Fall 2024

BECE309L – AI & ML

Data Preparation for Machine Learning

Module – 5

Dr. Nitish Katal

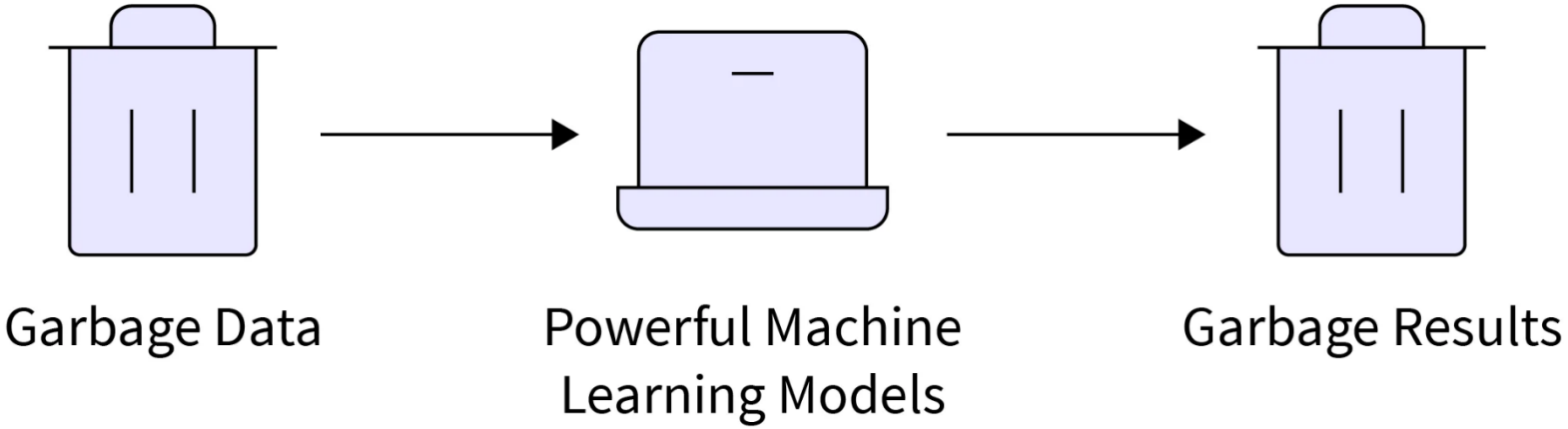
Data Preparation for Machine Learning

- Basics of Vectors and Matrices
- Data and Related Definitions
- Data Cleaning
- Data Integration
- Data Transformation
- Data Reduction

Data Preparation for Machine Learning

- Data preprocessing phase is the most challenging and time-consuming part of data science, but it's also one of the **most important parts**.

- If you fail to clean and prepare the data, **it could compromise the model.**



Data Preparation for Machine Learning

Data Collection

- Articulating the problem
- Defining data required
- Gathering and combining data from different sources

→

Data Preprocessing

- Formatting
- Cleaning
- Sampling

→

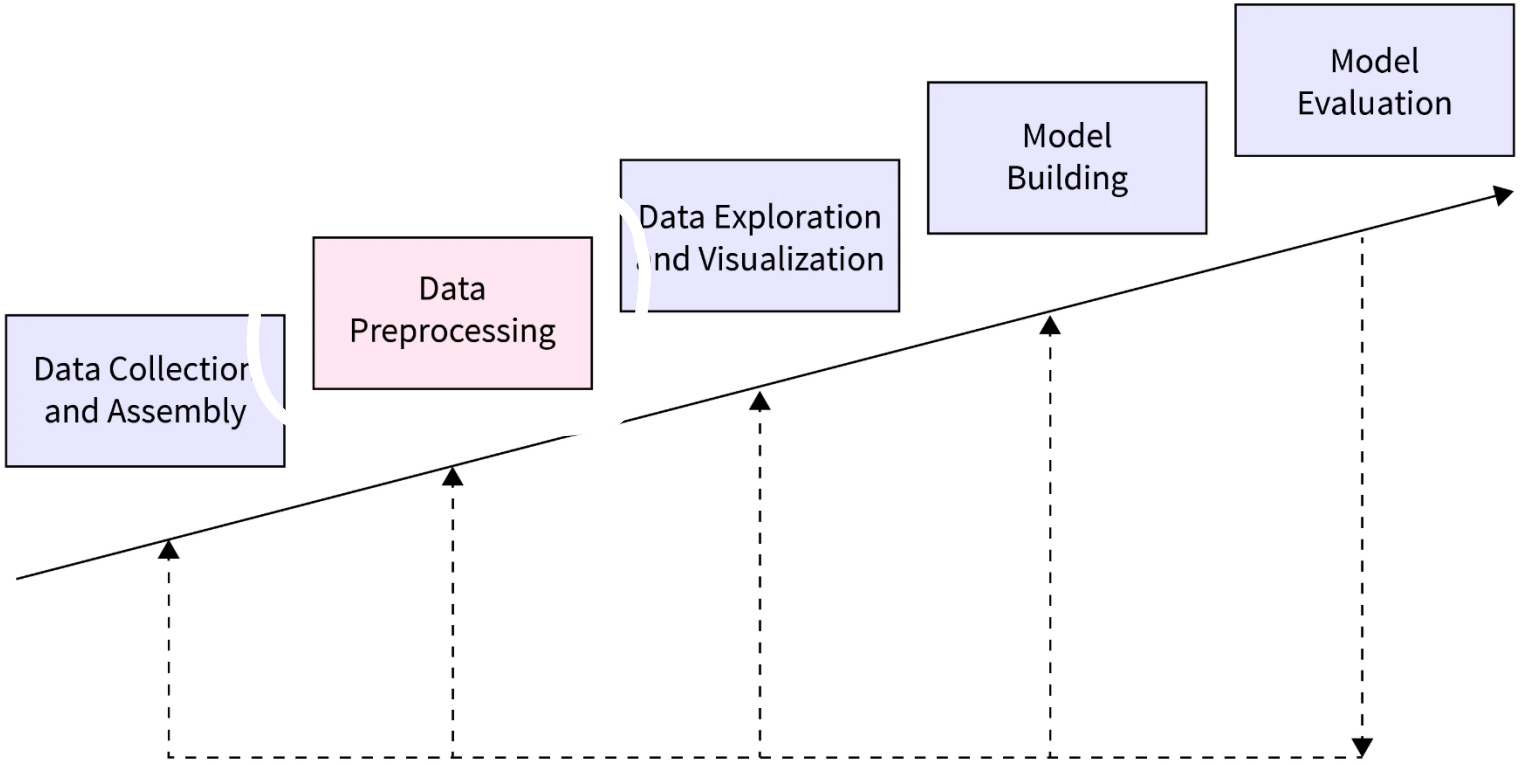
Data Transformation

Feature engineering

- Scaling/normalizing
- Decomposition
- Aggregation

What is Data Preprocessing?

- **Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model.**
- Whenever the data is gathered from different sources it is collected in raw format which is *not feasible for the analysis*.



Need for Data Preprocessing?

- Data in the real world is dirty
  - *incomplete*: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
  - *noisy*: containing errors or outliers
  - *inconsistent*: containing discrepancies in codes or names

- No quality data, no quality results!
  - Quality decisions must be based on quality data
  - Data needs consistent integration of quality data

- For Better Results in ML : Proper Data Formatting must be Ensured.
  - *Example:* Random Forest algorithm does not support null values,
  - Another aspect: Dataset should be formatted in such a way that more than one ML/DL algorithm are executed.

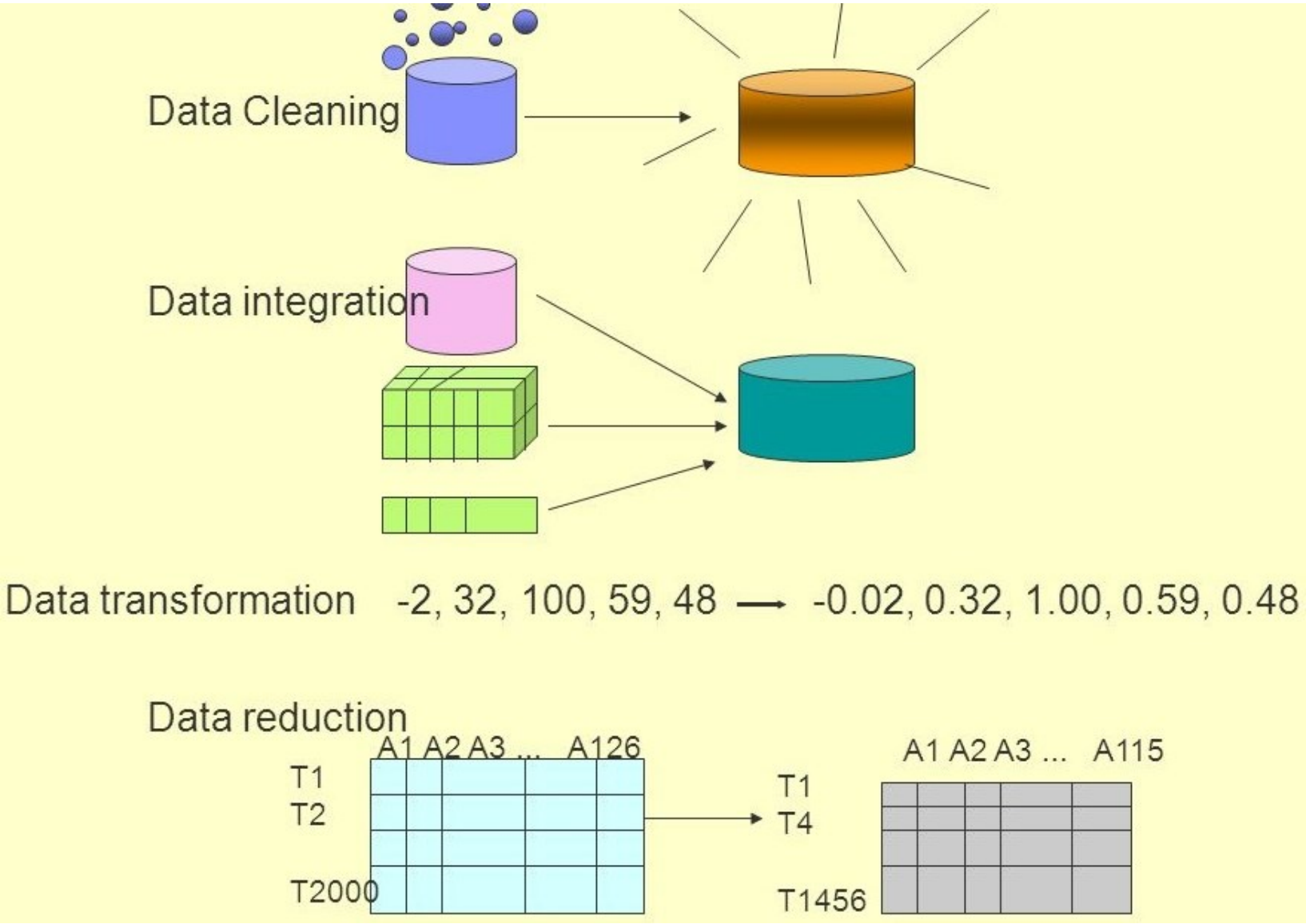
How can Data be Inconsistent

- Faulty instruments for data collection
- Human or computer errors
- Errors in data transmission
- Technology limitations
  - e.g., sensor data come at a faster rate than they can be processed
- Inconsistencies in naming conventions or data codes
  - e.g., 2/5/2002 could be 2 May 2002 or 5 Feb 2002

Major Tasks in Data Pre-processing

- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, or files
- Data transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization
  - Part of data reduction but with particular importance, especially for numerical data

Forms of Data Preprocessing



Forms of Data Preprocessing

Data Preprocessing	Data Cleaning	Dropping Observations	*Missing Data*
		Dropping Variables	
		Fill the Missing Values	
		Regression	*Noisy Data*
		kNN	
		Imputation of Categorical Variables	
	Data Transformations	Label / One Hot Encoding	
		Feature Scaling	Min-Max Normalization
			Standardization
		Attribute Selection	
	Data Reduction	Data Cube Aggregation	
		Numerosity Reduction	
		Dimensionality Reduction	

Basics of Vector and Matrices

Vector Modulus & Inner Product

- A vector $\boldsymbol{r} = \boldsymbol{r_i}\,\vec{\boldsymbol{i}} + \boldsymbol{r_j}\vec{\boldsymbol{j}} + \ \ldots$ can be represented as:

$$\boldsymbol{r} = \begin{bmatrix} \boldsymbol{r_i} & \boldsymbol{r_j} & \ldots \end{bmatrix}^{\boldsymbol{T}} = \begin{bmatrix} \boldsymbol{r_i} \\ \boldsymbol{r_j} \\ . \\ . \\ . \\ . \end{bmatrix}$$

- Size/ length/ modulus of $\boldsymbol{r}$ is expressed as $|\boldsymbol{r}| = \sqrt{r_i^2 + r_j^2 + \cdots}$
- Inner Product/ Dot Product of $\boldsymbol{r}$ with another vector $\boldsymbol{s} = \boldsymbol{s_i}\,\vec{\boldsymbol{i}} + \boldsymbol{s_j}\vec{\boldsymbol{j}}$ is expressed as:

$$\boldsymbol{r} \cdot \boldsymbol{s} = \boldsymbol{r_i} \times \boldsymbol{s_i} + \boldsymbol{r_j} \times \boldsymbol{s_j} + \cdots = |\boldsymbol{r}||\boldsymbol{s}| \cos \boldsymbol{\theta}$$

  - Where, $\boldsymbol{\theta}$ is angle between *r* and *s*

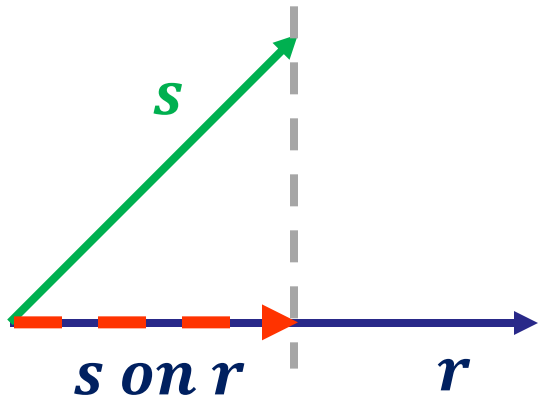
Basics of Vector and Matrices

- Scalar Projection: Of vector $\boldsymbol{s}$ on $\boldsymbol{r}$ can be represented as:

$$\frac{\boldsymbol{r} \cdot \boldsymbol{s}}{|\boldsymbol{r}|}$$

- Vector Projection: Of vector $\boldsymbol{s}$ on $\boldsymbol{r}$ can be represented as:

$$\frac{\boldsymbol{r} \cdot \boldsymbol{s}}{|\boldsymbol{r}|} \times \frac{\boldsymbol{r}}{|\boldsymbol{r}|}$$



- Linear Interdependence
  - A set of vectors $\boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3}$ can be linearly independent if none of them can be written as the linear combination of the other two:
  - Example:

$$\boldsymbol{b_3} \neq \boldsymbol{a_1 b_1} + \boldsymbol{a_2 b_2}$$

Basics of Vector and Matrices

- Matrix Transformation: Scaling and rotation of a matrix $X \in \mathbb{R}^{2\times 2}$ is given by:

$$X_s = X \cdot \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix} \qquad X_r = X \cdot \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

where a_1, a_2 are scaling constants and θ is the angle of rotation.

- Similarly for $X \in \mathbb{R}^{3\times 3}$

$$X_s = X \cdot \begin{bmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{bmatrix} \qquad X_r = X \cdot \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Basics of Vector and Matrices

- Matrix Diagonalization: Eigen decomposition of a matrix $X \in \mathbb{R}^{3\times3}$ is given by:

$$X = VDV^{-1}$$

  - Where, D is the diagonal matrix with eigen values $(\lambda_1, \lambda_2, \lambda_3)$ in leading diagonal, and
  - V is a matrix with corresponding eigen vectors (v_1, v_2, v_3) as columns.

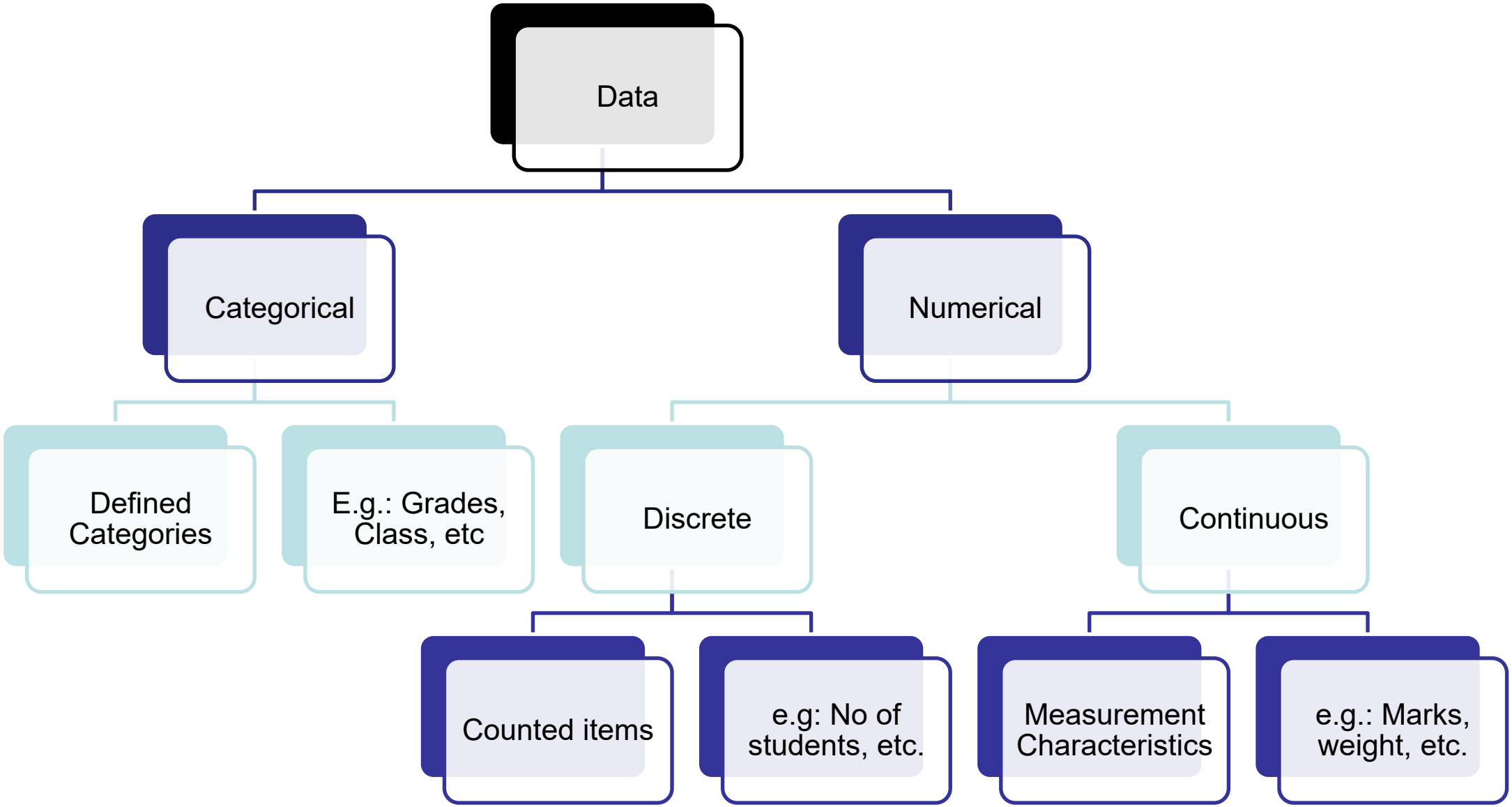
- D & V are given as:

$$D = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \qquad V = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}$$

Data and Related Definitions

- Data are **recorded measurements**, **generated by humans** and/or **machines** and **stored** in **structured** or **unstructured formats**.
- Measurements: numbers assigned to particular attributes or characteristic of a variable, through a standard process.
- Variables: characteristic of any entity being studied that is capable of taking on different values.
- Types of data: Categorical (category) and numerical (number or numerical indication of category).
- Levels of data:
  - nominal, ordinal, interval and ratio.
- Data helps in
  - evaluating the performance,
  - making better decisions, and
  - improving the processes.
- Analytics is different from analysis: Analysis is exploring past events whereas analytics is exploring potential future events.

Categories of Data



Data Cleaning

- Data is the new oil.
  - It should be refined according to the goal
- Data in the real world is dirty
  - *incomplete*: lacking attribute values, or containing only aggregate data
  - *noisy*: containing errors or outliers
  - *inconsistent*: containing discrepancies in codes or names

#	Id	Name	Birthday	Gender	IsTeacher?	#Students	Country	City
1	111	John	31/12/1990	M	0	0	Ireland	Dublin
2	222	Mery	15/10/1978	F	1	15	Iceland	
3	333	Alice	19/04/2000	F	0	0	Spain	Madrid
4	444	Mark	01/11/1997	M	0	0	France	Paris
5	555	Alex	15/03/2000	A	1	23	Germany	Berlin
6	555	Peter	1983-12-01	M	1	10	Italy	Rome
7	777	Calvin	05/05/1995	M	0	0	Italy	Italy
8	888	Roxane	03/08/1948	F	0	0	Portugal	Lisbon
9	999	Anne	05/09/1992	F	0	5	Switzerland	Geneva
10	101010	Paul	14/11/1992	M	1	26	Ytali	Rome

Missing values

Invalid values

Misfielded values

Uniqueness

Formats

Attribute dependencies

Misspellings

Steps in Data Cleaning

Removing Unobserved Observations	Missing Data Handling	Structural Error Solving	Outliers Management
• Duplicate/ redundant / irrelevant values deletion	• Fixing Issues of unknown values	• Fixing problems with mislabeled classes, typos in names of features, same attribute with different name, etc	• Unwanted values which are not fitting in datasets

Incomplete (Missing) Data

- Data is not always available
  - E.g., many observations have no recorded value for several attributes,
    - such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data
- Missing data may need to be inferred

How to Handle Missing Data?

- Ignore the Observation: usually done when class label is missing (when doing classification)—not effective when the % of missing values per attribute varies considerably
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
  - a global constant : e.g., “unknown”, a new class?!
  - the attribute mean
  - the attribute mean for all samples belonging to the same class: smarter
  - the most probable value: inference-based such as Bayesian formula or decision tree

Data Cleaning : Noisy data?

