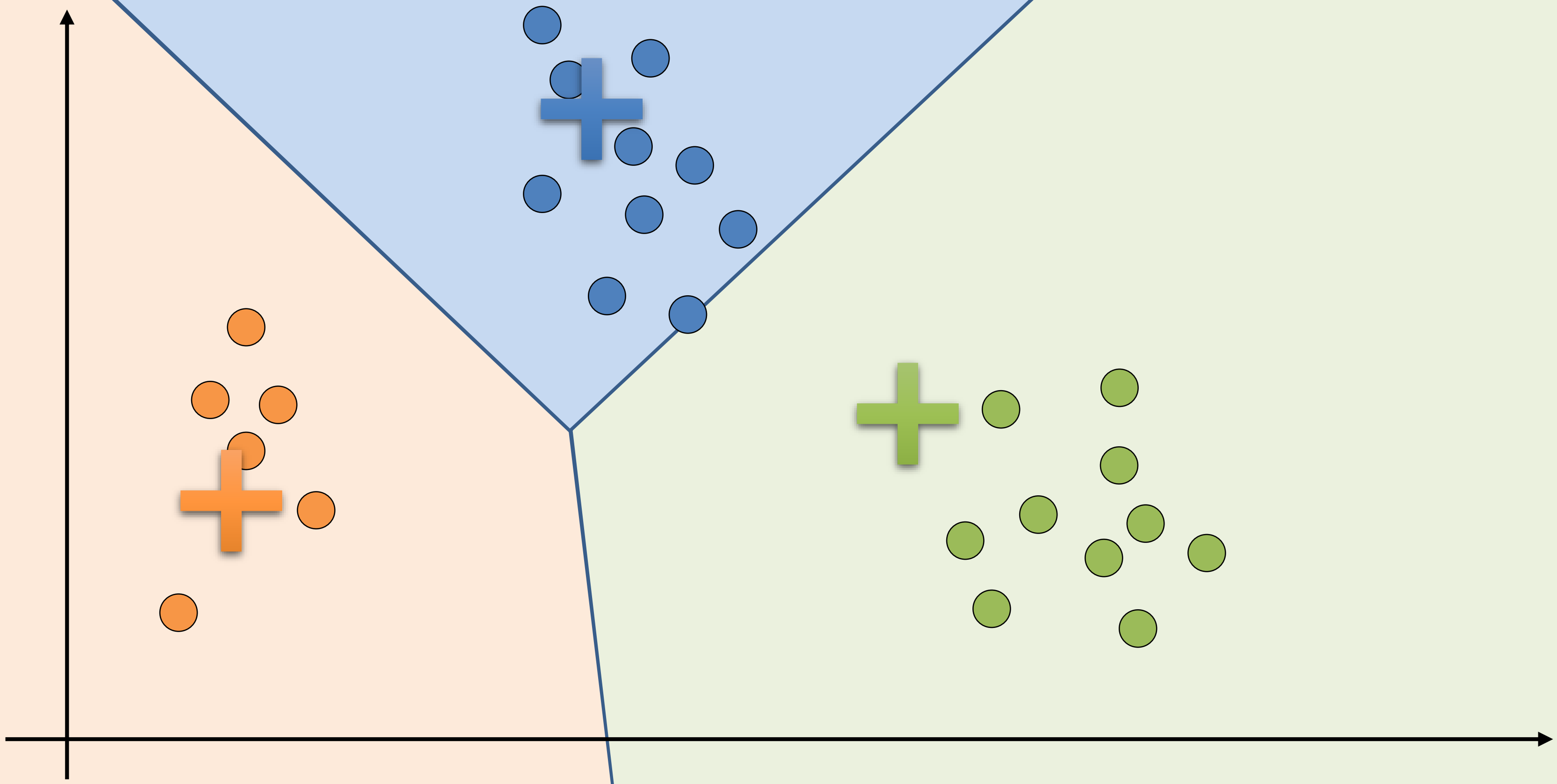
K-Means Clustering: *Intuition*

- Assign Points



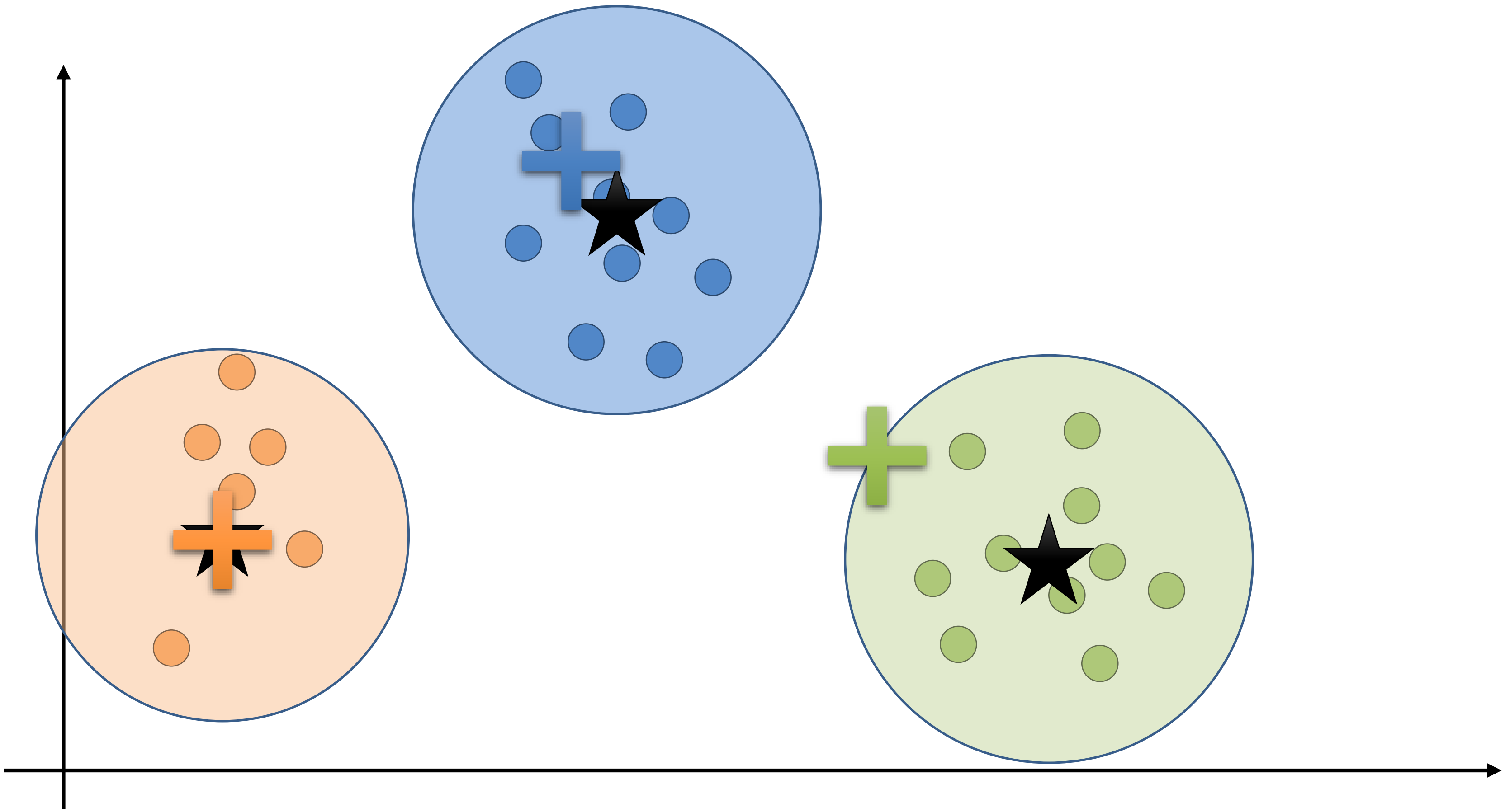