- Random error in a measured variable.
  - Incorrect attribute values may be due to
    - faulty data collection instruments
    - data entry problems
    - data transmission problems
    - technology limitation
    - inconsistency in naming convention
- Other data problems which requires data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

Data Cleaning: How to handle Noisy data?

- Binning method: (Bins- Smallest unit of space inside a **database)**
  - first sort data and partition into (equi-depth) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
  - used also for discretization
- Clustering
  - detect and remove outliers
- Semi-automated method: combined computer and human inspection
  - detect suspicious values and check manually
- Regression
  - smooth by fitting the data into regression functions

Data Cleaning: How to handle Inconsistent data?

- Manual correction using external references
- Semi-automatic using various tools
  - To detect violation of known functional dependencies and data constraints
  - To correct redundant data

Data Cleaning Process

Key Characteristics of Quality Data

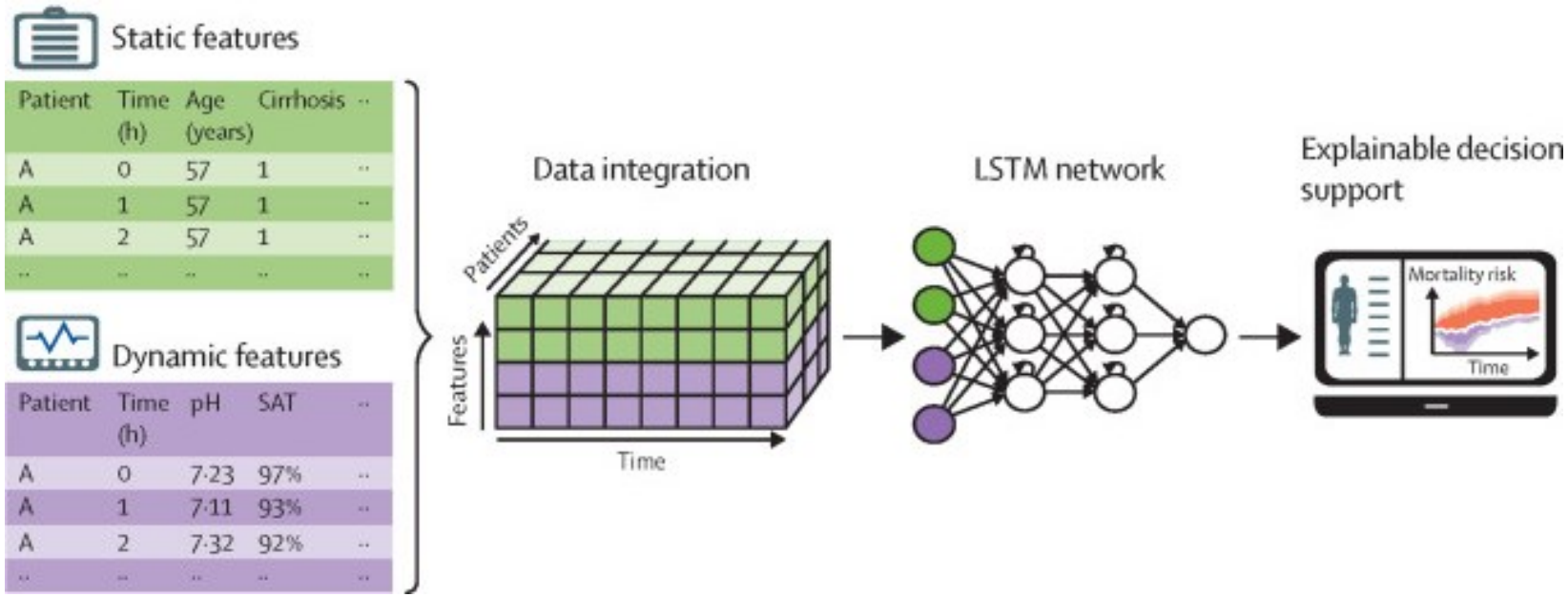
- **Validity.** Degree to which your data conforms to defined rules or constraints.
- **Accuracy.** Ensure your data is close to the true values.
- **Completeness.** The degree to which all required data is known.
- **Consistency.** Ensure your data is consistent within the same dataset and/or across multiple data sets.
- **Uniformity.** The degree to which the data is specified using the same unit of measure.

Benefits of Data Cleaning

- Removal of errors when multiple sources of data.
- Ability to map the different functions and what your data is intended to do.
- Monitoring errors and better reporting to see where errors are coming from, making it easier to fix incorrect or corrupt data for future applications.
- Using tools for data cleaning will make for more efficient practices and quicker decision-making.

Data Integration

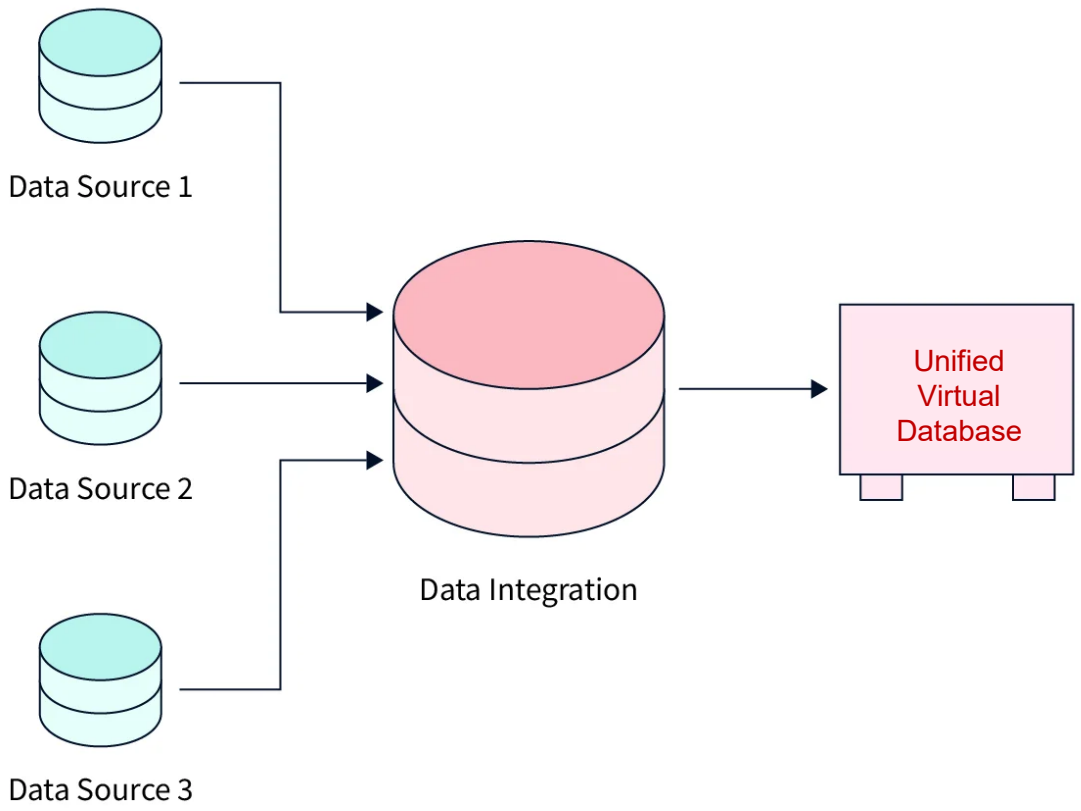
- *Combines data* from *multiple sources* into a *coherent store*
- Schema integration:
  - Integrate *metadata* from *different sources*
  - Entity identification problem: identify real world entities from multiple data sources



Data Integration

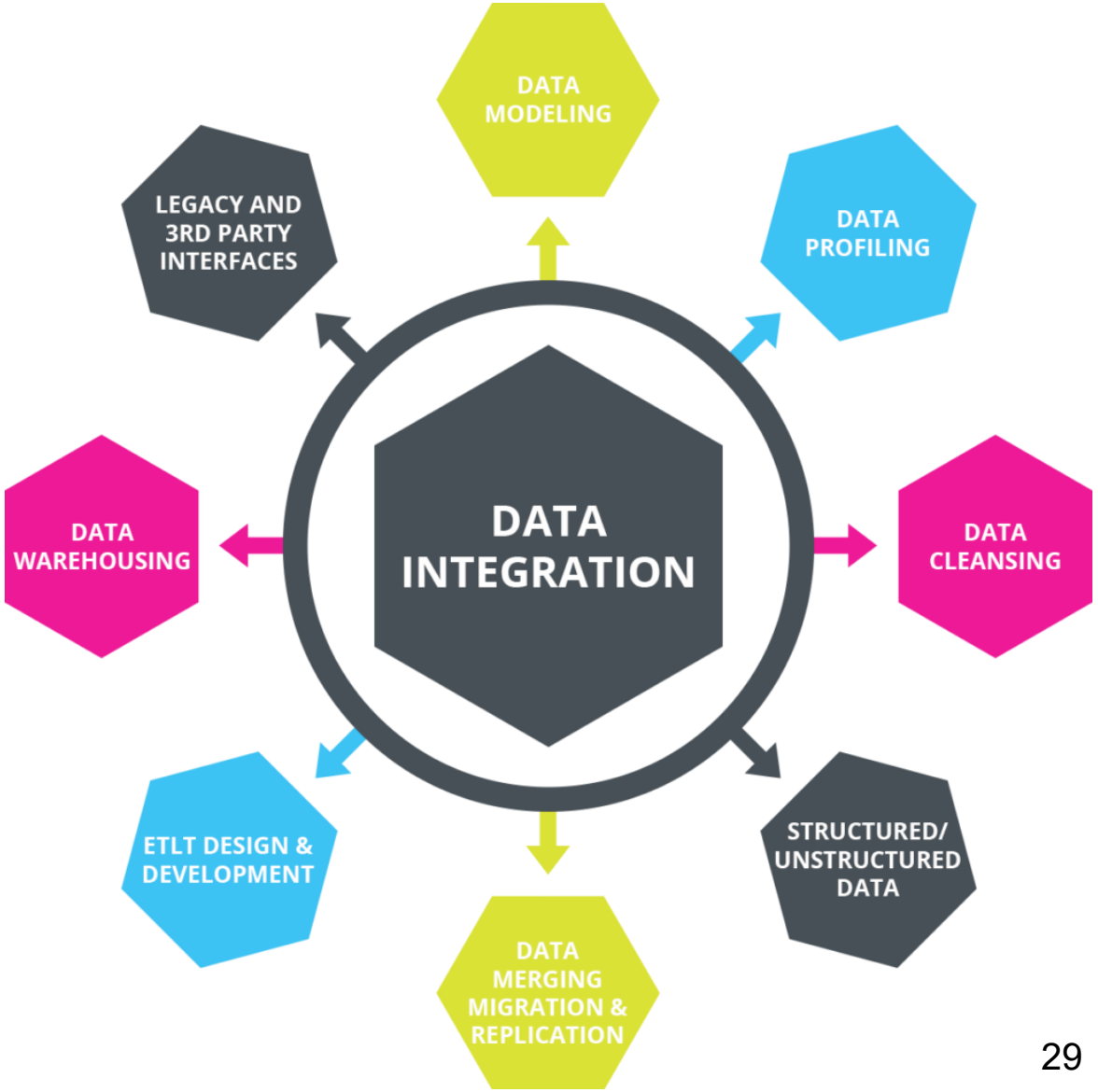
- Detecting and resolving data value conflicts: for the same real world entity, attribute values from different sources are different

- **Possible Reasons:**
  - Different representations,
  - Different scales, e.g., metric vs. British units, different currency

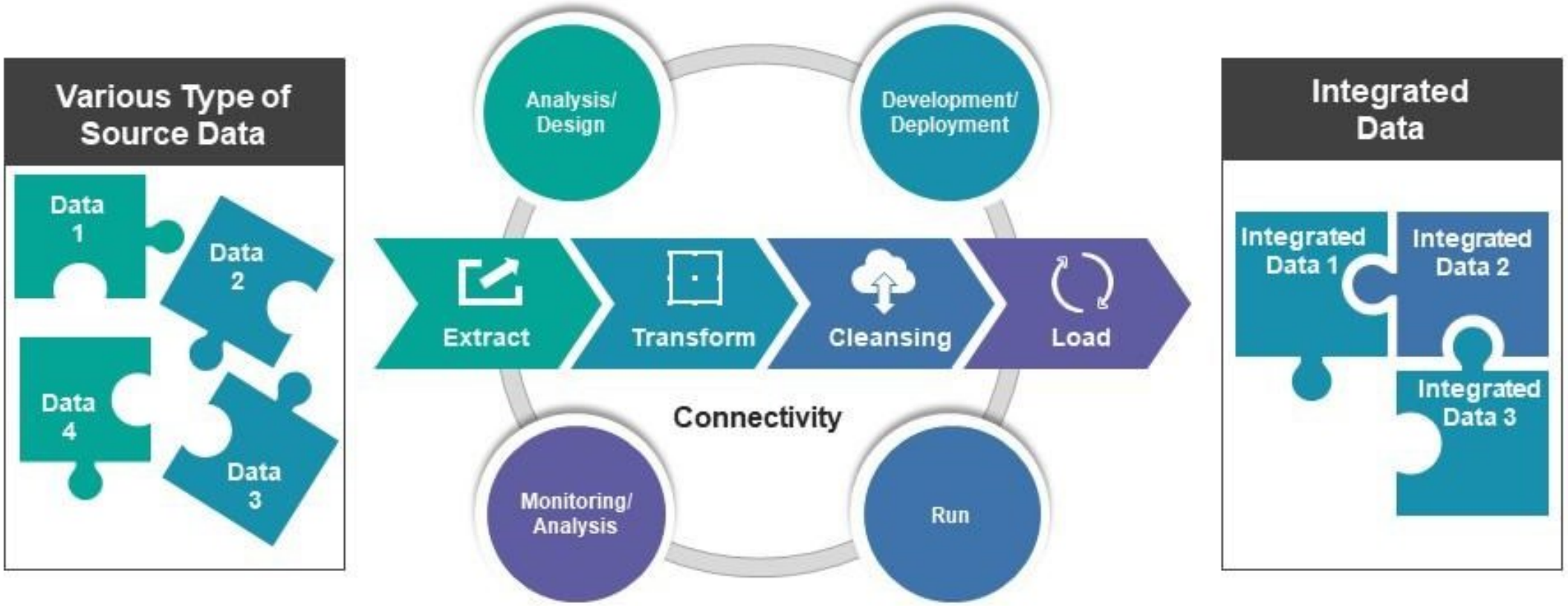


Why Data Integration is important?

- Gathering enormous volumes of data from various sources needs to be meaningful.
- Ease of accessible for analysis, when fresh data enters the database every second.
- Integrated data unlocks a layer of connectivity thereby improving the productivity.
- By connecting systems that contain valuable data and integrating them, ease of achieve data continuity and seamless knowledge transfer can be achieved.

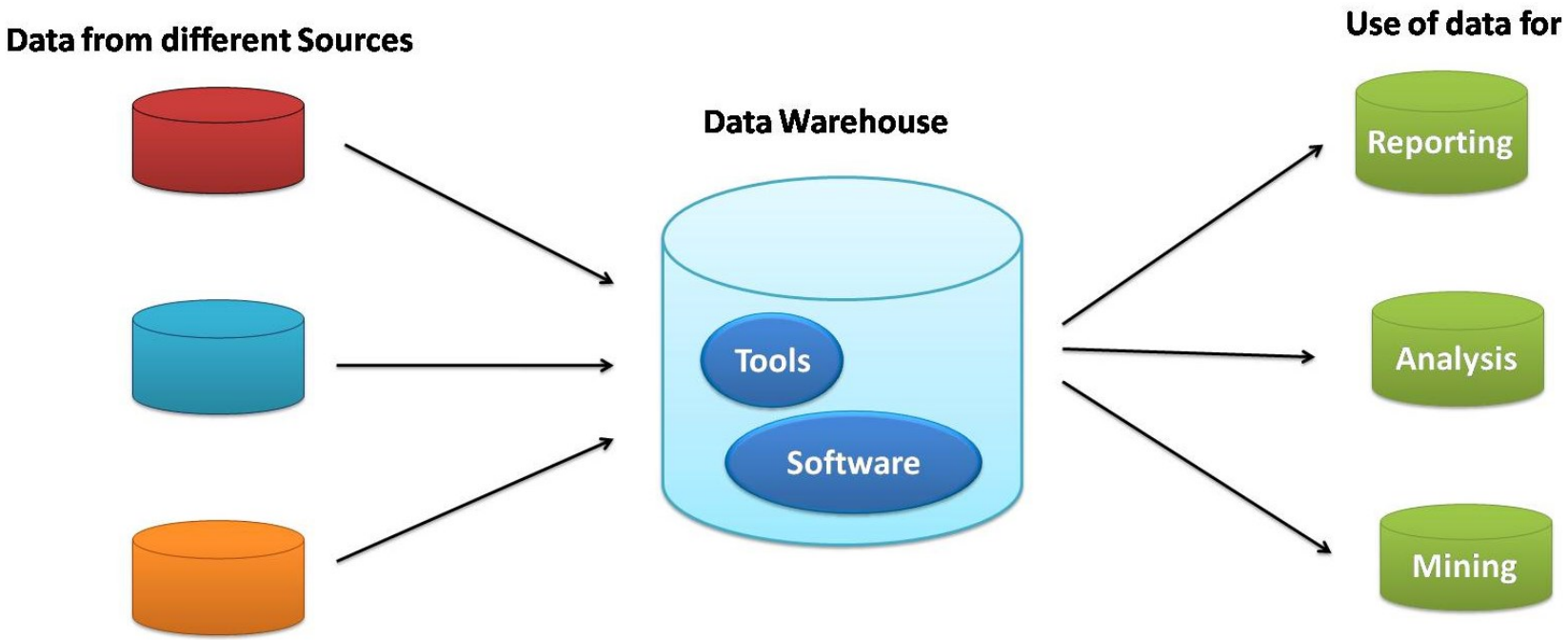


Data Integration



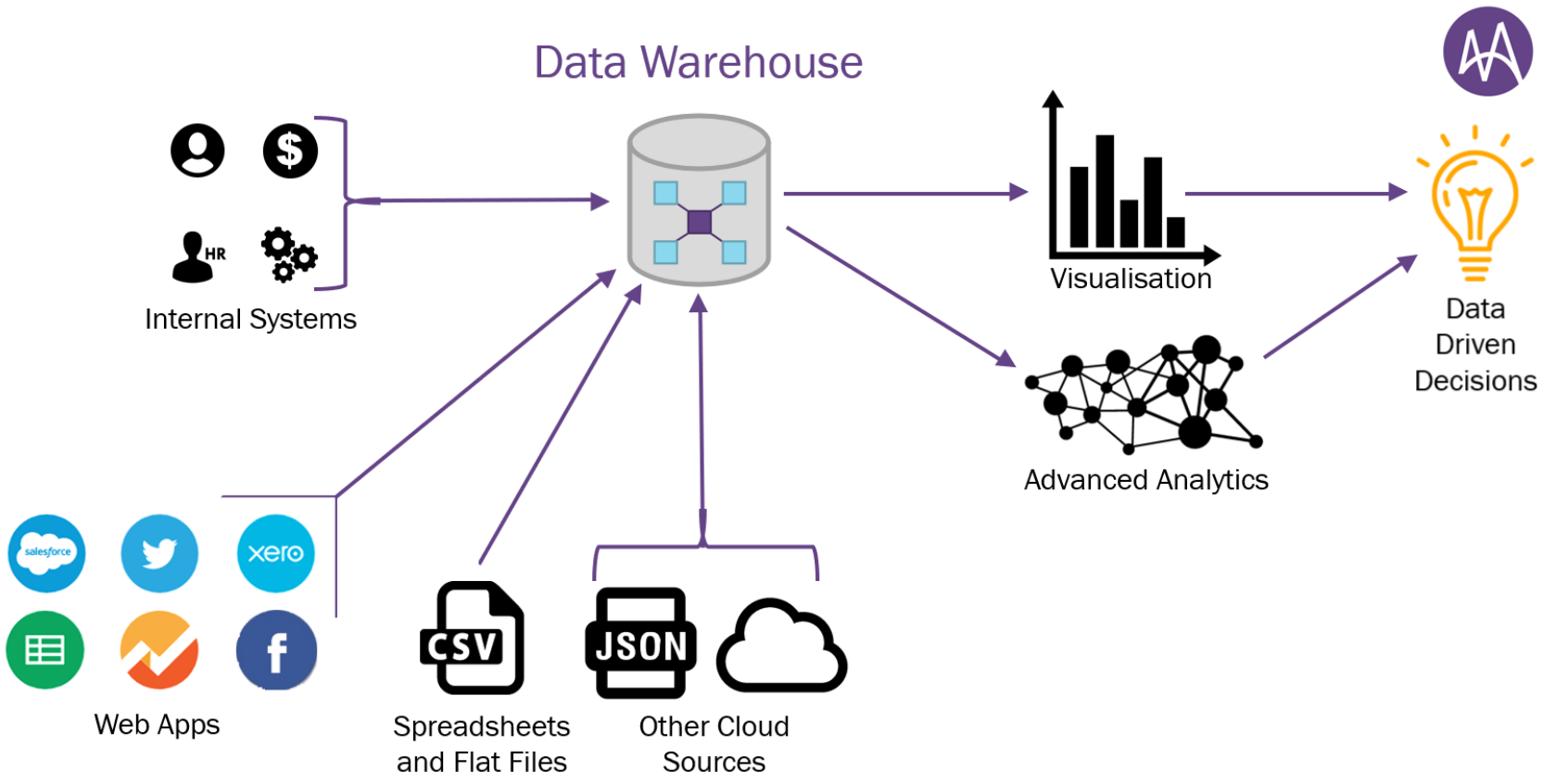
Methods for Data Integration

- **Creating a data warehouse:**
  - Allow for *integration* of *different sources* of *data* into a *master relational database*.
- **When critical data is collected, stored and easily available,**
  - it's much easier to assess micro and macro processes,
  - Manage operations and make strategic decisions based on this intelligence.



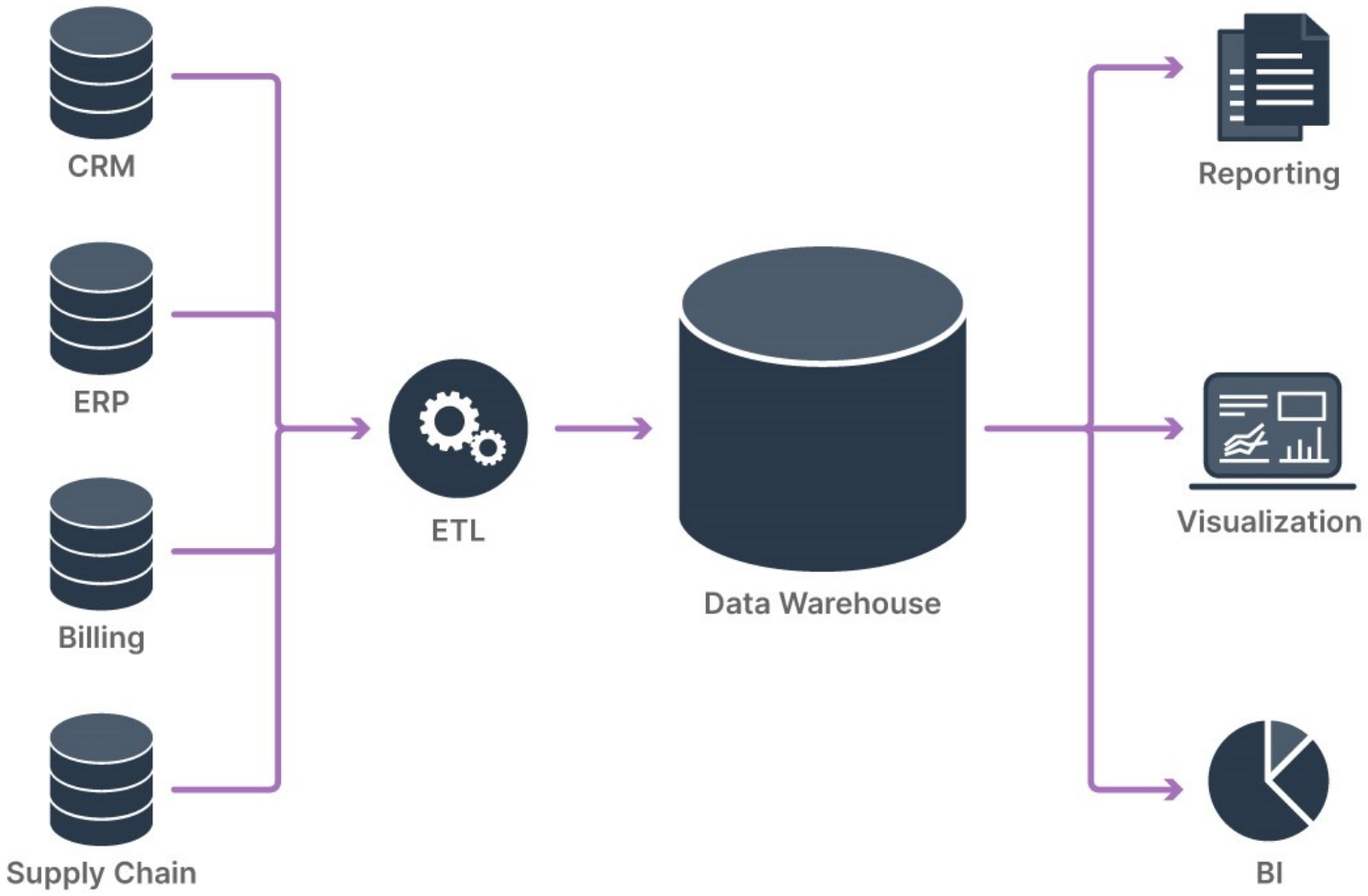
Ways to Integrate Data

- In this case, data integration works by *providing a cohesive and centralized look* at the entirety of an organization's information, streamlining the process of gaining intelligence insights.



Ways to Integrate Data

To achieve this, the managed service provider would a process called **ETL**.

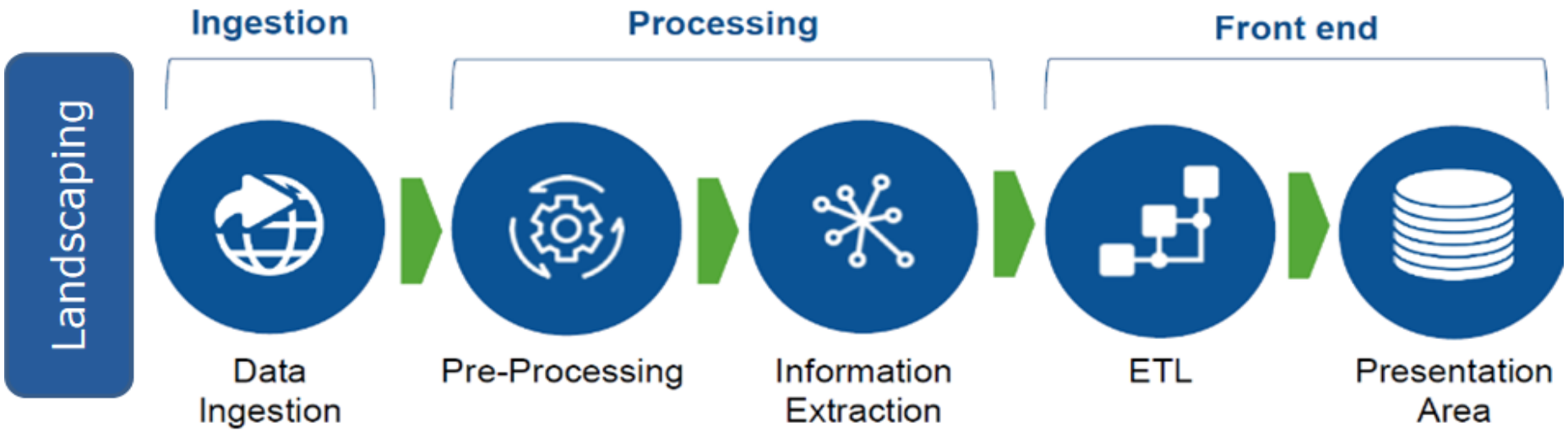


Ways to Integrate Data

- **ETL (Extract, Transform, Load):** Process of sending data from source systems an organization possesses to the data warehouse where this information will be viewed and used.
- Most data integration systems involve one or more ETL pipelines, which make data integration easier, simpler, and quicker.

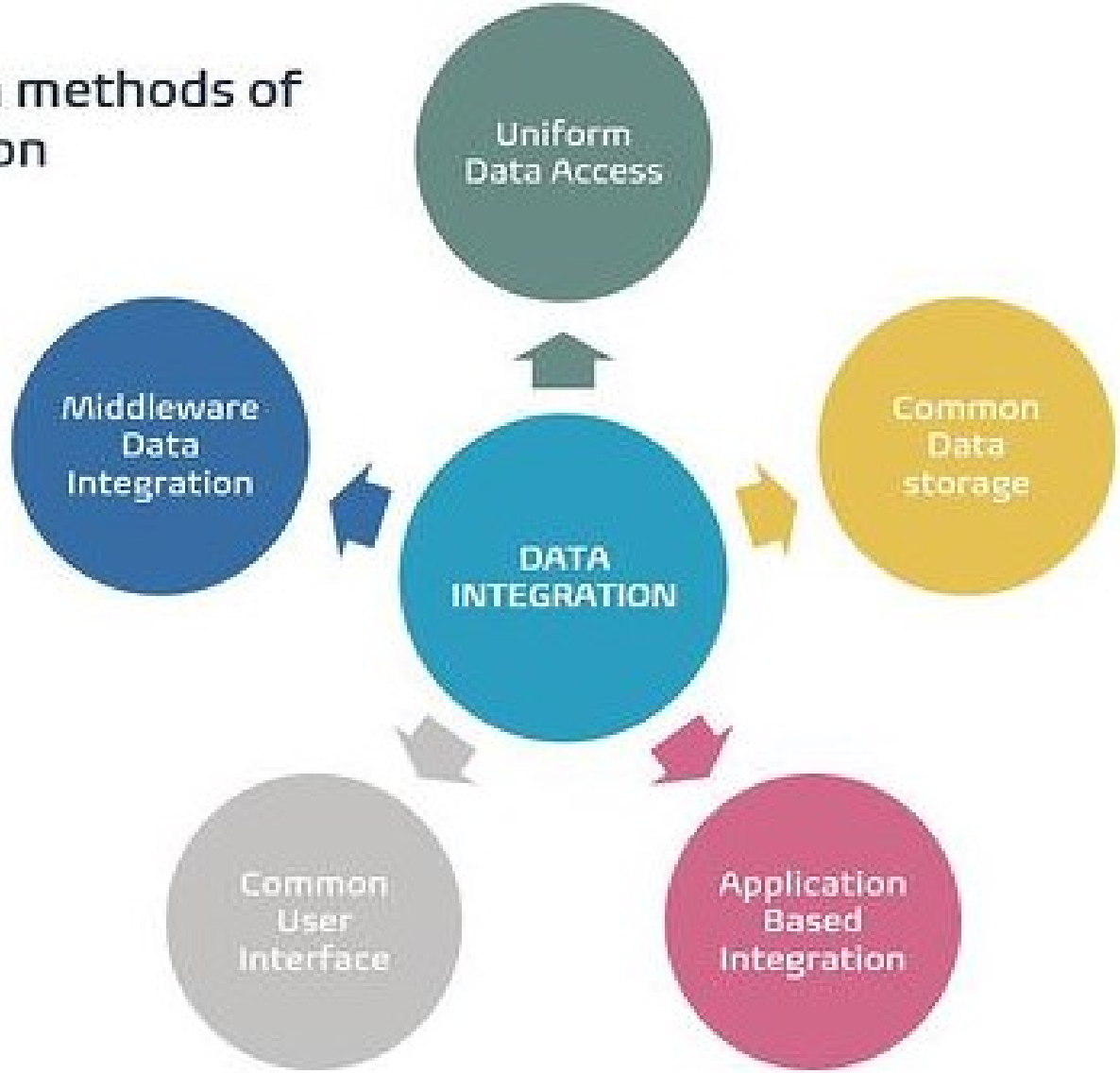
Ways to Integrate Data

- **Building Data Pipelines:**
  - A variety of built-in data connectors (for data ingestion), pre-defined transformations, and built-in job scheduler for automating the ETL pipeline.
  - Such tools **make data integration** *easier*, *faster*, and *more cost effective* by reducing the dependency on human expertise for manual operation.



Types of Data Integration Methods

Most common methods of data integration



Types of Data Integration Methods

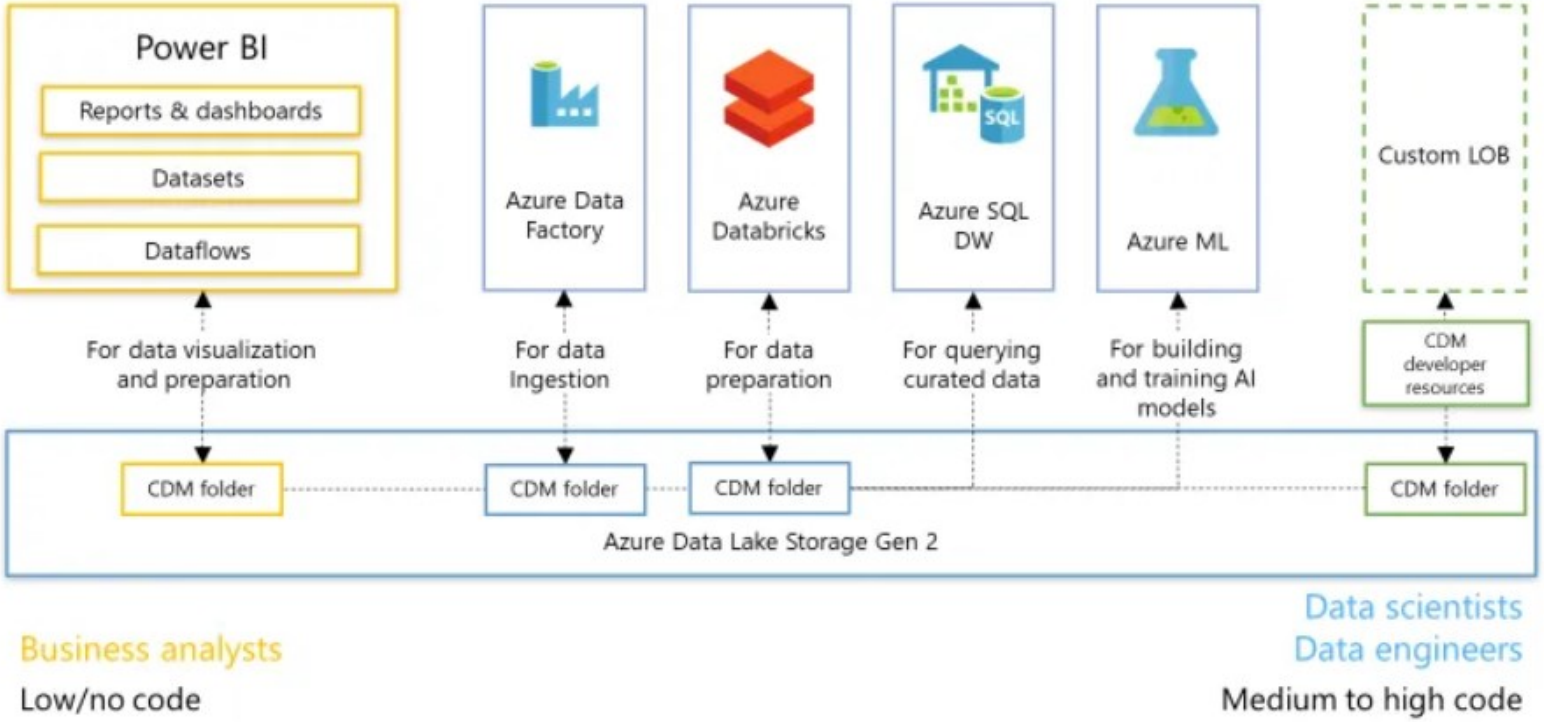
Uniform Data Access:

- With Uniform Data Access, data can be accessed from very disparate sets and present it uniformly.
- Uniform Data Access does this while allowing the data to stay in its original location.
- It leaves the data in the source system and defines a set that can provide a unified view to various features across a platform.
- There is *zero latency* from the *source system* to the *consolidated view*.

Types of Data Integration Methods

Common Data Storage:

- CDS is a *storage space* that *enables* you to *manage* and *securely store data* used by *multiple applications* or *programs*.
- Helps *collecting data* from *various sources*, *combining them* to a *central space* and *management* (Database files, mainframes, and flat files).



Types of Data Integration Methods

Application based Integration:

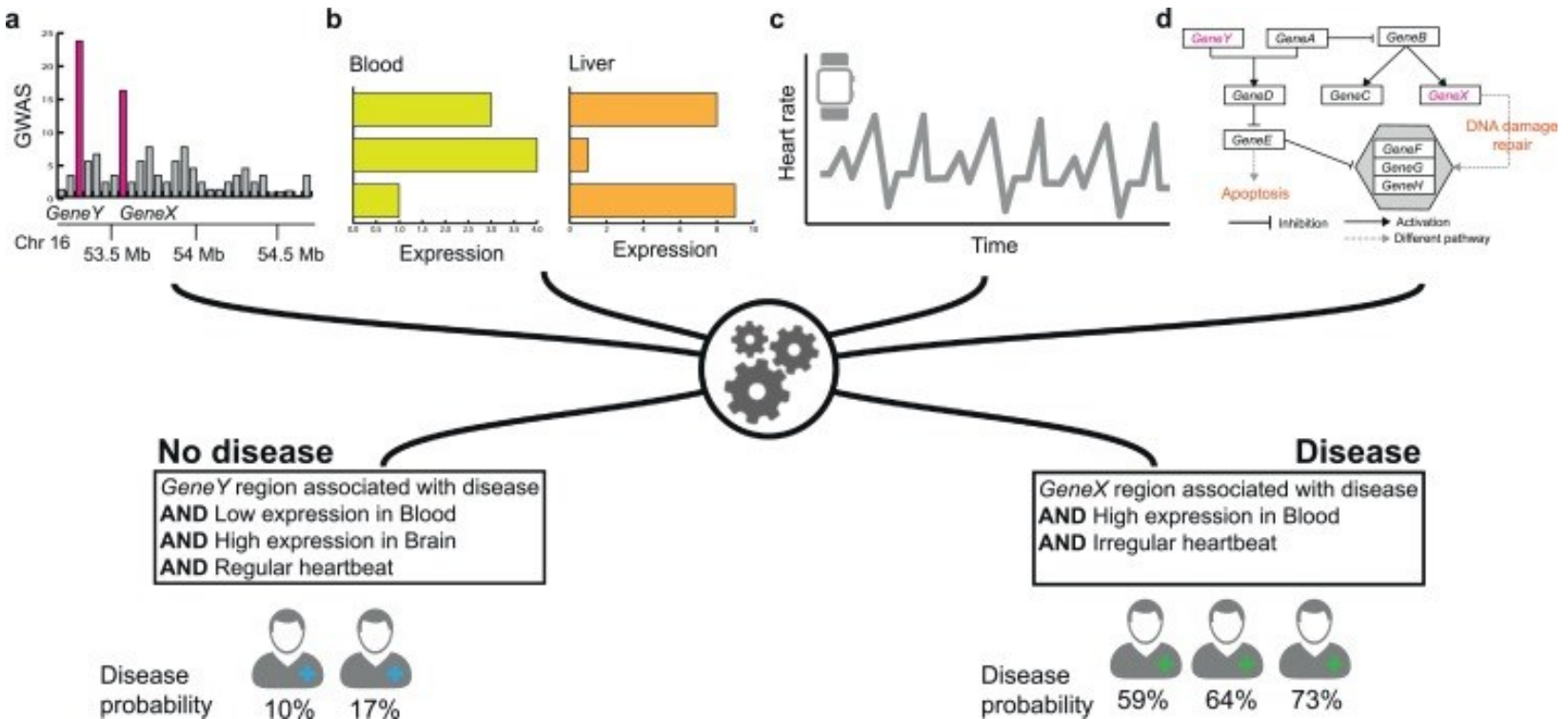
- Specialized programs that locate, retrieve and integrate data.
- This integration accesses various data sources and returns integrated results to user.
- Limitations if you are *handling large volumes* of *data* and *large numbers* of *sources* because it *requires* the *applications* to *implement* all the *integration efforts*.

Types of Data Integration Methods

Common User Interface:

- ***Manually locating*** the ***information*** in ***each data source*** and ***comparing*** or ***cross-referencing*** them ***yourself*** in ***order*** to get the ***insight*** you ***need***.
- Users must deal with ***different user interfaces*** and ***query languages*** and therefore ***need to have detailed knowledge on location***, ***logical data representation***, and ***data semantics***.

An example for Common User Interface



Types of Data Integration Methods

Middle ware Data integration:

- Middleware is a **layer of software** that *creates a common platform for all interactions internal and external* to the organization
  - System-to-system,
  - System-to-database,
  - Human-to-system,
  - Web-based, and mobile-device-based interactions.

- Middleware integration *refers to applications that connect two or more applications*.

Handling Redundant Data in Integration

- Redundant data ***occur often*** when ***integrating multiple DBs***
- **Same attribute** may have **different names** in **different databases**
- One attribute may be a **"derived"** attribute in another table,
  - e.g., annual revenue
- Redundant data may be able to be detected by correlational analysis

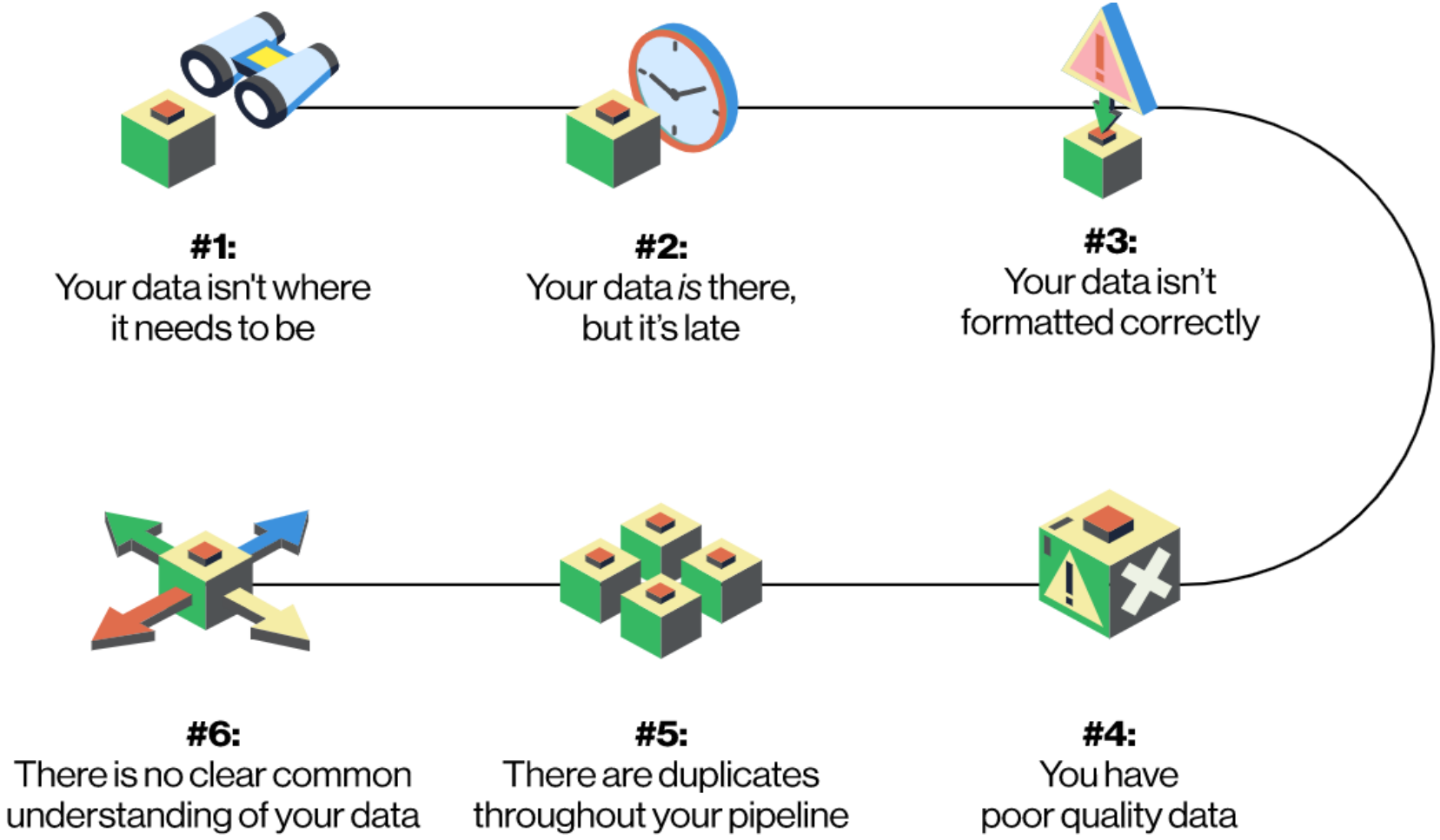
$$r_{A,B} = \frac{\Sigma(A-\overline{A})(B-\overline{B})}{(n-1)\sigma_A\sigma_B}$$

- Careful integration can help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Advantages of Data Integration

- Elimination of errors
- Saves time
- Better sense of all the available information.
- Streamline the processes and improve the efficacy of data usage.
- Inter-system cooperation
- Seamless knowledge transfer between systems.
- Data integrity and data quality.