K-Means Clustering: *Intuition*

- Assign Points



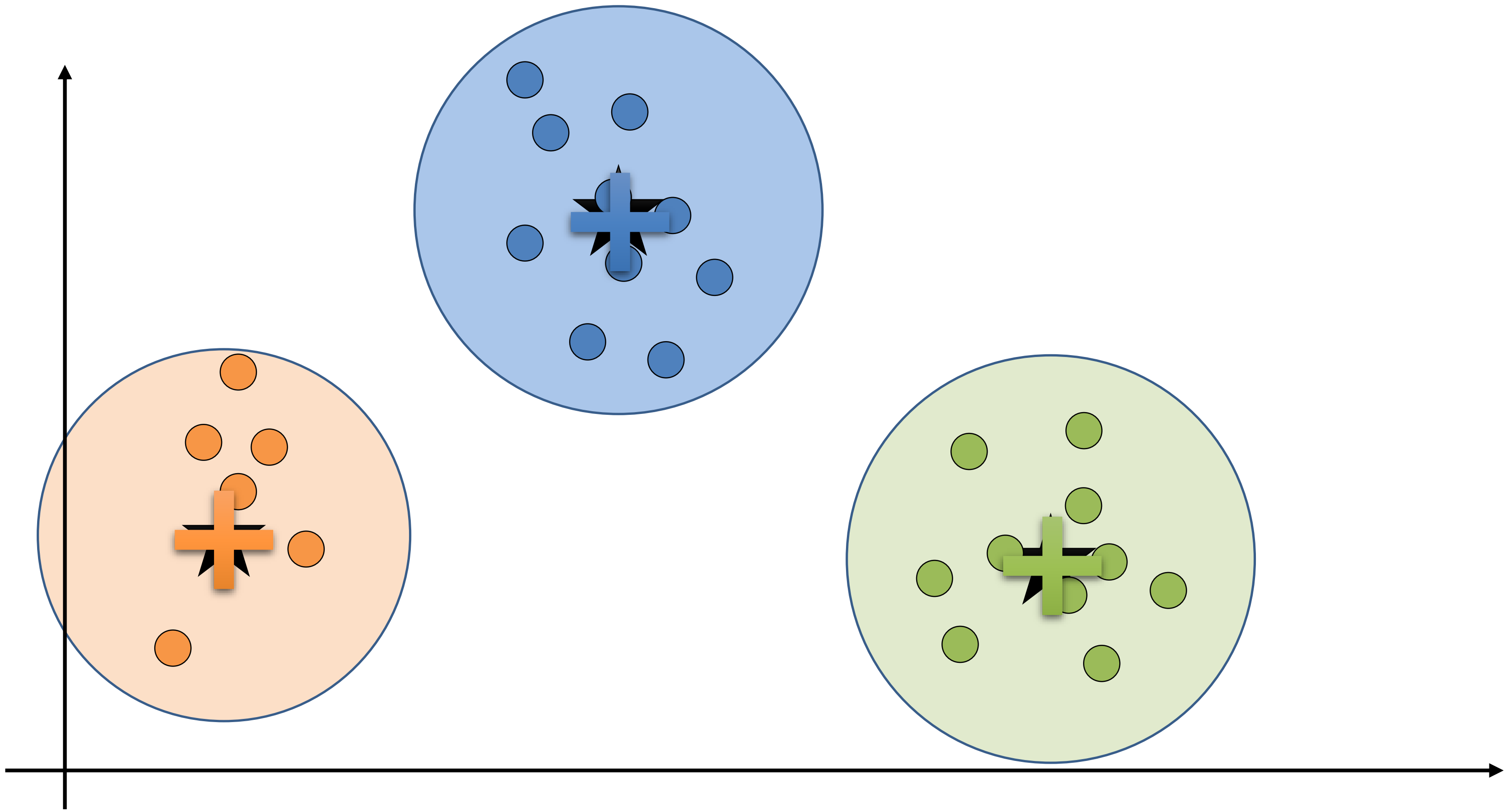