Six challenges and its solution of Data Integration



Solution to challenges:

- Clean up your data
- Introduce clear processes for data management
- Back up your data
- Choose the right software to assist you with data integration
- Manage and maintain your data

Data Transformations

Data Transformation

- **Data Transformation**: Process of converting data from one format or structure into another format or structure.
  - It is the middle step in ETL (*extract*, *transform*, *load*) process.
- These method can be simple/ complex based on the changes required.
  - Typically performed via a mixture of manual and automated steps.
  - *Tools and technologies used*: Vary widely based on the format, structure, complexity, and volume of the data being transformed.
- Data transformation process/steps: Data discovery, data mapping, code generation, code execution and data review.
- Types of data transformation: Batch data transformation (coding based) and interactive data transformation (analysis/analytics based).

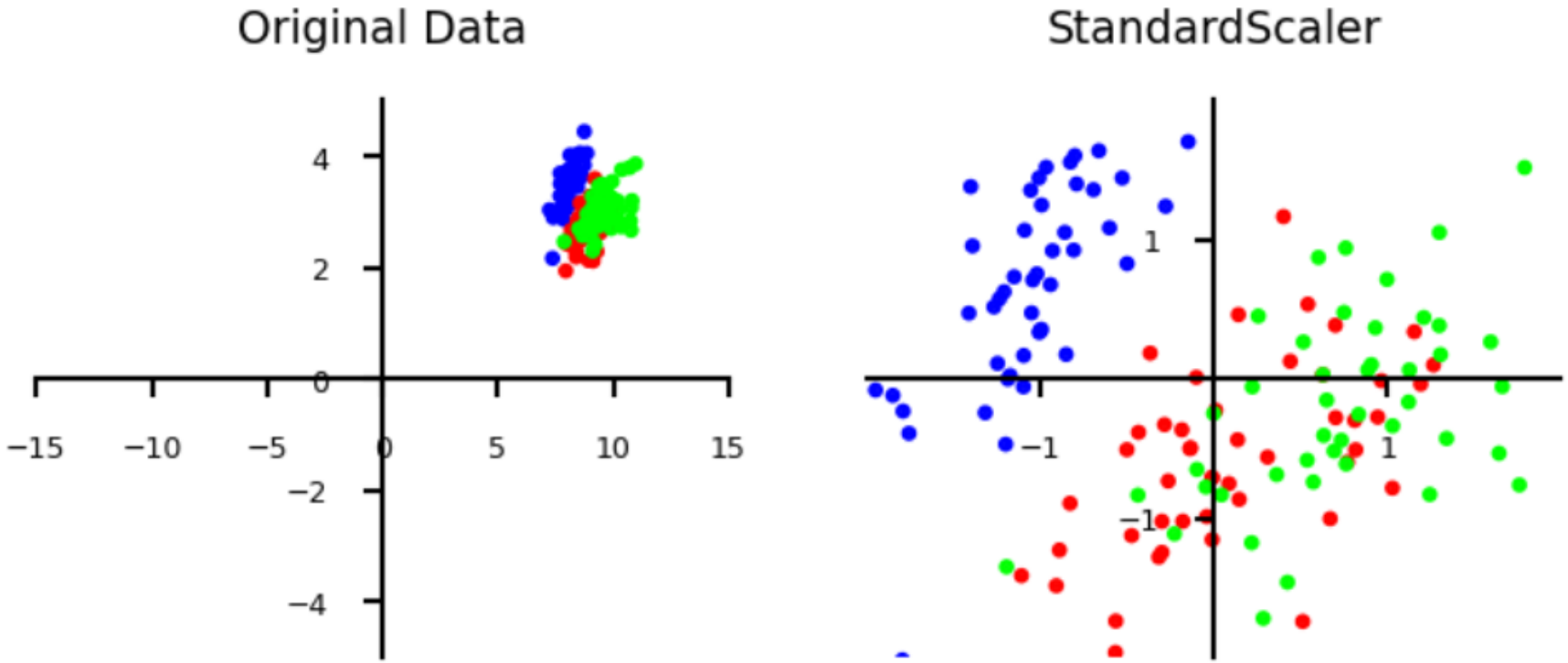
Data Transformation

- Machine learning models make a lot of assumptions about the data
- In reality, these assumptions are often violated
- We build pipelines that transform the data before feeding it to the learners
  - Scaling (or other numeric transformations)
  - Encoding (convert categorical features into numerical ones)
  - Automatic feature selection
  - Feature engineering (e.g. binning, polynomial features,...)
  - Handling missing data
  - Handling imbalanced data
  - Dimensionality reduction (e.g. PCA)
  - Learned embeddings (e.g. for text)
- Seek the best combinations of transformations and learning methods
  - Often done empirically, using cross-validation
  - Make sure that there is no data leakage during this process!

Data Transformation:

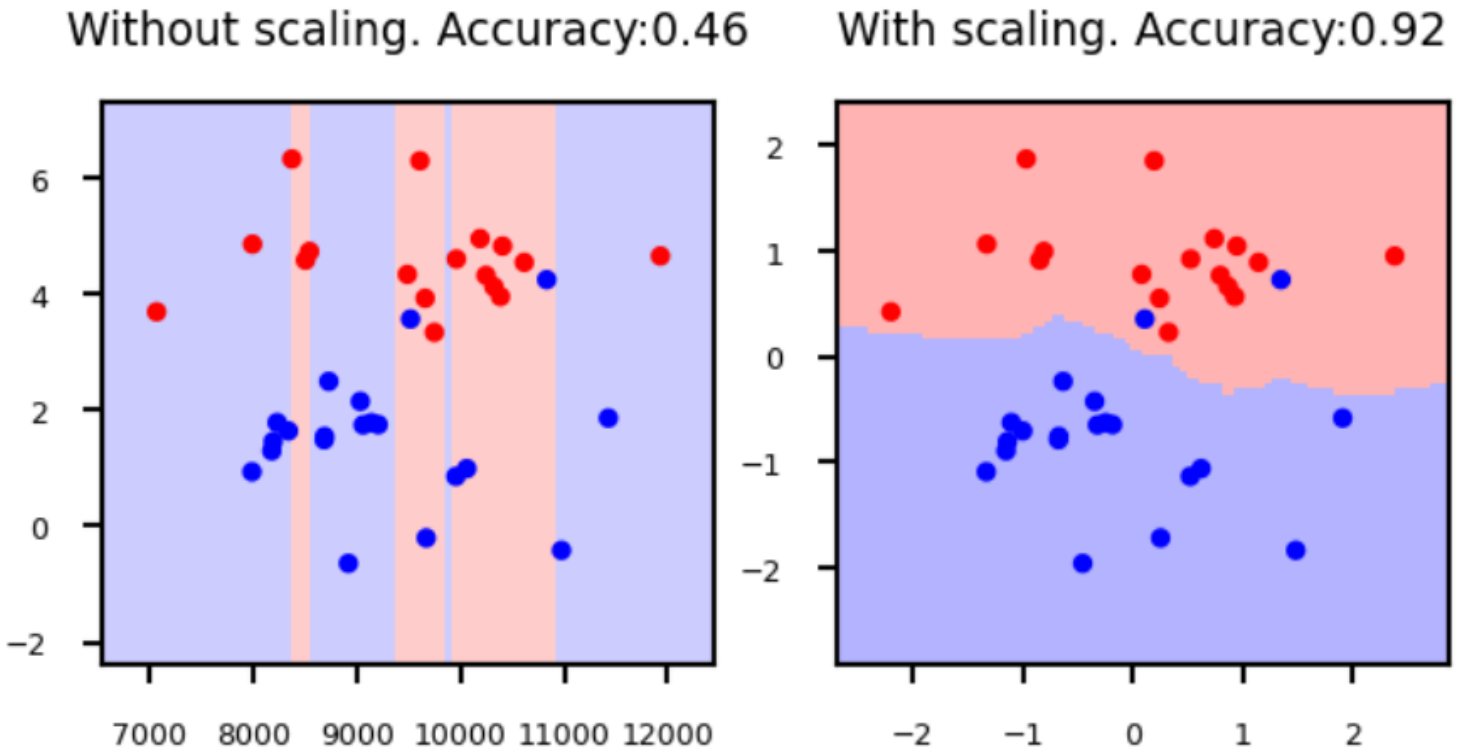
Scaling

- Use when different numeric features have different scales (different range of values)
- Features with much *higher values* may ***overpower*** the *others*
- **Goal**: bring them all within the same range
- Different methods exist



Scaling

- *Why Scaling is needed?*
  - *KNN*: Distances depend mainly on feature with larger values
  - *SVMs*: (kernelized) dot products are also based on distances
  - *Linear model*: Feature scale affects regularization
  - Weights have similar scales, more interpretable

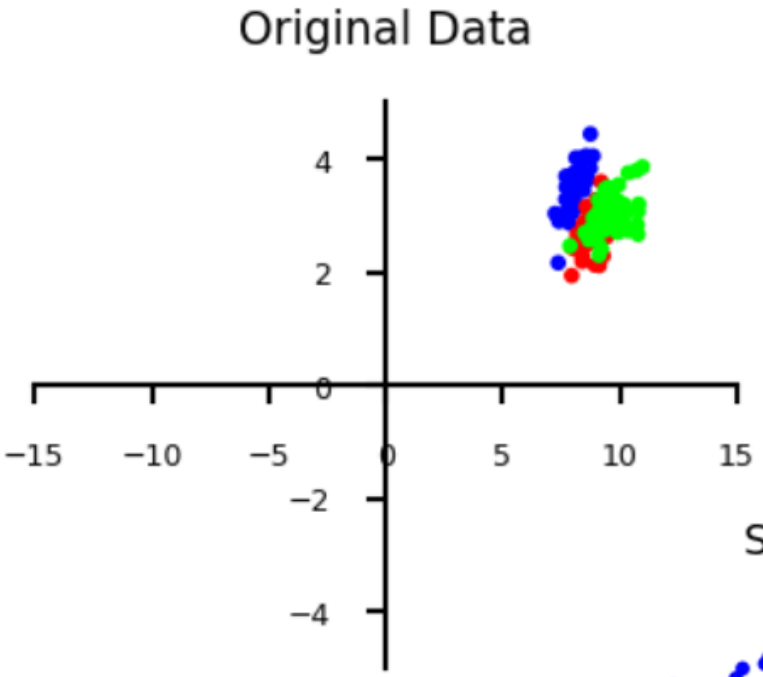


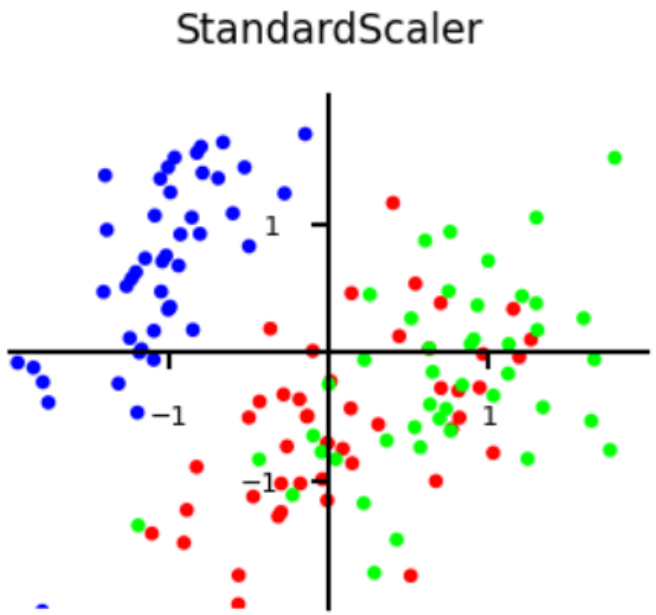
Data Transformation:

Standard Scaling [standardization]

- Generally most useful, assumes data is more or less normally distributed per feature,
- **Subtract** the **mean value μ**, scale by **standard deviation σ**
- New feature has μ = 0 and σ = 1, values can still be arbitrarily large

$$\mathbf{x}_{new} = \frac{\mathbf{x} - \mu}{\sigma}$$



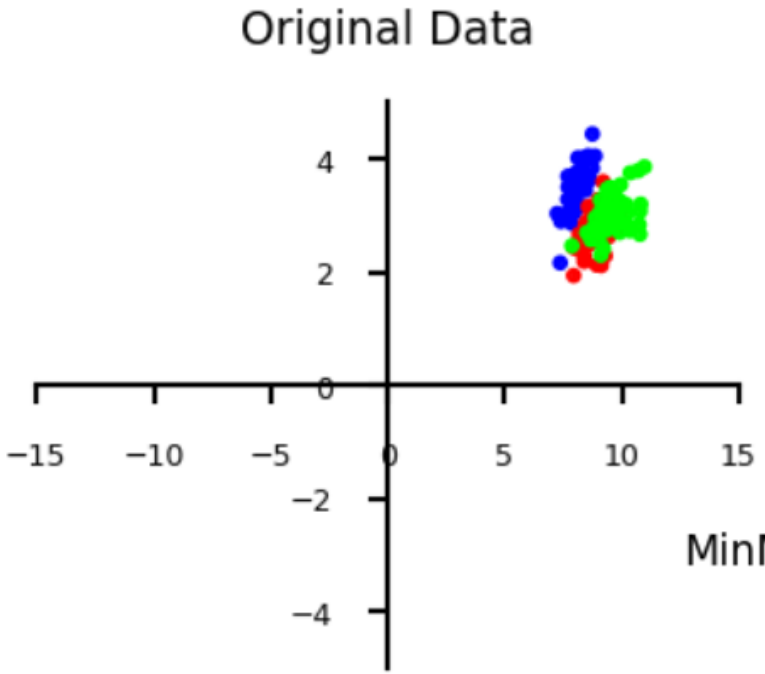


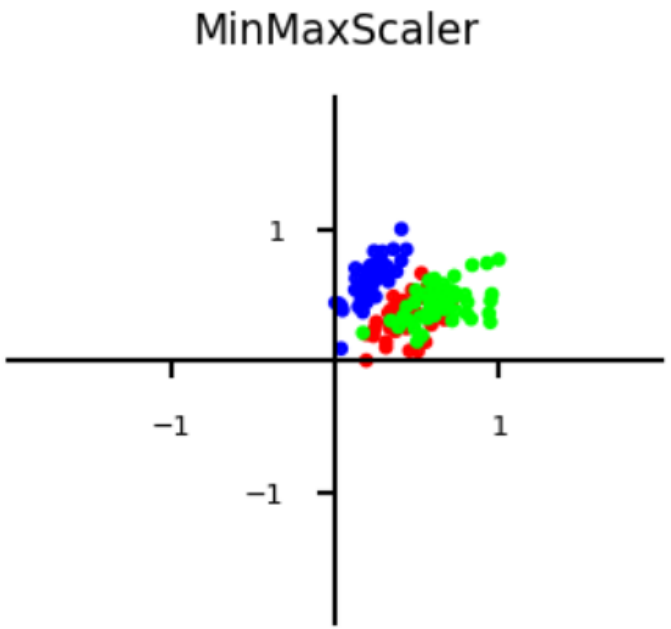
Data Transformation:

Min-Max Scaling

- Scales all features between a given min & max value (e.g. 0 and 1)
- Makes sense if min/max values have meaning in your data
- Sensitive to **outliers**

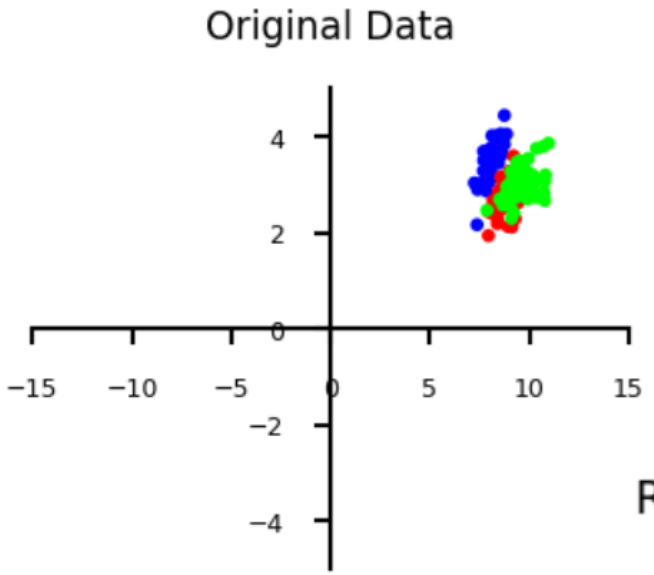
$$\mathbf{x}_{new} = \frac{\mathbf{x} - x_{min}}{x_{max} - x_{min}} \cdot (max - min) + min$$

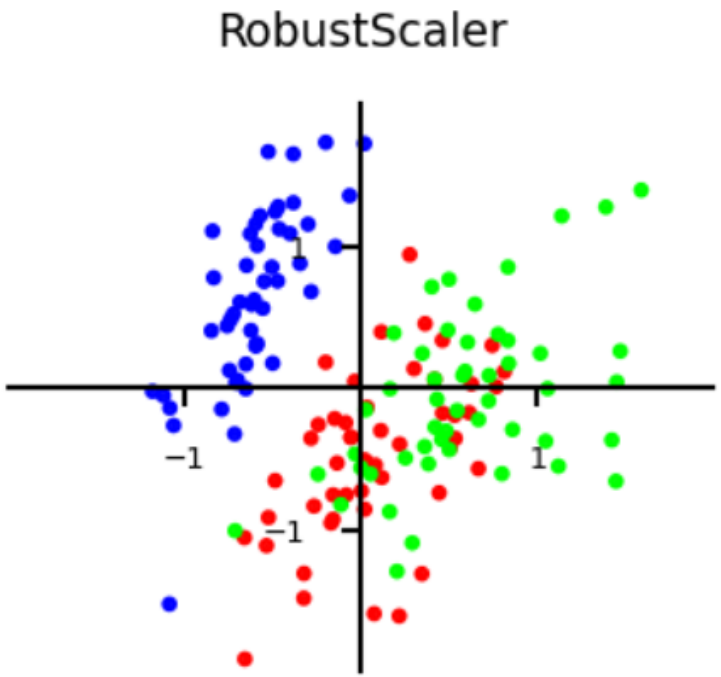




Robust Scaling

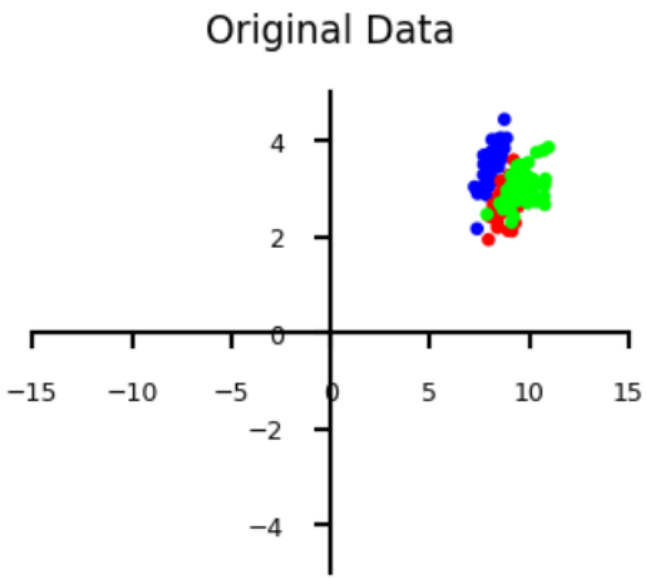
- Subtracts the median, scales between quantiles q_{25} and q_{75}
- New feature has median 0, $q_{25} = -1$ and $q_{75} = 1$
- Similar to standard scaler, but ignores outliers

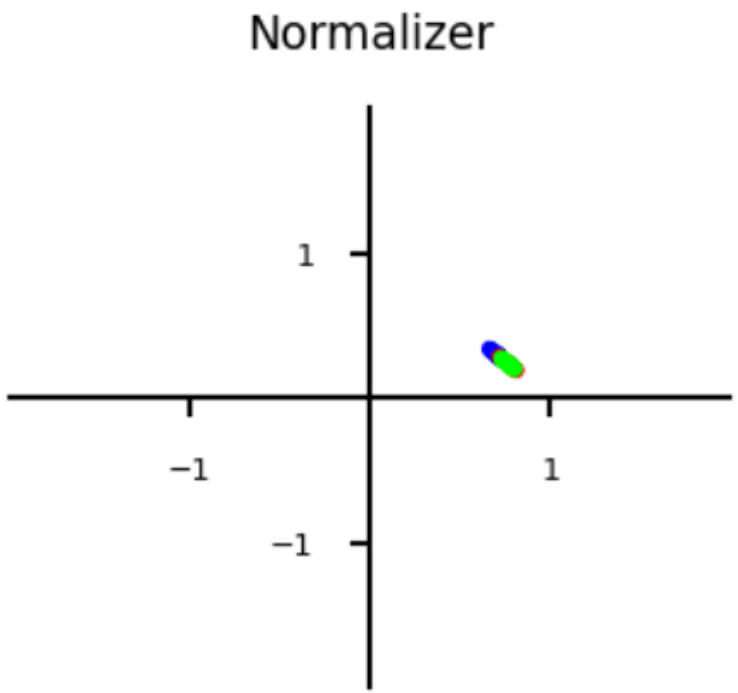




Normalization Scaling

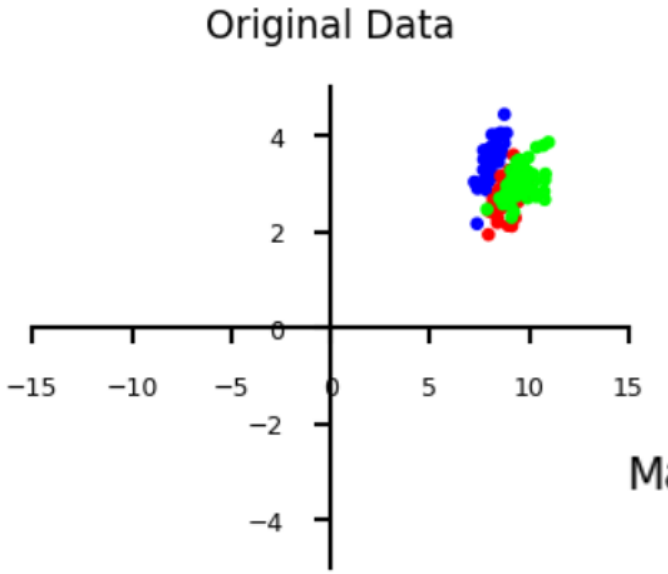
- Makes sure that **feature values** of each point (each row) **sum up to 1** (L1 norm)
- Useful for **count data**
  - (e.g. word counts in documents)
- Can also be used with **L2 norm**
  - (sum of squares is 1)
- Useful when computing distances in high dimensions
- Normalized Euclidean distance is equivalent to **cosine similarity**

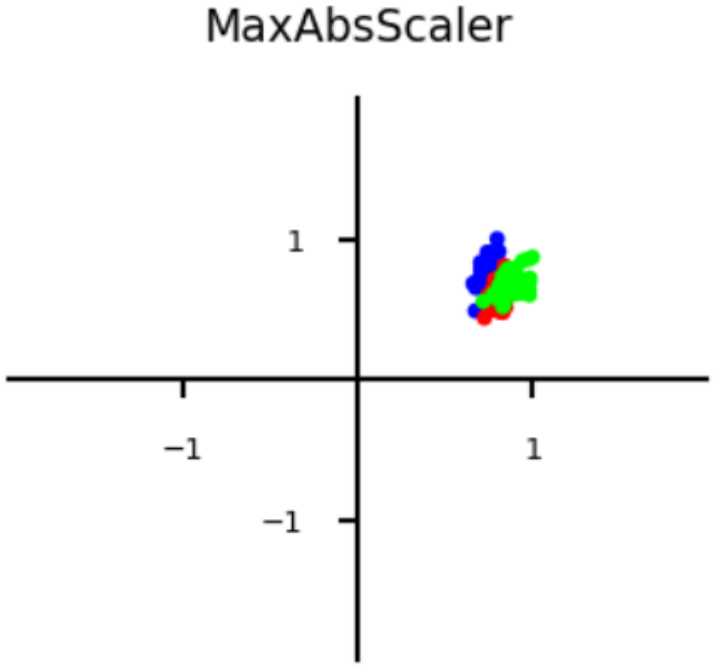




Max. Abs. Scaling

- For sparse data
  - (many features, but few are non-zero)
- Maintain sparseness
  - (efficient storage)
- Scales all values so that maximum **absolute** value is 1
- Similar to Min-Max scaling without changing 0 values

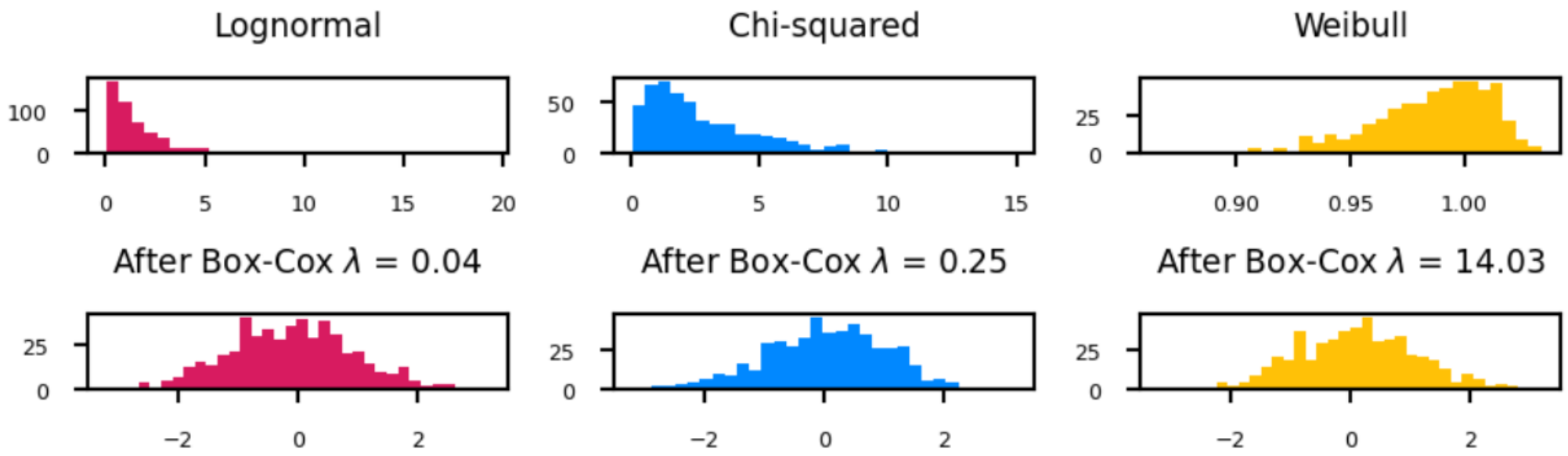




Power Transforms

- Some features follow certain distributions
  - E.g. number of twitter followers is log-normal distributed
- Box-Cox transformations transform these to normal distributions (λ is fitted)
  - Only works for positive values, use Yeo-Johnson otherwise

$$bc_\lambda(x) = \begin{cases} log(x) & \lambda = 0 \\ \frac{x^\lambda - 1}{\lambda} & \lambda \neq 0 \end{cases}$$



Categorical Feature Encoding

- Many algorithms can only handle numeric features, so we need to encode the categorical ones

	Boro	Salary	Vegan
0	Manhattan	103	0
1	Queens	89	0
2	Manhattan	142	0
3	Brooklyn	54	1
4	Brooklyn	63	1
5	Bronx	219	0

Ordinal Feature Encoding

- Simply assigns an integer value to each category in the order they are encountered
- Only really useful if there exist a natural order in categories
- Model will consider one category to be 'higher' or 'closer' to another

	boro	boro_ordinal	Salary
0	Manhattan	2	103
1	Queens	3	89
2	Manhattan	2	142
3	Brooklyn	1	54
4	Brooklyn	1	63
5	Bronx	0	219

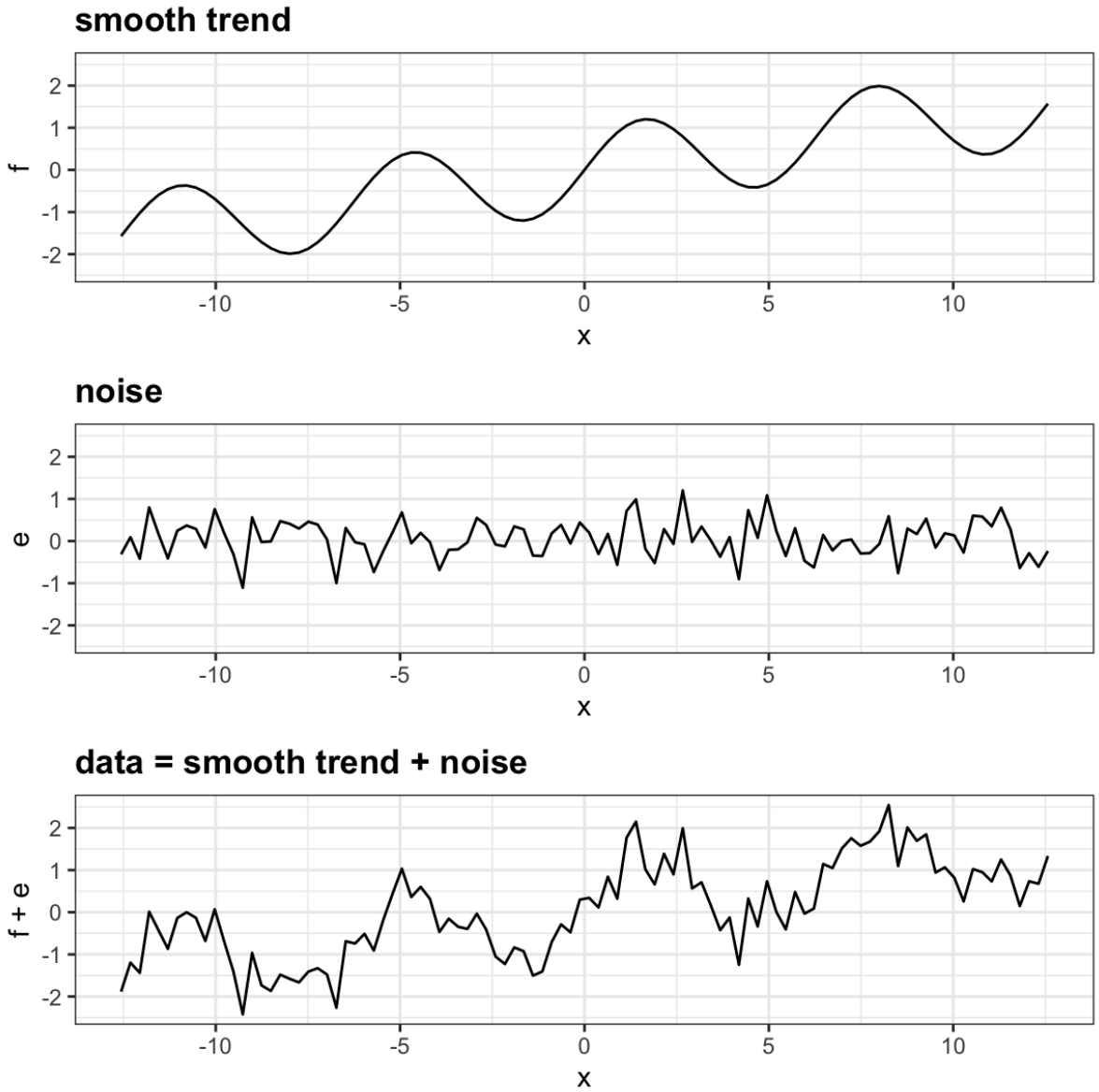
One Hot Encoding

- Simply adds a new 0/1 feature for every category, having 1 (hot) if the sample has that category
- Can **explode** if a feature has lots of values, causing issues with high dimensionality
- What if test set contains a new category not seen in training data?
  - Either ignore it (just use all 0's in row), or handle manually (e.g. resample)

	boro	boro_Bronx	boro_Brooklyn	boro_Manhattan	boro_Queens	salary
0	Manhattan	0	0	1	0	103
1	Queens	0	0	0	1	89
2	Manhattan	0	0	1	0	142
3	Brooklyn	0	1	0	0	54
4	Brooklyn	0	1	0	0	63
5	Bronx	1	0	0	0	219

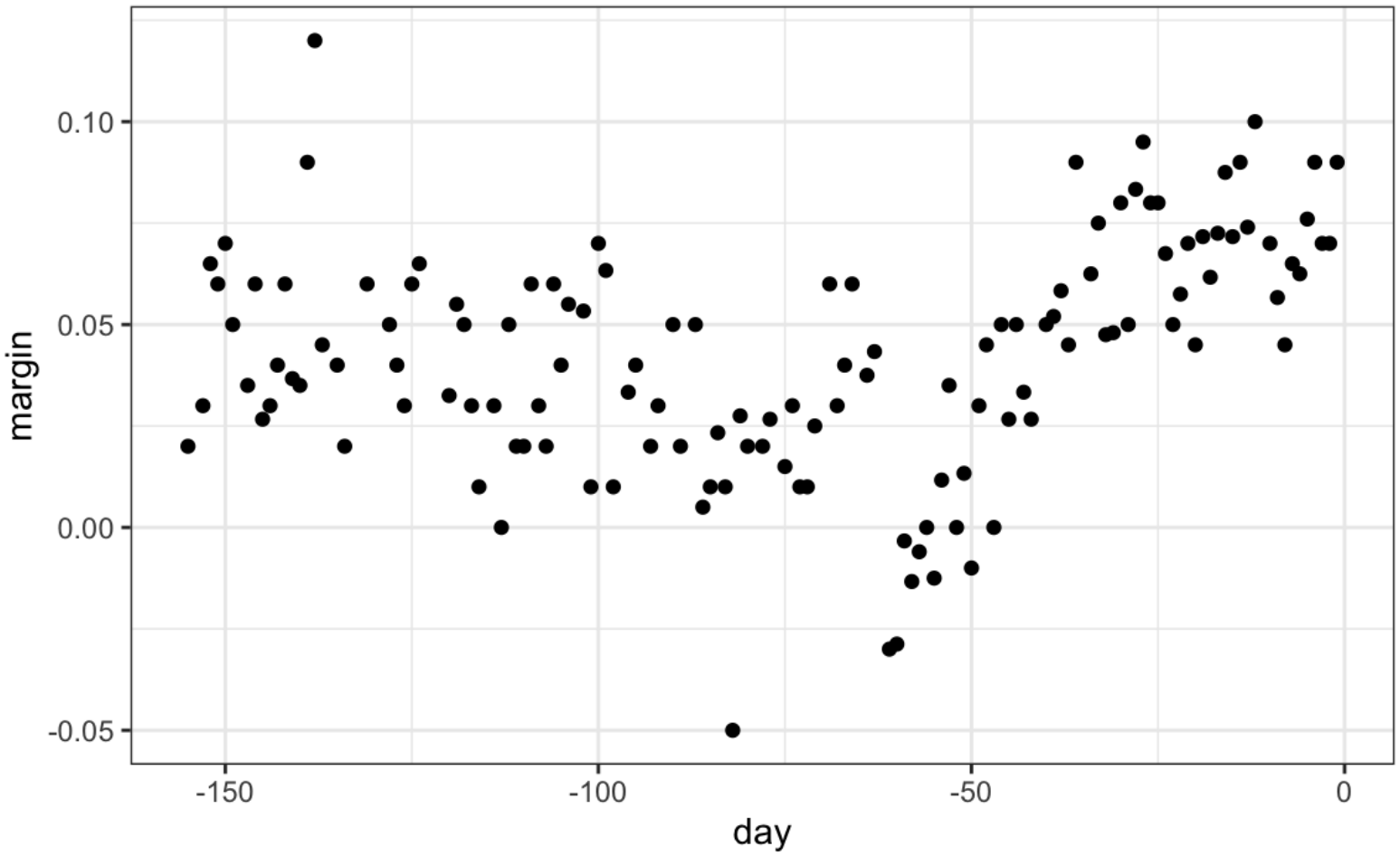
Data Smoothing

- Smoothing is a very powerful technique used all across data analysis.
- Other names given to this technique are curve fitting and low pass filtering.
- It is designed to detect trends in the presence of noisy data in cases in which the shape of the trend is unknown.
- Smoothing name:
  - we assume that the trend is smooth, as in a smooth surface.
  - In contrast, the noise, or deviation from the trend, is unpredictably wobbly:



Data Smoothing

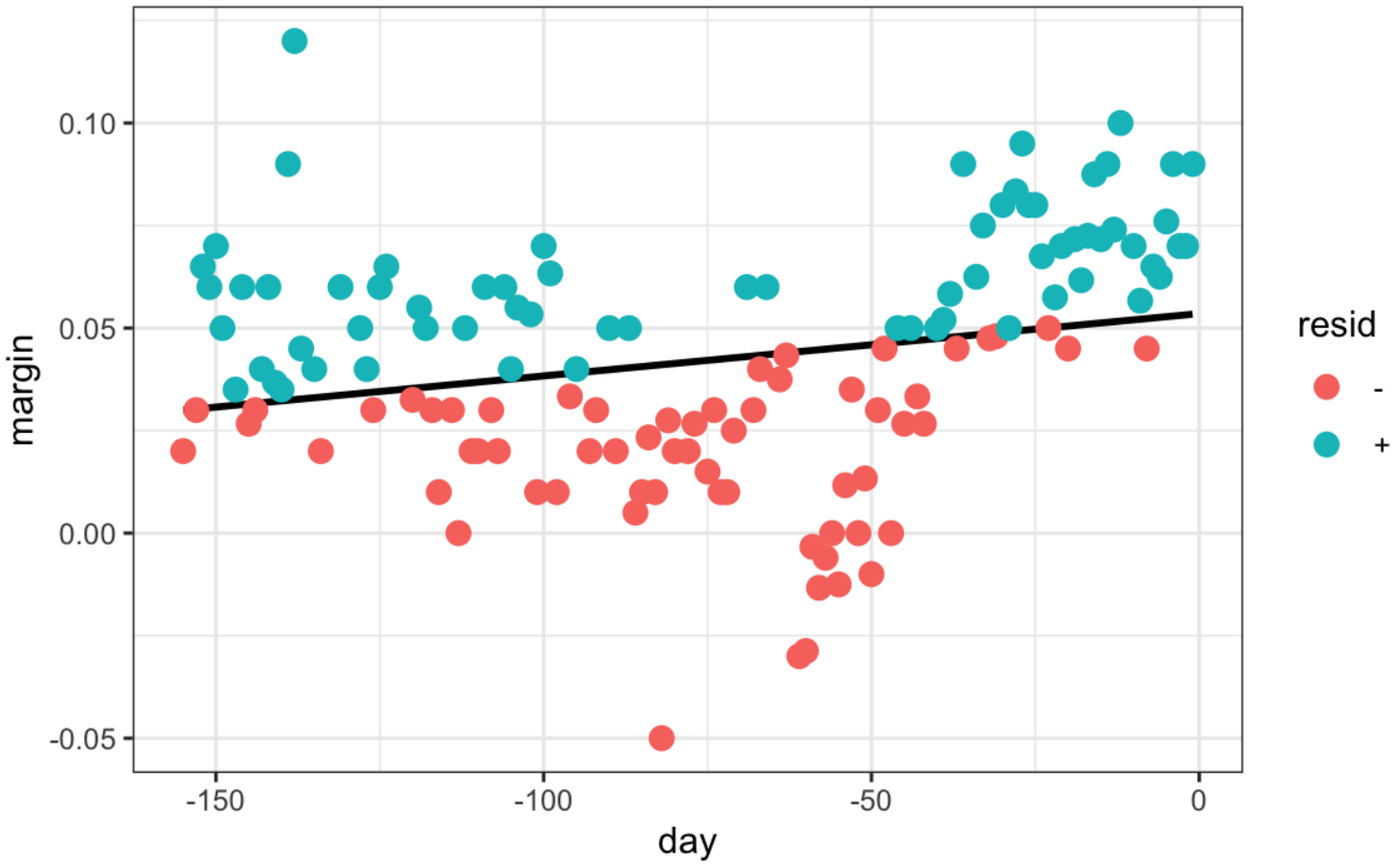
- *Example*: Estimate the time trend in the 2008 US popular vote poll margin (difference between Obama and McCain).



We are simply interested in learning the shape of the trend after the election is over.

Data Smoothing

- *Example*: If regression is used to estimate the trend



- The line we see **does not appear to describe the trend** very well.
- ***For example***, on September 4 (day -62), the Republican Convention was held and the *data suggest that it gave John McCain a boost in the polls*.
- However, the **regression line does not capture this potential trend**.
- To see the **lack of fit more clearly**, we note that ***points above the fitted line (blue) and those below (red) are not evenly distributed*** across days.
- We therefore need an alternative, more flexible approach.

Bin Smoothing Method

- The general idea of smoothing is to **group data points into strata** in **which the value of f(x)** can be **assumed to be constant**.
- We assume f(x) changes very slowly;
  - Almost constant in small windows of time.
- ***Example***: An example of this idea for the poll_2008 data is to assume that public opinion remained approximately the same within a week's time.

- If we fix a day to be in the center of our week, call it x_0, then for any other day x such that $|x - x_0| \leq 3.5$, we assume f(x) is a constant f(x)=μf(x). This assumption implies that:

$$E[Y_i|X_i = x_i] \approx \mu \text{ if } |x_i - x_0| \leq 3.5$$

- In smoothing, we call the size of the interval satisfying $|x_i - x_0| \leq 3.5$ the *window size*, *bandwidth* or *span*.

Bin Smoothing Method

- Assumption implies that a good estimate for $f(x)$ is the average of the Y_i values in the window.
- If we define A_0 as the set of indexes ii such that $|x_i - x_0| \leq 3.5$ and N_0 as the number of indexes in A_0, then our estimate:

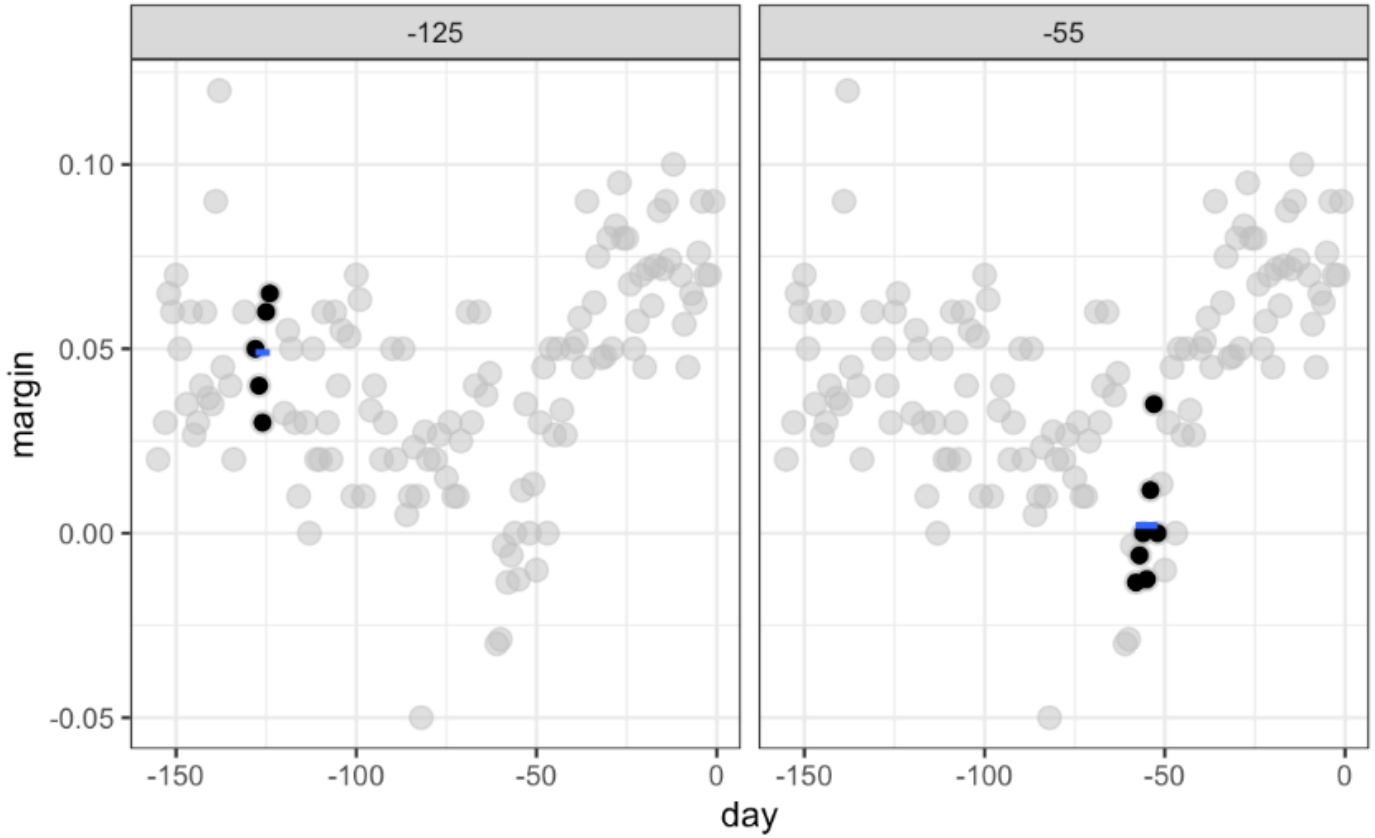
$$\hat{f}(x_0) = \frac{1}{N_0} \sum_{i \in A_0} Y_i$$

- The idea behind *bin smoothing* is to make this calculation with each value of x as the center.