K-Means Clustering: *Intuition*

- Compute cluster means



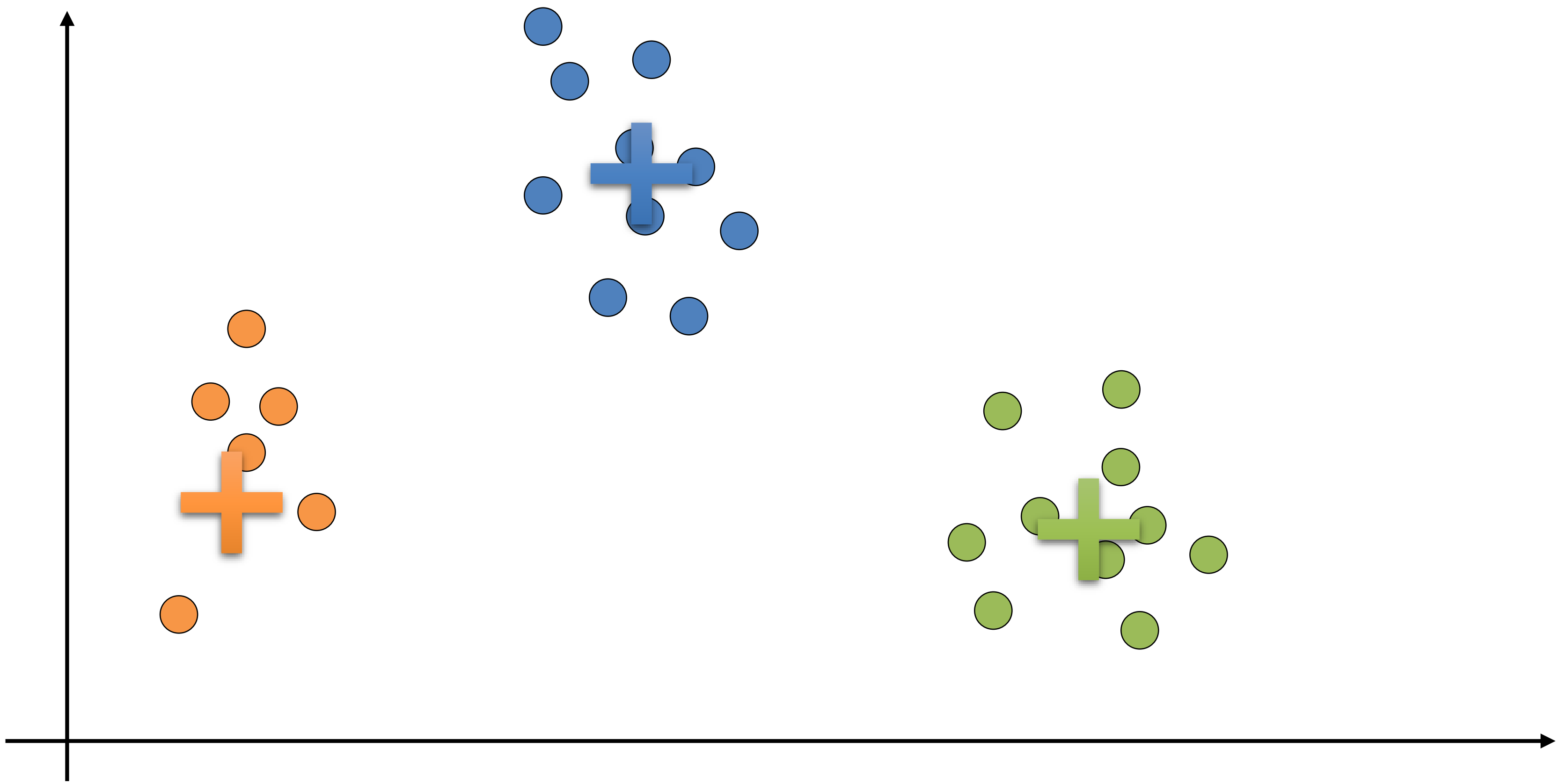
K-Means Clustering: *Intuition*

- Update cluster centers



K-Means Clustering: *Intuition*

- Repeat?
  - Yes to check that nothing changes → Converged!