- In the poll example, for each day, we would compute the average of the values within a week with that day in the center. Here are two examples: $x_0 = -125$ and $x_0 = -55$.
- The blue segment represents the resulting average.

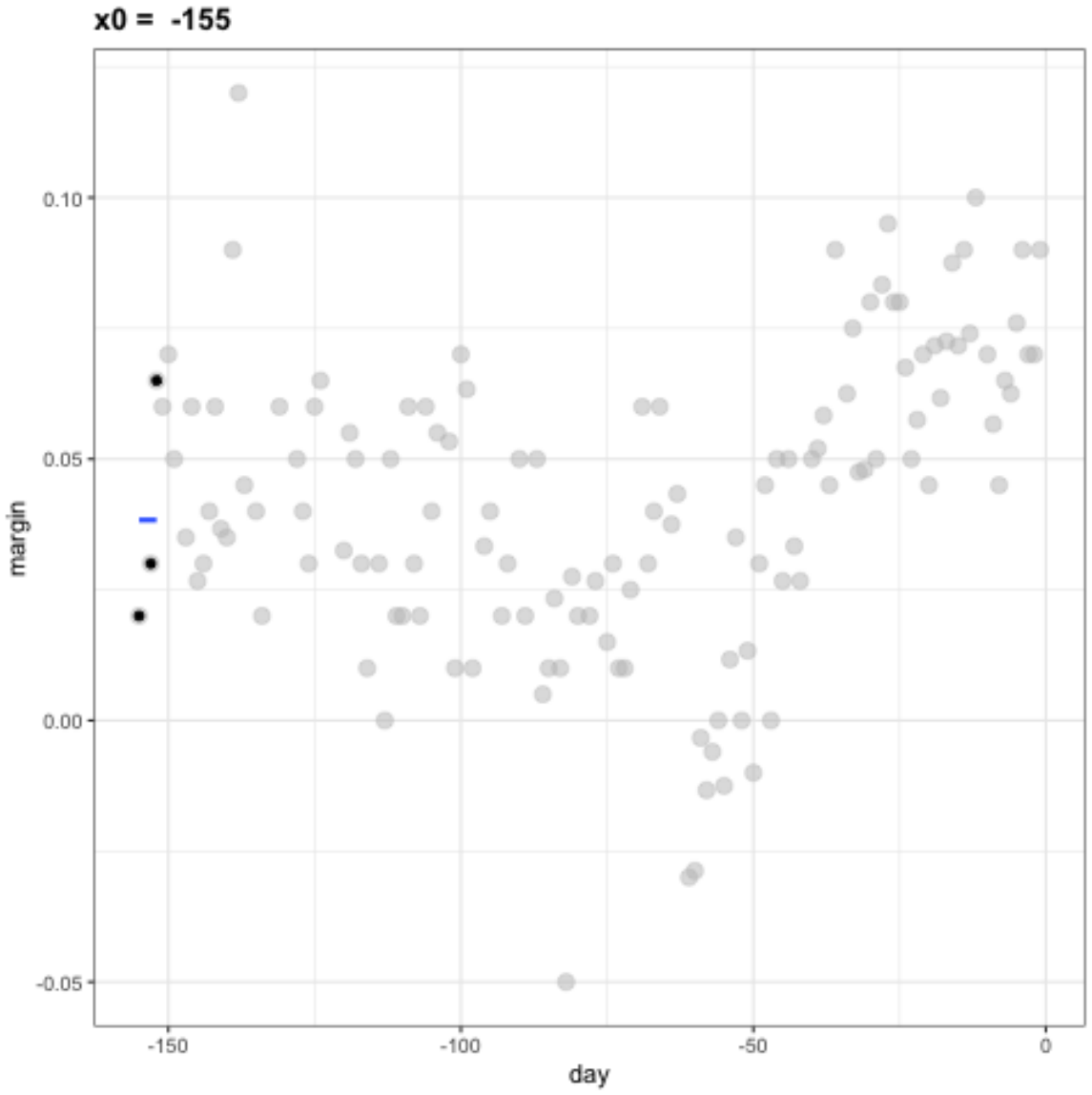
Bin Smoothing Method

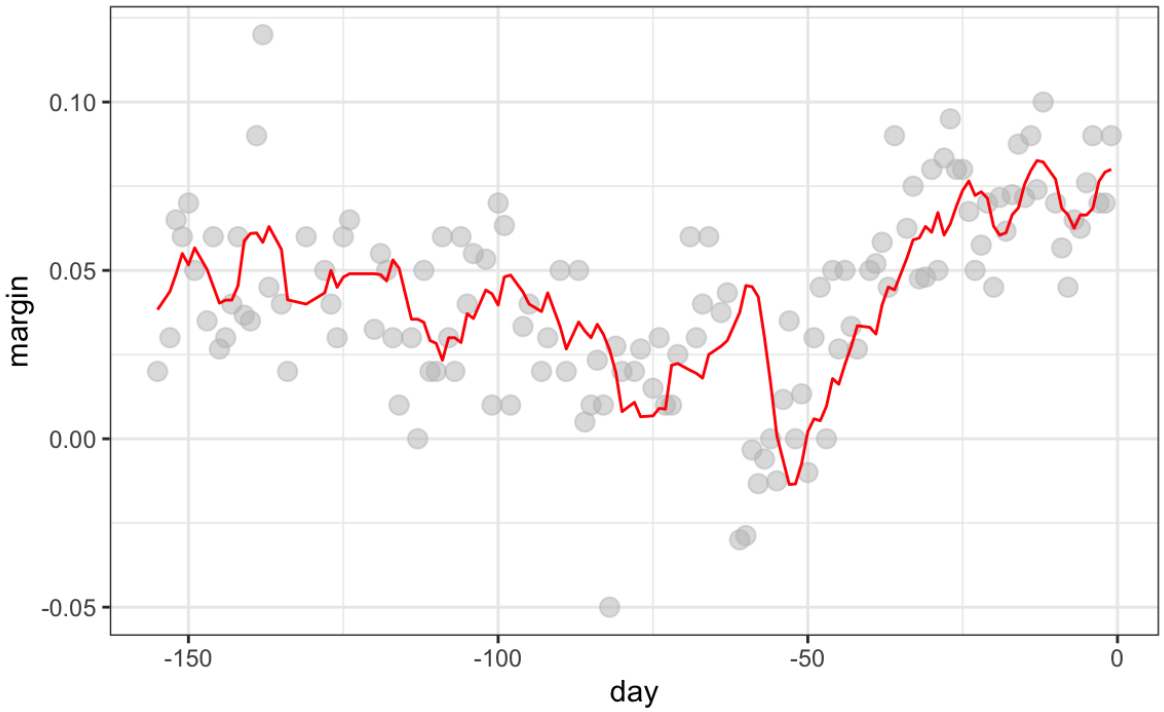
- In the poll example, for each day, we would compute the average of the values within a week with that day in the center. Here are two examples: $x_0 = -125$ and $x_0 = -55$.
- The blue segment represents the resulting average.



Data Transformation:

Bin Smoothing Method





using Kernals

- The final result from the bin smoother is quite wiggly.
- One reason for this is that each time the window moves, two points change.
- We can attenuate this somewhat by taking weighted averages that give the center point more weight than far away points, with the two points at the edges receiving very little weight.
- Think of the bin smoother approach as a weighted average:

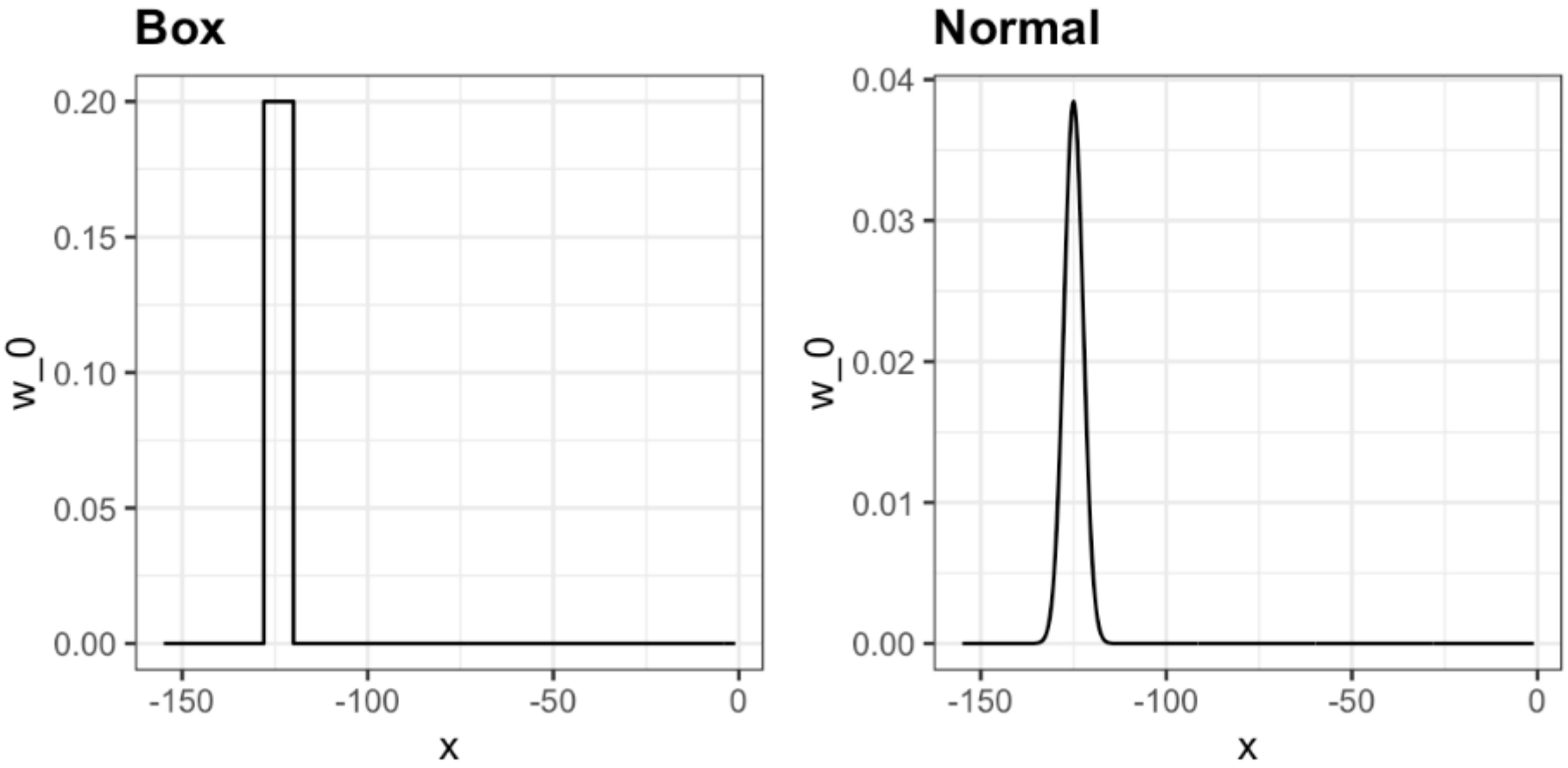
$$\hat{f}(x_0) = \sum_{i=1}^{N} w_0(x_i) Y_i$$

- in which each point receives a weight of either 0 or $1/N_0$, with N_0 the number of points in the week.

Data Transformation:

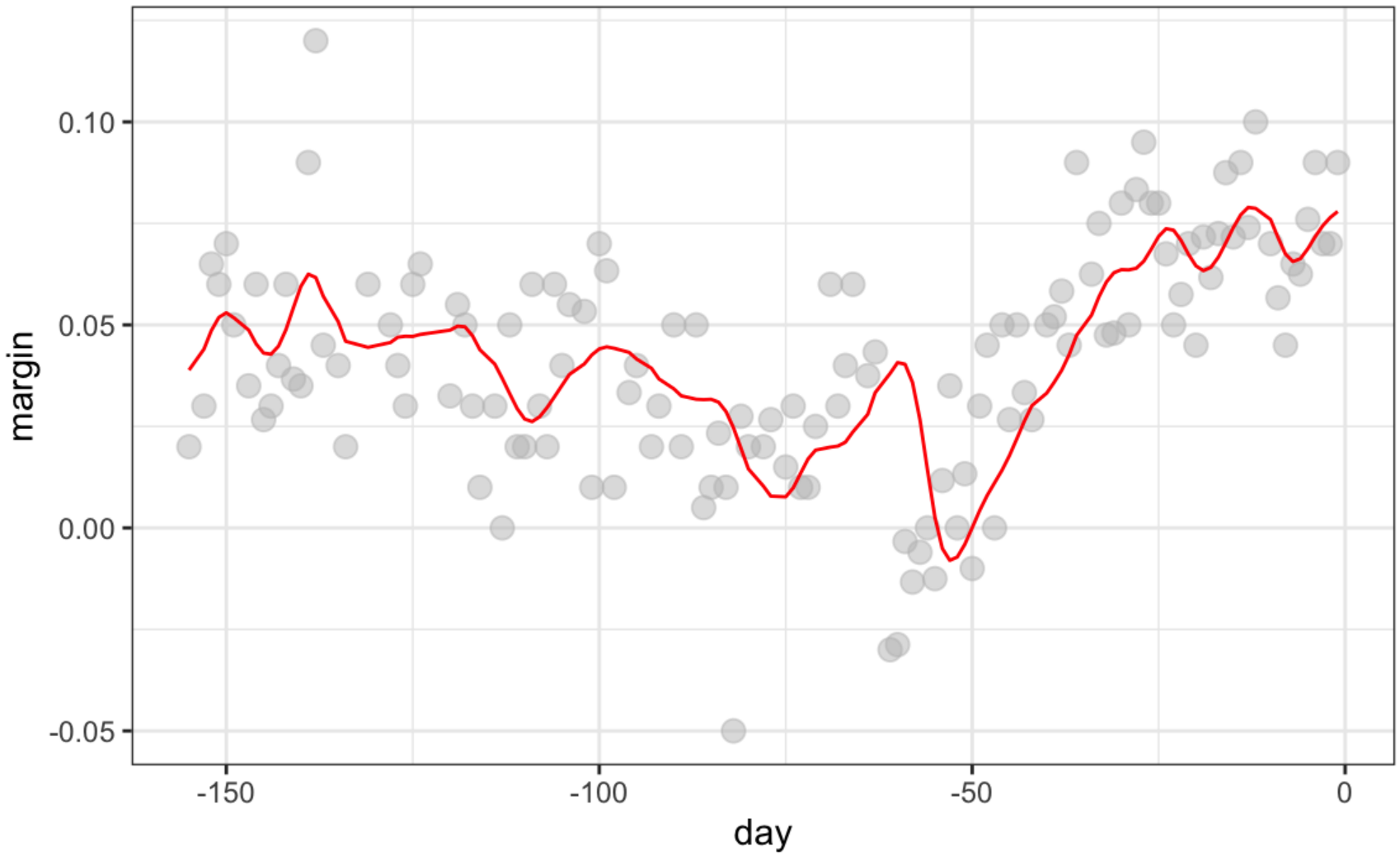
using Kernals

-



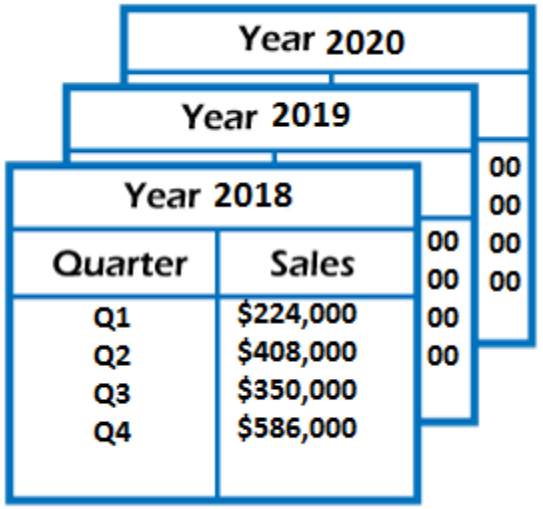
Data Transformation:

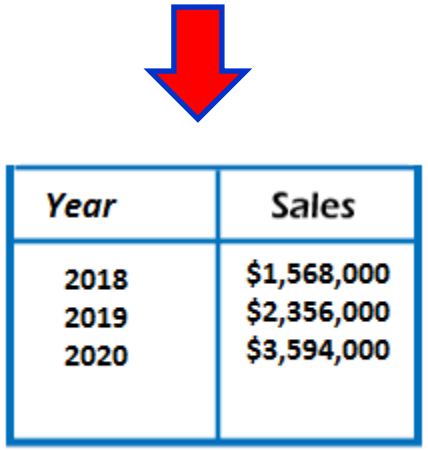
using Kernals



Data Reduction

- ***Choices*** is an ***important part*** of the ***human experience***.
- Too Much Data : Hinders Decision Making
  - Having too many options actually becomes a burden, slowing down decision-making capabilities.
  - Having too much data can *bog down cloud storage* and *choke high-performance computing workloads*.
- Data reduction: Obtain a reduced representation of the data set that is much ***smaller in volume*** but ***yet produces the same analytical results***
- **Why data reduction?** — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to **run** on the complete data set.





Year	Sales
2018	$1,568,000
2019	$2,356,000
2020	$3,594,000

Data Reduction

- *Data reduction strategies*
  - Dimensionality reduction, e.g., remove unimportant attributes
    - Wavelet transforms
    - Principal Components Analysis (PCA)
    - Feature subset selection, feature creation
  - Numerosity reduction
    - Regression and Log-Linear Models
    - Histograms, clustering, sampling
    - Data cube aggregation
    - Data compression

Data Reduction

Dimensionality Reduction:

- Approach attempts to **reduce the number of "dimensions,"** or aspects/variables, from a data set.
- Example: Summation, average, etc.

Data Cube Aggregation:

- **Aggregates multidimensional data** at various levels to create a "**data cube**" (or multidimensional data object).
- Data is processed and reduced to a smaller but equally useful form of information that still ***represents trends in relevant information*** for analytics.

Numerosity Reduction:

- Is replacing the **original data with a smaller form** of data **representing** the original with more or less fidelity.
- Common compression technique is used: audio, video, & image.

Clustering:

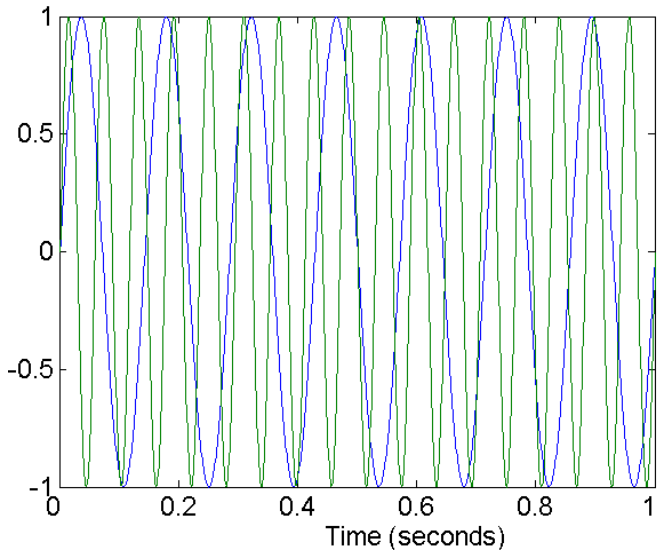
- Uses data attributes to build **set of clusters** in which data is split.
- **Similarities** and **dissimilarities** between the data objects result in different placements and distances between the clusters and the objects as a whole.

Dimensionality Reduction

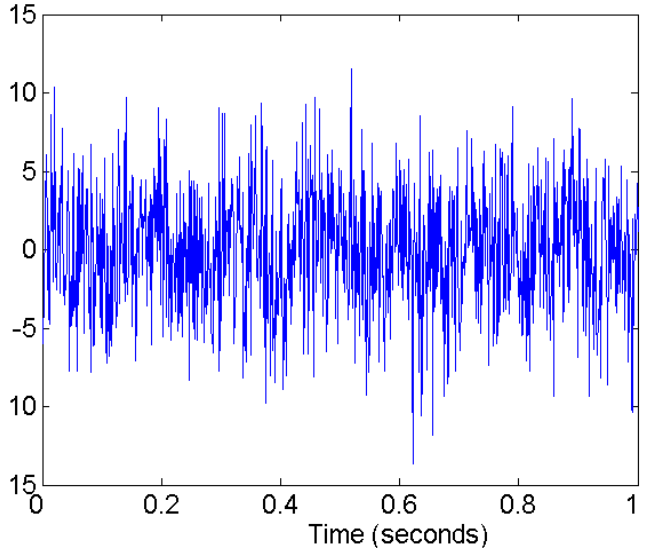
- **Curse of dimensionality**
  - When dimensionality increases, data becomes increasingly sparse
  - *Density* and *distance between points*, which is critical to clustering, outlier analysis, *becomes less meaningful*
  - The possible combinations of subspaces will grow exponentially
- **Dimensionality reduction**
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization
- **Dimensionality reduction techniques**
  - Wavelet transforms
  - Principal Component Analysis
  - Supervised and nonlinear techniques (e.g., feature selection)

Dimensionality Reduction

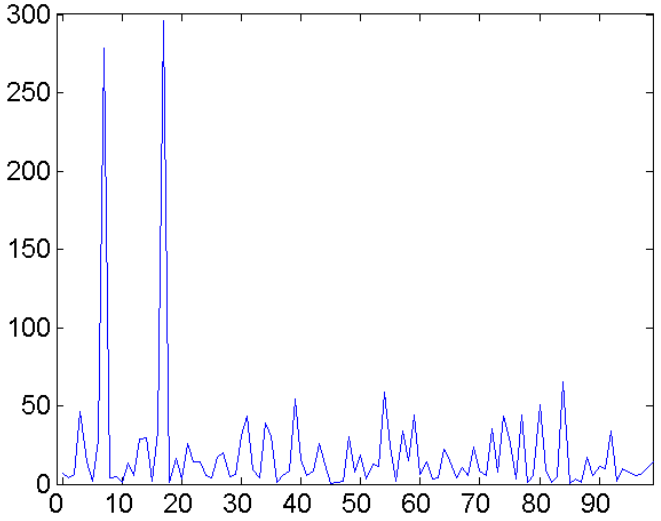
- **Mapping Data to a New Space**
  - **Fourier Transformation**
  - **Wavelet Transformation**



Two Sine Waves



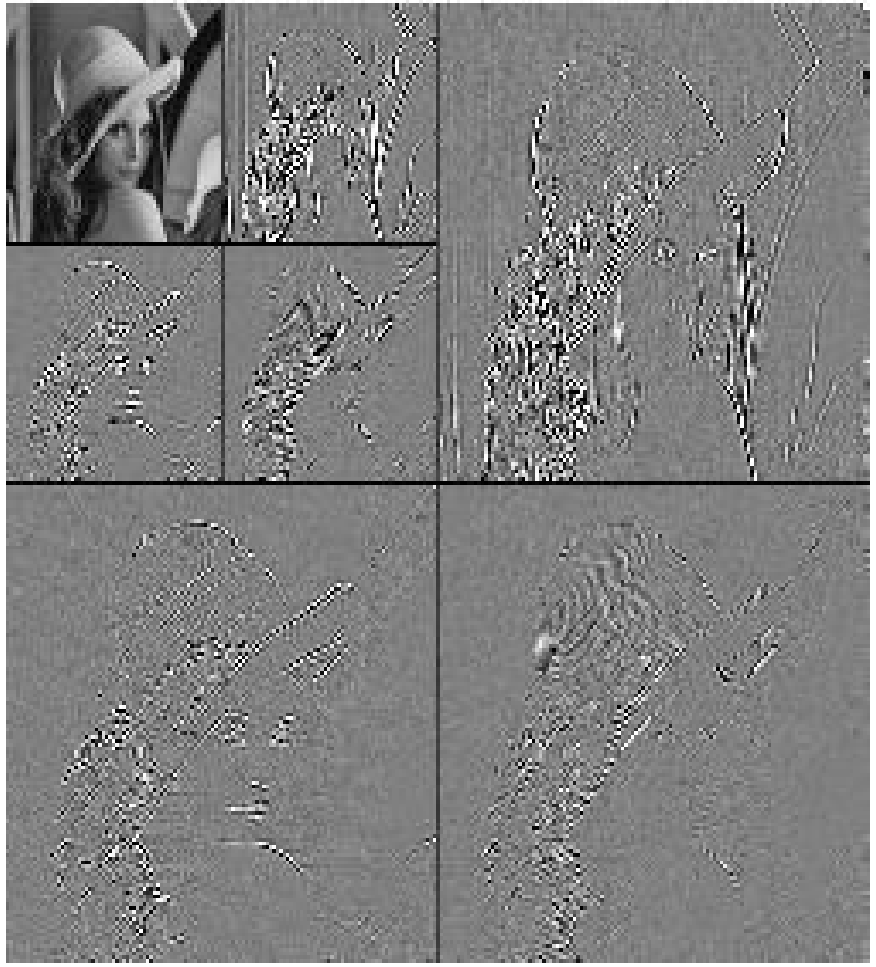
Two Sine Waves + Noise



Frequency

Dimensionality Reduction

- **Wavelet Transform**
  - **Decomposes a signal into different frequency subbands**
  - **Applicable to n-dimensional signals**
  - **Data are transformed to preserve relative distance between objects at different levels of resolution**
  - **Allow natural clusters to become more distinguishable**
  - **Used for image compression**



Data Transformation

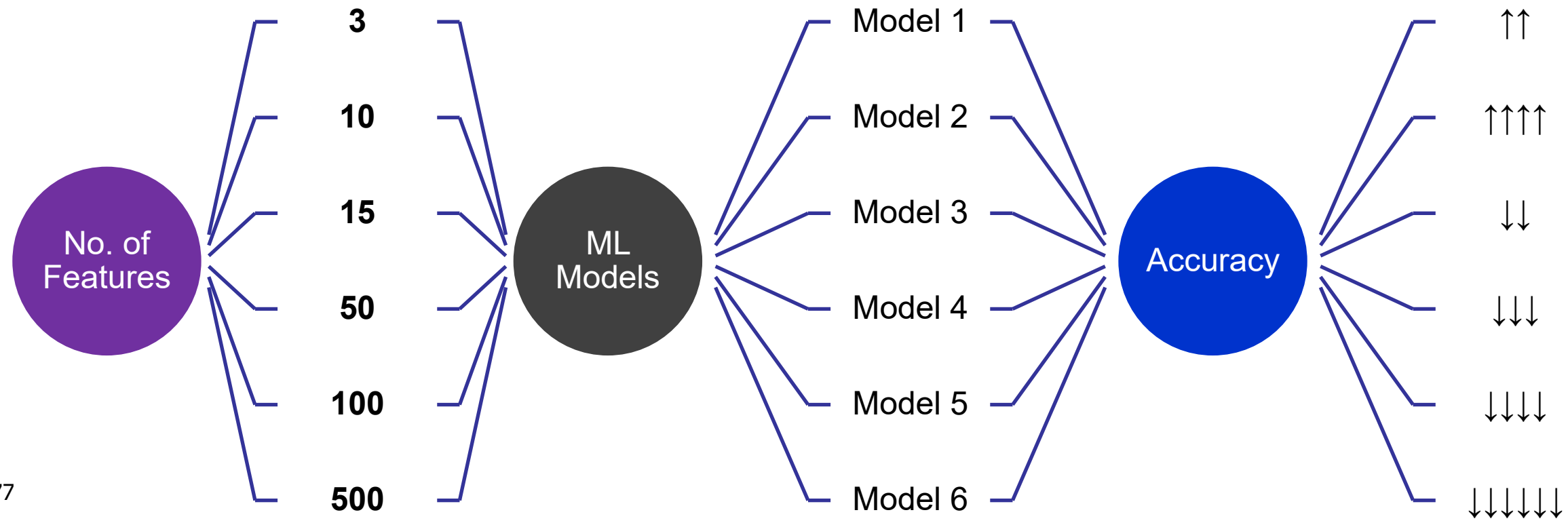
- Benefits:
  - Data is transformed to make it better-organized.
  - Properly formatted and validated data improves data quality.
  - Data transformation facilitates compatibility between applications, systems, and types of data.
- Challenges:
  - Data transformation can be expensive (due to computing resources, licensing cost, etc.). It can be resource-intensive.
  - Lack of expertise and carelessness can introduce problems during transformation.

Data Reduction

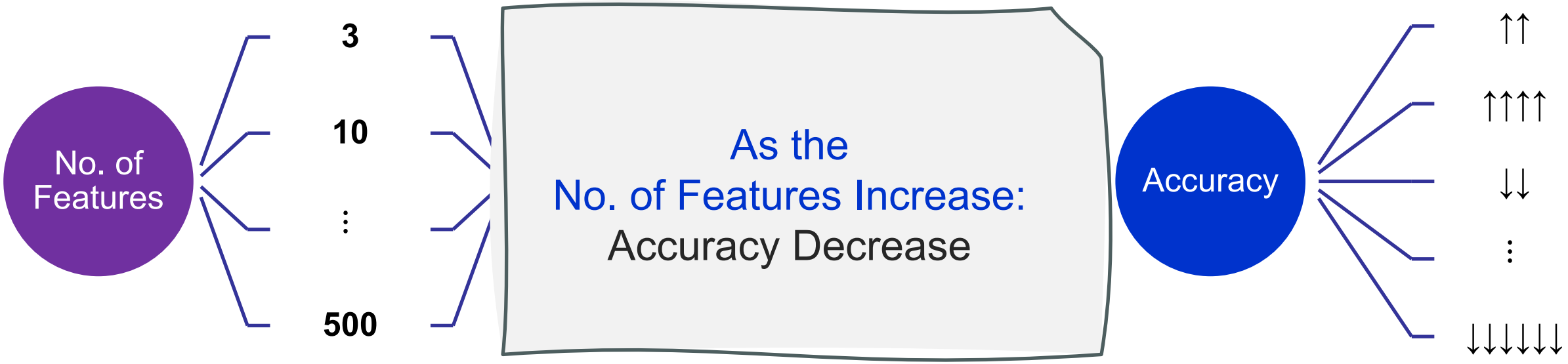
- Data Reduction: *Process* of *transforming numerical* or *alphabetical* digital information derived empirically or experimentally into a *corrected*, *ordered*, and *simplified form*.
- Concept: Reduction of vast amounts of data down to the meaningful parts.
- In the case of discrete data, reduction involves *smoothing* and *interpolation*. In the case of digital data, reduction involves some *editing*, *scaling*, *encoding*, *sorting*, *collating*, etc.
- Benefits of data reduction:
  - Reduced representation of data will be smaller in volume but produces the same (or similar) analytical results;
  - Data analytics/mining on the reduced data set consumes much lesser time compared to that done on the complete data set.

Data Reduction

- **Curse of Dimensionality**
  - Dimensionality : Number of features we have in the dataset.
  - Assume: for a Housing Price Prediction; we have 500 features

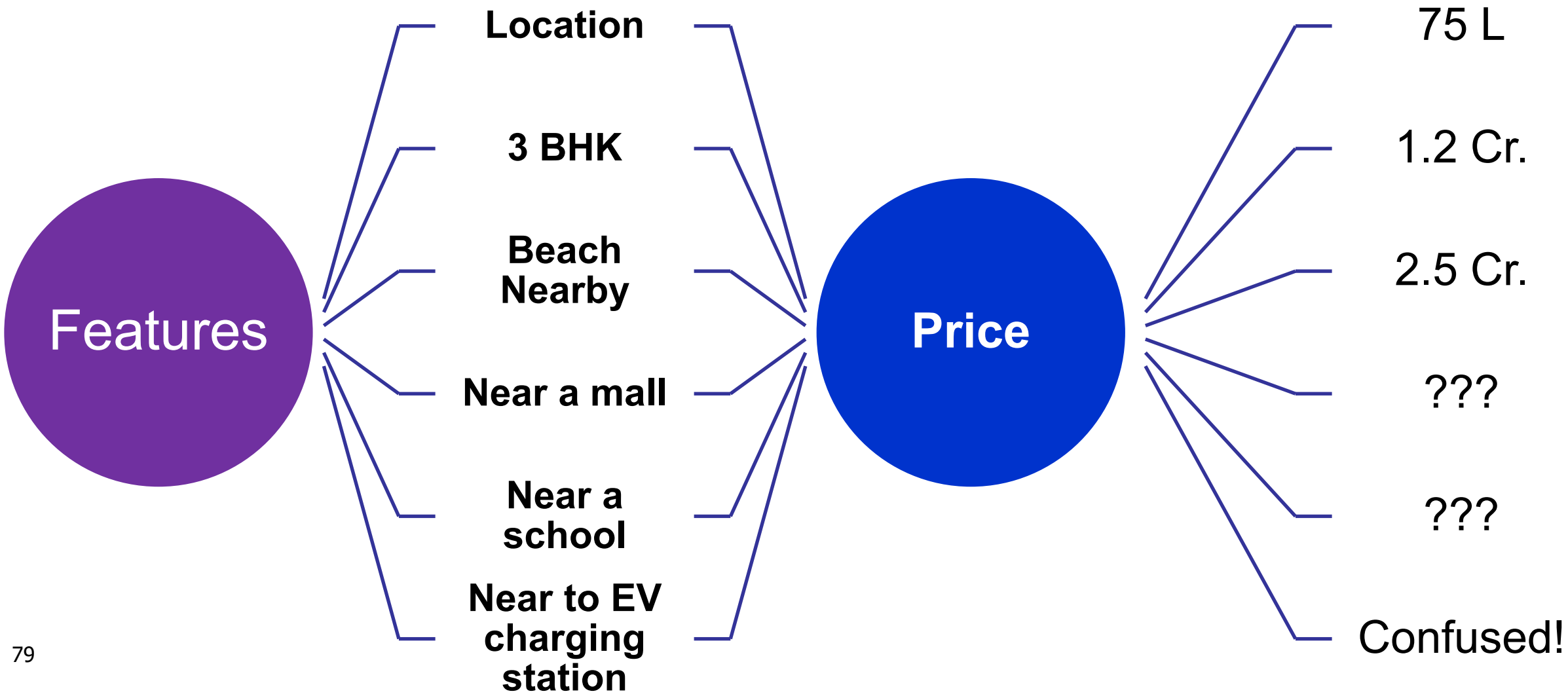


Data Reduction



- **Model Performance Degrades**
  - **Reason:** the model would have to do a lot of calculations

Data Reduction



Data Reduction

- Two Ways to Address the Curse of Dimensionality
  - **Feature Selection**
    - We try to consider the **important features** from our dataset and
    - Then **train** our model
  - **Dimensionality Reduction**
    - *Feature extraction*
      - Derive a feature from the set of features, where the essence of the previous features will be captured
      - E.g.: $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \dots, \boldsymbol{f}_7\} \rightarrow \{\boldsymbol{d}_1, \boldsymbol{d}_2\}$

Why Dimensionality Reduction

- Prevent $\longrightarrow$ Curse of Dimensionality
- Improve Performance of the Model
- Better Visualization
  - How to visualize 500 features?

Principal Component Analysis (PCA)

- ***Summary:***
- As we keep on increasing the number of features,
- At the time of training our models; at one point of time; our model might get confused; as it has so many features to learn
- Because of that; the Performance of the Model will **Degrade**.

Covariance Analysis

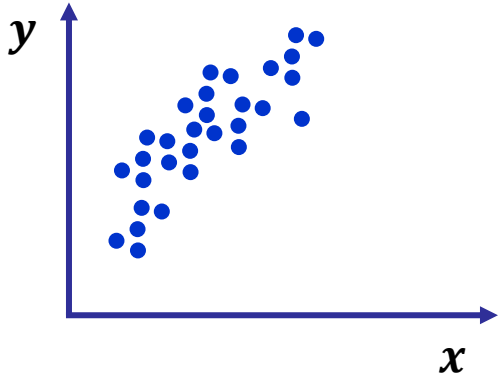
- Feature Selection

x	y
i_1	o_1
i_2	o_2
i_3	o_3
$\vdots$	$\vdots$
i_n	o_m

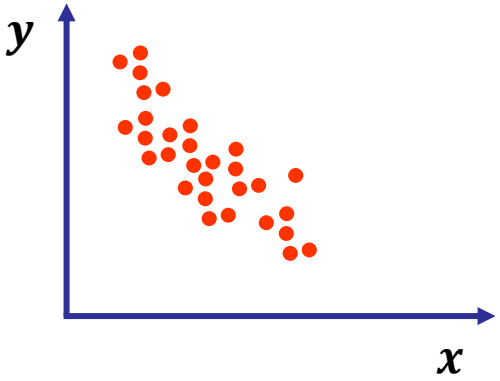
$x \uparrow$	$y \uparrow$
$x \downarrow$	$y \downarrow$

$x \uparrow$	$y \downarrow$
$x \downarrow$	$y \uparrow$

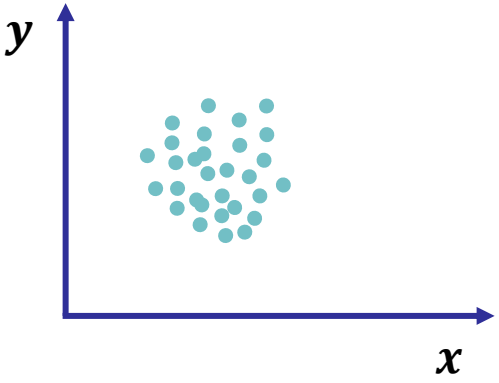
Linear relationship
(+ve Cov)



Inverse
Linear relationship
(-ve Cov)



***No** relationship*
(0 Cov)



Covariance Analysis
(to find mathematical relationship between x & y)

Covariance Analysis

- Feature Selection
  - For features with very **high covariance**
    - These set of features are important
    - As this feature is directly influencing the output
  - If Covar. Is **ZERO**
    - There is no relation between feature and can be neglected

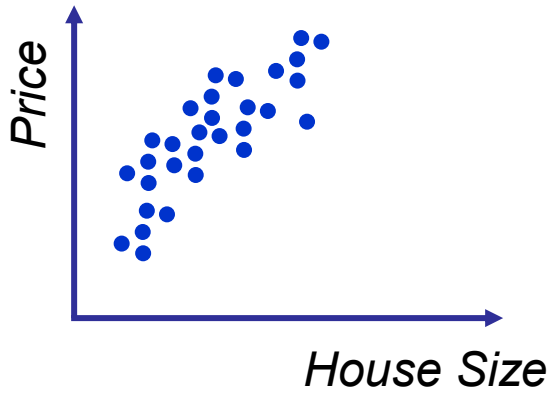
Pearson Correlation

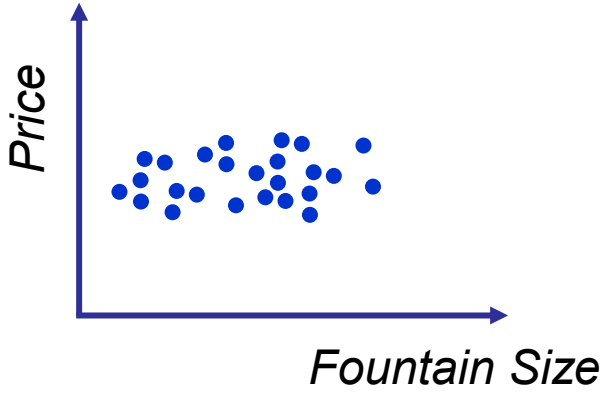
$$\textbf{Pearson Correaltion} = \frac{\boldsymbol{cov(x, y)}}{\boldsymbol{\sigma_x * \sigma_y}} = [\boldsymbol{-1\ to + 1}]$$

- If PC is more inclined towards + 1 $\longrightarrow$ x, y more Positive Correlation
- If PC is more inclined towards - 1 $\longrightarrow$ x, y more Negative Correlation

Feature Selection

- **Example**: Housing Dataset
  - Features
    - House Size
    - No. of Rooms
  - Output Feature
    - Price





Feature Extraction

- **Example**: Housing Dataset
  - Independent Features
    - No. of Rooms
    - Room Size
  - Output Feature
    - Price

Reduce the Features to ONE from TWO

Apply some Transform

Generate New Feature

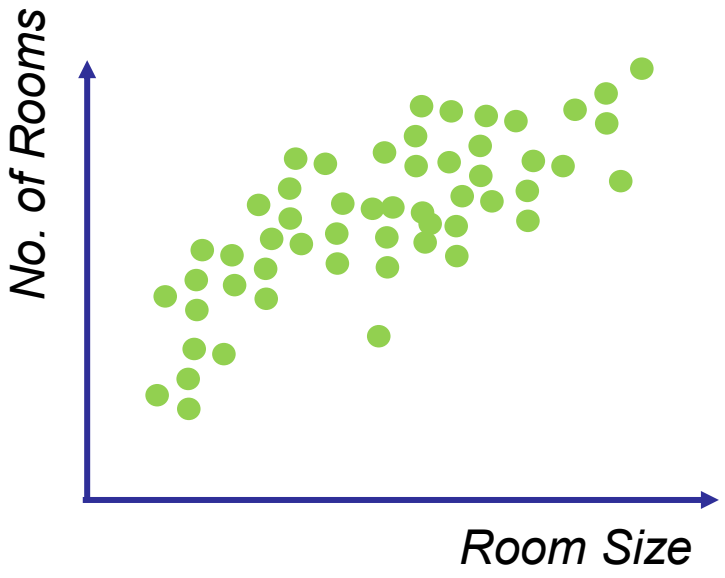
Predict

House Size

Feature Extraction

- **Example**: Housing Dataset

Room Size	No. of Rooms	Price



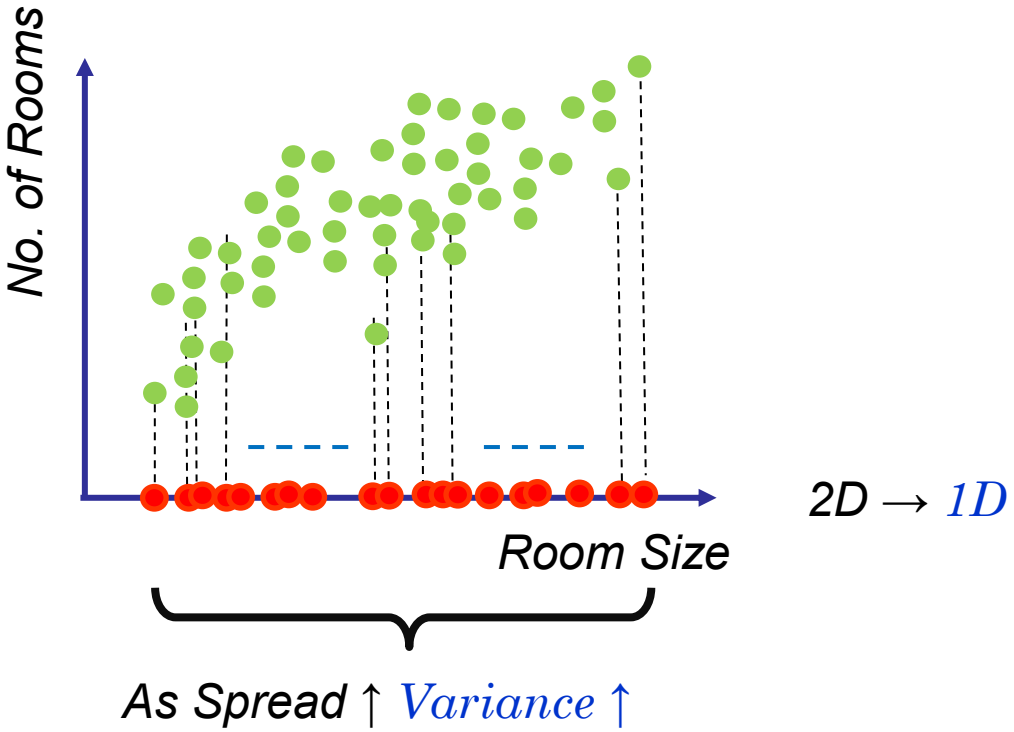
Reduce TWO Features
- *Room Size*
- *No. of Rooms*