K-Means Algorithm: **Details**

```
centers ← pick k initial Centers

while (centers are changing) {
   // Compute the assignments (E-Step)
   asg ← [(x, nearest(centers, x)) for x in data]
```

What do we mean by "nearest":

A: Euclidean Distance

$$\arg\min_{c \in \text{centers}} ||c - x||_2^2 = \sum_{i=1}^{d} (c_i - x_i)^2$$

K-Means Algorithm: **Details**

```
centers ← pick k initial Centers

while (centers are changing) {
   // Compute the assignments (E-Step)
   asg ← [(x, nearest(centers, x)) for x in data]

   // Compute the new centers (M-Step)
   for i in range(k):
      centers[i] =
         mean([x for (x, c) in asg if c == i])
}
```

Compute the “Expected” Assignment

Find centers that maximize the data “likelihood”

K-Means Algorithm: **Details**

```
centers ← pick k initial Centers

while (centers are changing) {
   // Compute the assignments (E-Step)
   asg ← [(x, nearest(centers, x)) for x in data]

   // Compute the new centers (M-Step)
   for i in range(k):
      centers[i] =
         mean([x for (x, c) in asg if c == i])
}
```

Guaranteed to converge!

… to what?

To a local optimum. ☹

Depends on Initial Centers

K-Means Algorithm: **Details**

```
centers ← pick k initial Centers
```

How do we pick initial centers?

```
while (centers are changing) {
  // Compute the assignments (E-Step)
  asg ← [(x, nearest(centers, x)) for x in data]

  // Compute the new centers (M-Step)
  for i in range(k):
    centers[i] =
      mean([x for (x, c) in asg if c == i])
}
```

Guaranteed to converge!

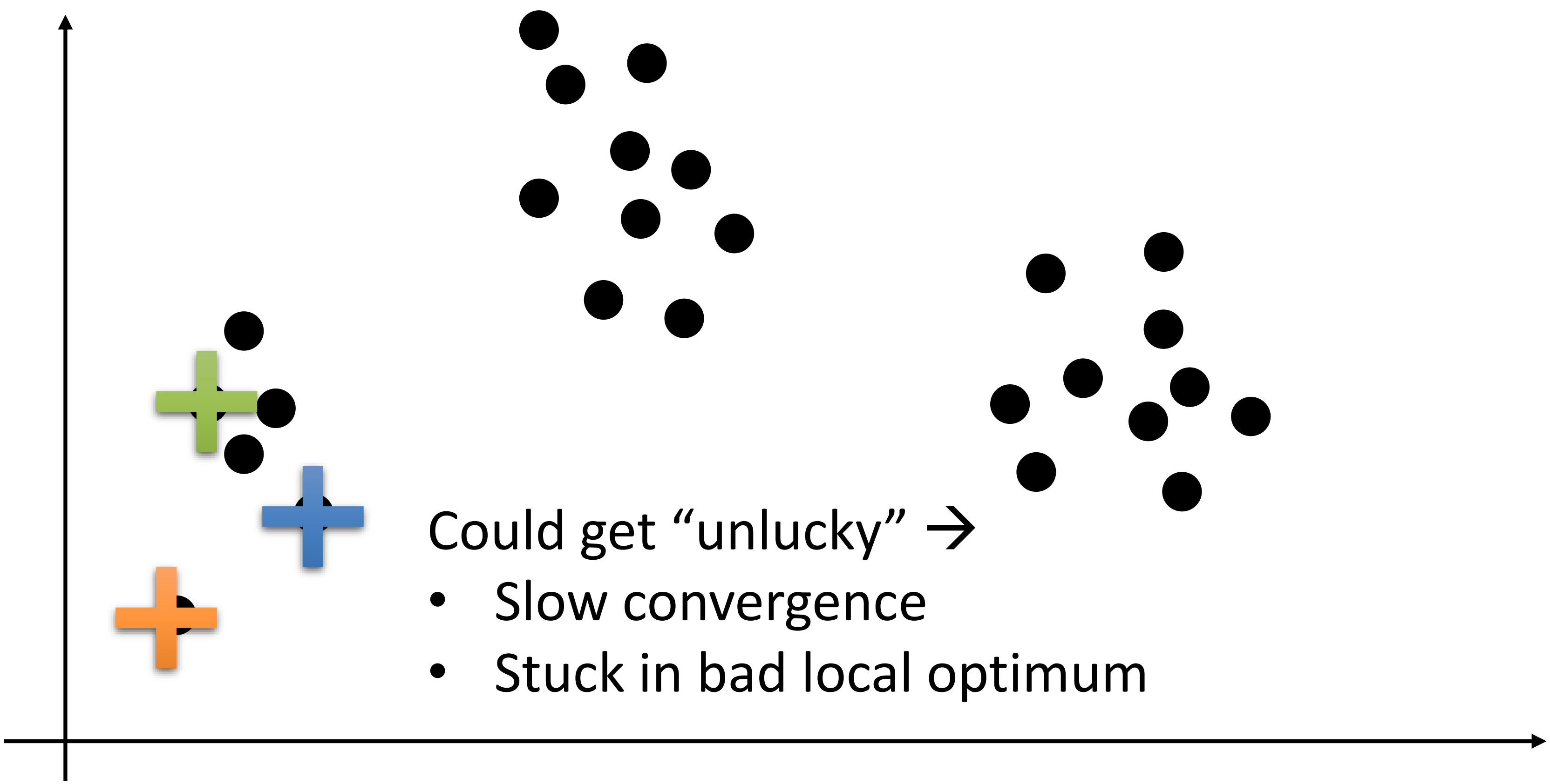
... to what?

To a local optimum. ☹

Depends on Initial Centers

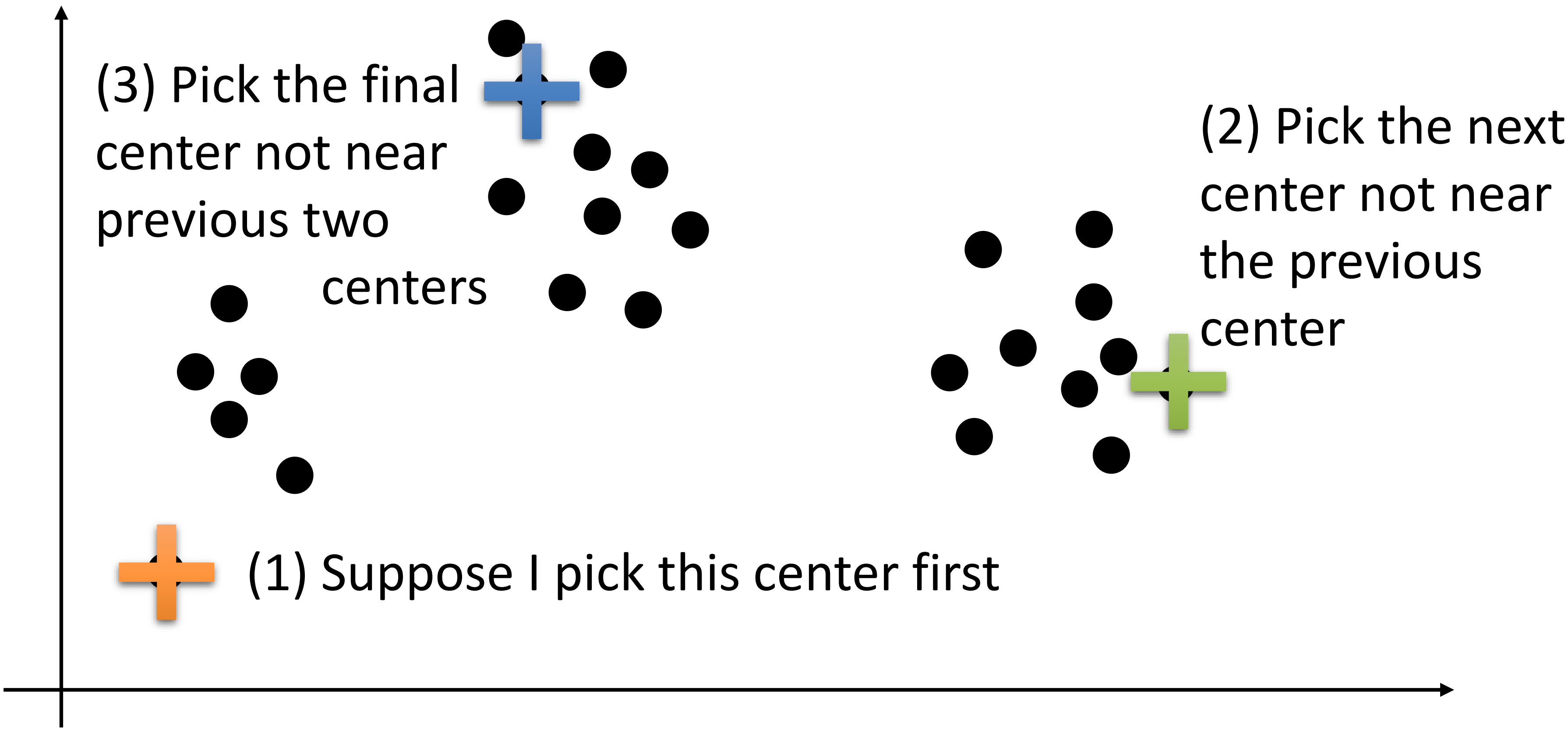
Picking the Initial Centers

- **Simple Strategy:** select *k* points at random
  - What could go wrong?



Picking the Initial Centers

- **Better Strategy:** kmeans++
  - Randomized approx. algorithm
  - Intuition select points that are not near existing centers

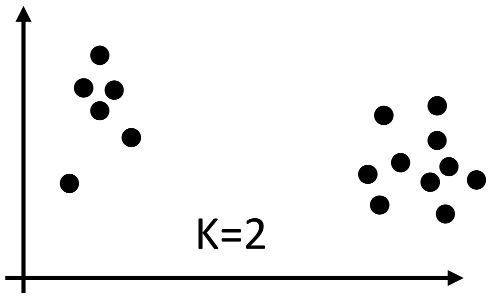


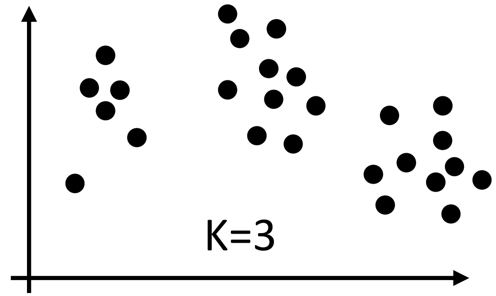
K-Means++ Algorithm

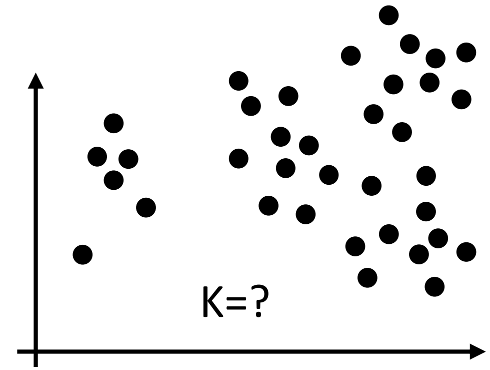
```
centers ← set(randomly select a single point)

while len(centers) < k:
  # Compute the distance of each point
  # to its nearest center dSq = d^2
  dSq ← [(x, dist_to_nearest(centers, x)^2) for x in data]
  # Sample a new point with probability
  # proportional to dSq
  c ← sample_one(data, prob = dSq / sum(dSq))
  # Update the clusters
  centers.add(c)
```

http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf

How do we choose K?







- Basic Elbow Method (Easy and what you do in HW)
  - Try range of K-values and plot average distance to centers
- Cross-Validation (Better)
  - Repeatedly split the data into training and validation datasets
  - Cluster the training dataset
  - Measure Avg. Dist. To Centers on validation data