} *House Size*

Feature Extraction

- **Example**: Housing Dataset

Room Size	No. of Rooms	Price

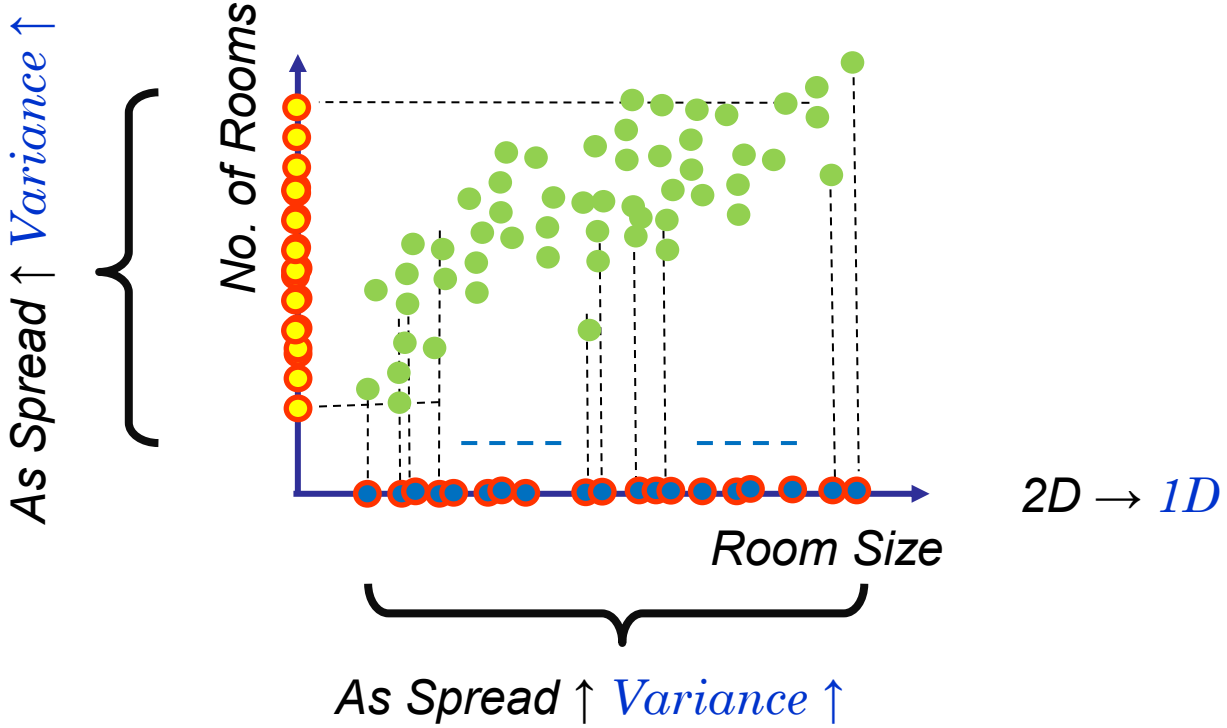


Feature Extraction

- **Example**: Housing Dataset

Room Size	No. of Rooms	Price

Information is Lost if only 1D is considered



Feature Extraction

- **Example**: Housing Dataset

Room Size	No. of Rooms	Price

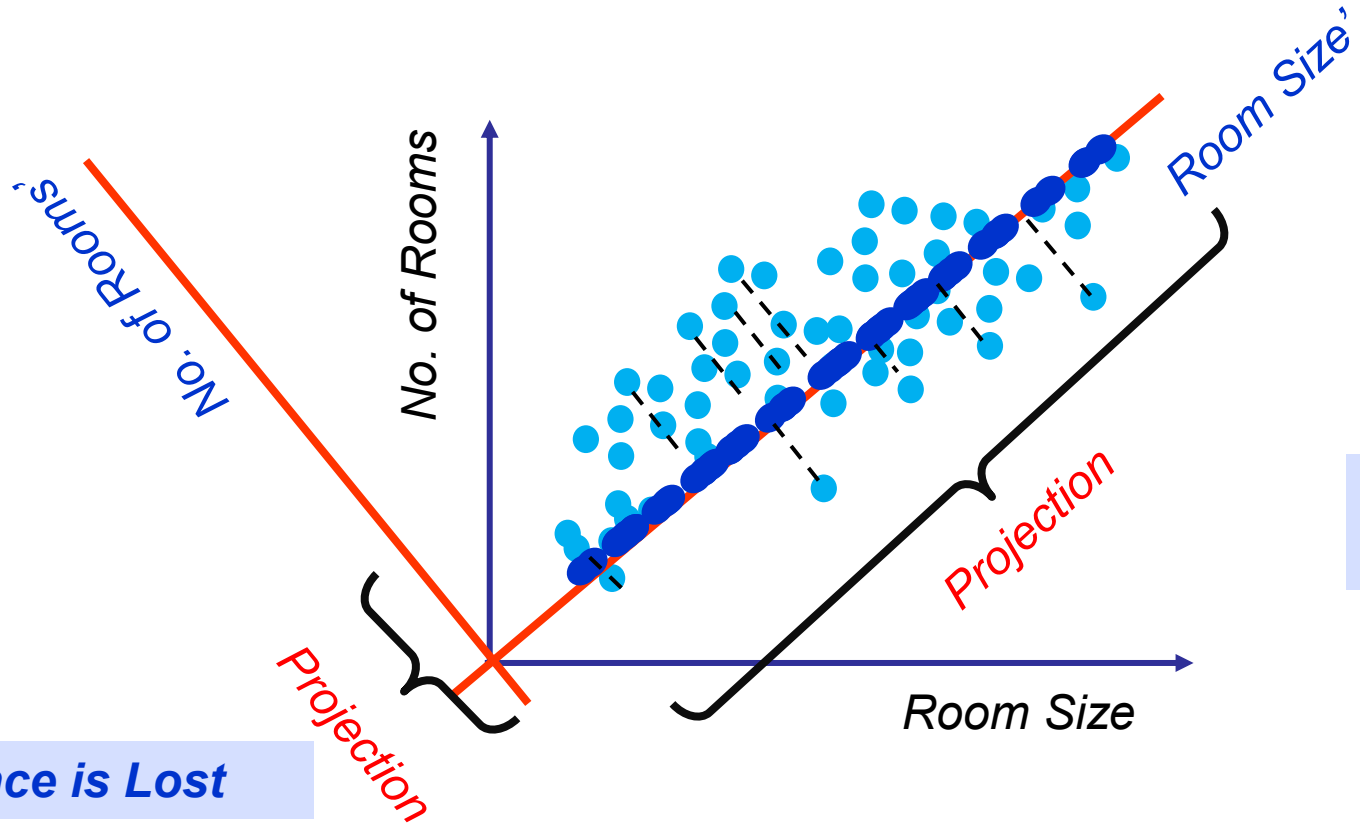
In $n_D \rightarrow m_D$; $n >>> m$ *conversion*

If so much Information is Lost

Model might give **WRONG** Predictions

Principal Component Analysis (PCA)

- *In $n_D \rightarrow m_D$; $n >>> m$ conversion apply some transformation{Eigen Decomposition} is applied on some matrix.*

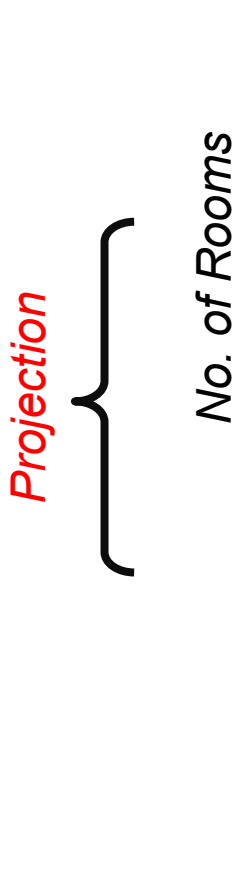


Maximum Variance is Captured

Least Variance is Lost

Principal Component Analysis (PCA)

Maximum Variance is Captured on X Axis
Least is lost on Y Axis

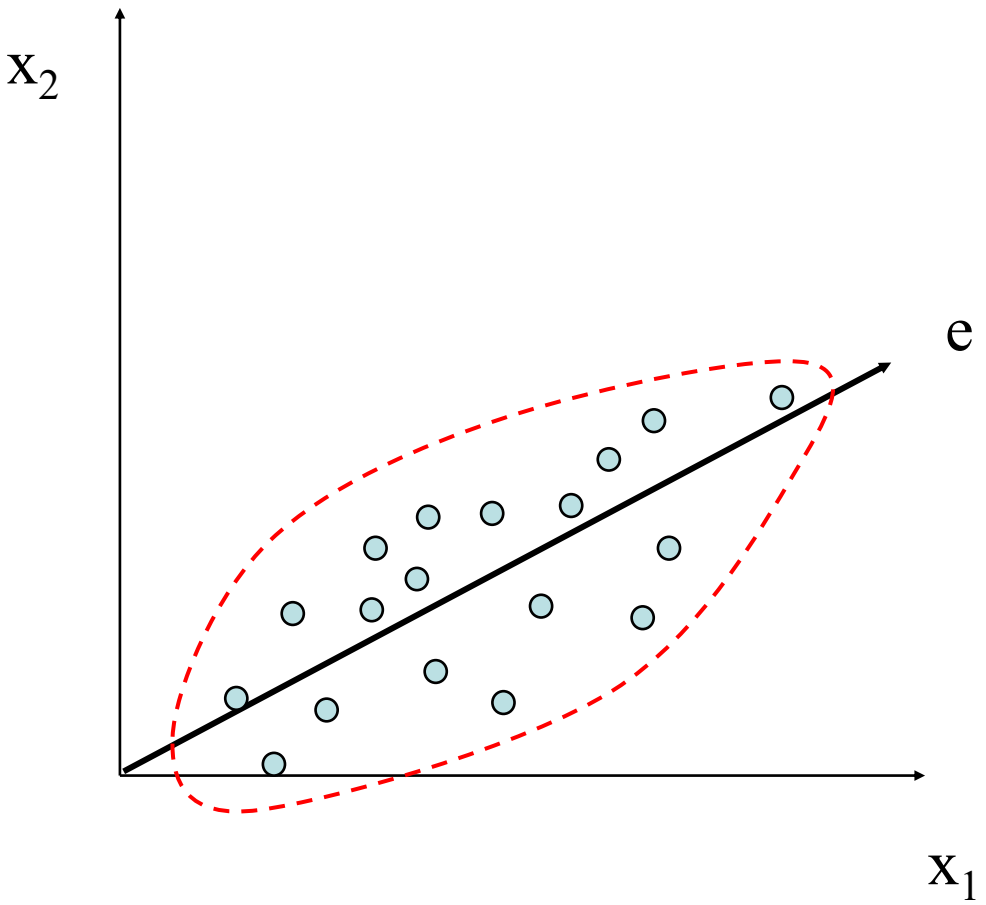


Variance : **PC1 > PC2 > PC3 > PC4 …**

Best PC : **Captures Max Variance**

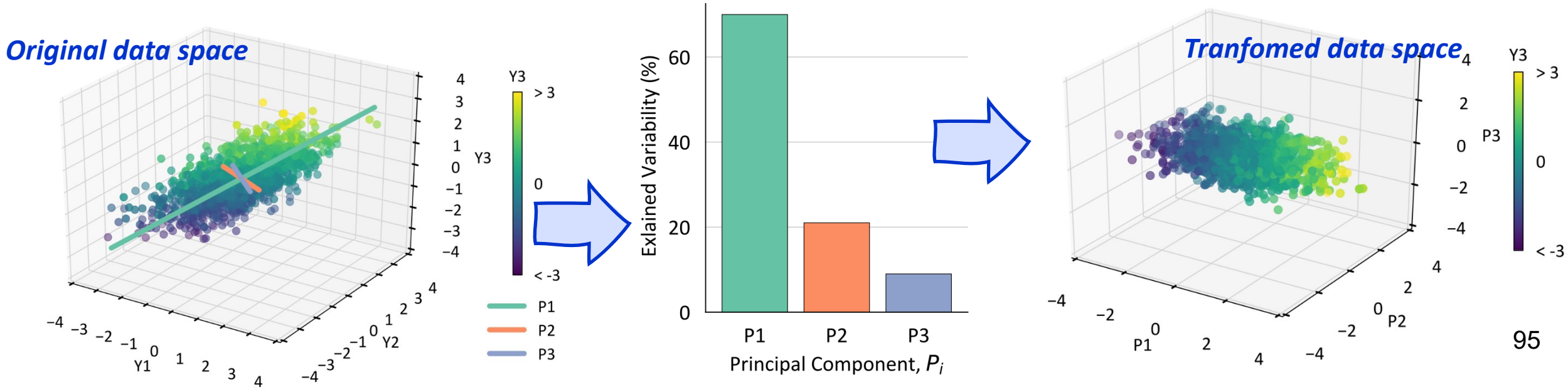
Principal Component Analysis (PCA)

- Find a *projection* that *captures* the *largest amount* of *variation* in data
- The *original data* are *projected* onto a *much smaller space*, resulting in dimensionality reduction.
- We find the eigenvectors of the covariance matrix, and these <u>eigenvectors define the new space</u>



Principal Component Analysis (PCA)

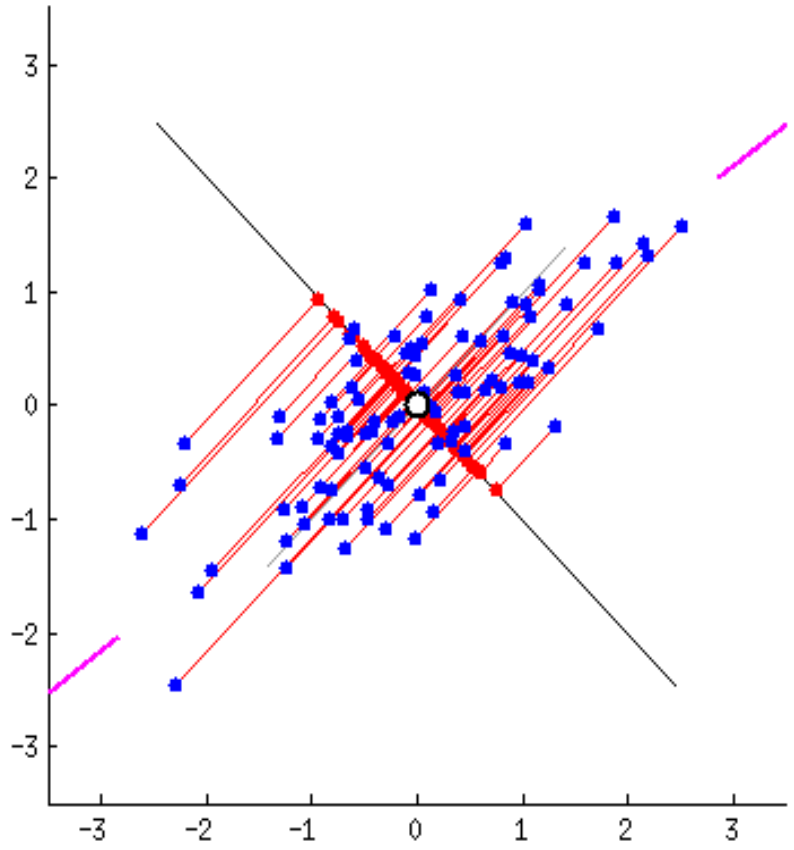
- PCA transforms the variables into a new set of variables called as principal components (PC).
- PCs are linear combination of original variables and are orthogonal.
  - 1st PC accounts for most of the possible variation of original data.
  - 2nd PC does its best to capture the variance in the data.
    - There can be only 2 PC for a 2D data set.



Principal Component Analysis (PCA)

The steps involved in PCA Algorithm are as follows-

- ❑ Get data.
- ❑ Compute the mean vector (μ).
- ❑ Subtract mean from the given data.
- ❑ Calculate the covariance matrix.
- ❑ Calculate the eigen vectors and eigen values of the covariance matrix.
- ❑ Choosing components and forming a feature vector.
- ❑ Deriving the new data set.



PCA : Example

Consider the two dimensional patterns (2, 1), (3, 5), (4, 3), (5, 6), (6, 7), (7, 8).
Compute the principal component using PCA Algorithm.

$x_1 = (2,1)$
$x_2 = (3,5)$
$x_3 = (4,3)$
$x_4 = (5,6)$
$x_5 = (6,7)$
$x_6 = (7,8)$

x_1	x_2	x_3	x_4	x_5	x_6
$\begin{bmatrix}2\\1\end{bmatrix}$	$\begin{bmatrix}3\\5\end{bmatrix}$	$\begin{bmatrix}4\\3\end{bmatrix}$	$\begin{bmatrix}5\\6\end{bmatrix}$	$\begin{bmatrix}6\\7\end{bmatrix}$	$\begin{bmatrix}7\\8\end{bmatrix}$

PCA : Example

Compute Mean

$$\textbf{Mean } \boldsymbol{\mu} = \left[\left(\frac{2+3+4+5+6+7}{6}\right), \left(\frac{1+5+3+6+7+8}{6}\right)\right]$$

Mean Vector $\boldsymbol{\mu}$
$\begin{bmatrix} 4.5 \\ 5 \end{bmatrix}$

PCA : Example (… contd.)

Subtract Mean from the Vector

$$x_1 - \mu = (2 - 4.5, \quad 1 - 5)$$
$$x_2 - \mu = (3 - 4.5, \quad 5 - 5)$$
$$x_3 - \mu = (4 - 4.5, \quad 3 - 5)$$
$$x_4 - \mu = (5 - 4.5, \quad 6 - 5)$$
$$x_5 - \mu = (6 - 4.5, \quad 7 - 5)$$
$$x_6 - \mu = (7 - 4.5, \quad 8 - 5)$$

$x_1{}'$	$x_2{}'$	$x_3{}'$	$x_4{}'$	$x_5{}'$	$x_6{}'$
$\begin{bmatrix} -2.5 \\ -4 \end{bmatrix}$	$\begin{bmatrix} -1.5 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -0.5 \\ -2 \end{bmatrix}$	$\begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1.5 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 2.5 \\ 3 \end{bmatrix}$

PCA : Example (… contd.)

Compute Covariance Matrix

$$\textbf{Covariance Matrix} = \frac{\sum(\boldsymbol{x_i} - \boldsymbol{\mu}) * (\boldsymbol{x_i} - \boldsymbol{\mu})^{\boldsymbol{T}}}{\boldsymbol{n}}$$

PCA : Example (… contd.)

Compute Covariance Matrix

$$\textbf{Covariance Matrix} = \frac{\sum(\boldsymbol{x_i} - \boldsymbol{\mu}) * (\boldsymbol{x_i} - \boldsymbol{\mu})^{\boldsymbol{T}}}{\boldsymbol{n}}$$

$$\boldsymbol{m_1} = (\boldsymbol{x_1} - \boldsymbol{\mu}) * (\boldsymbol{x_1} - \boldsymbol{\mu})^{\boldsymbol{T}}$$
$$= \begin{bmatrix} -2.5 \\ -4 \end{bmatrix} * \begin{bmatrix} -2.5 & -4 \end{bmatrix} = \begin{bmatrix} 6.25 & 10 \\ 10 & 16 \end{bmatrix}$$

$$\boldsymbol{m_2} = \begin{bmatrix} -1.5 \\ 0 \end{bmatrix} * \begin{bmatrix} -1.5 & 0 \end{bmatrix} = \begin{bmatrix} 2.25 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\boldsymbol{m_3} = \begin{bmatrix} -0.5 \\ -2 \end{bmatrix} * \begin{bmatrix} -0.5 & -2 \end{bmatrix} = \begin{bmatrix} 0.25 & 1 \\ 1 & 4 \end{bmatrix}$$

$$\boldsymbol{m_4} = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} * \begin{bmatrix} 0.5 & 1 \end{bmatrix} = \begin{bmatrix} 0.25 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

$$\boldsymbol{m_5} = \begin{bmatrix} 1.5 \\ 2 \end{bmatrix} * \begin{bmatrix} 1.5 & 2 \end{bmatrix} = \begin{bmatrix} 2.25 & 3 \\ 3 & 4 \end{bmatrix}$$

$$\boldsymbol{m_6} = \begin{bmatrix} 2.5 \\ 3 \end{bmatrix} * \begin{bmatrix} 2.5 & 3 \end{bmatrix} = \begin{bmatrix} 6.25 & 7.5 \\ 7.5 & 9 \end{bmatrix}$$

PCA : Example (… contd.)

Compute Covariance Matrix

$$\textbf{Covariance Matrix} = \frac{\boldsymbol{m_1 + m_2 + m_3 + m_4 + m_5 + m_6}}{\boldsymbol{n}}$$

$$\boldsymbol{CM} = \frac{1}{6} * \begin{bmatrix} 17.5 & 22 \\ 22 & 34 \end{bmatrix} = \begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix}$$

PCA : Example (… contd.)

Compute Eigen Values & Vectors of CM

$$|CM - \lambda I| = 0$$

$$\begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = 0$$

$$\begin{bmatrix} 2.92 - \lambda & 3.67 \\ 3.67 & 5.67 - \lambda \end{bmatrix} = 0$$

PCA : Example (… contd.)

Solving the equations

$$(2.92 - \lambda)(5.67 - \lambda) - (3.67 * 3.67) = 0$$

$$\lambda^2 - 8.59\lambda + 3.09 = 0$$

Solving the quadratic equations; we obtain the Eigen values

$\lambda_1 = 8.22$
$\lambda_2 = 0.38$

2nd Eigen Value is *SMALLER* as compared to the 1st.
Thus can be Left Out.

PCA : Example (… contd.)

Obtaining Eigen Vectors

Using the following equation, the Eigen Vector can be determined:

$$MX = \lambda X$$

Where:
M is the Covar. Matrix.
X is the Eigen Vector
λ is Eigen Value

$$\Rightarrow \begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 8.22 * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Solving:

$$2.92x_1 + 3.67x_2 = 8.22x_1$$

$$3.67x_1 + 5.67x_2 = 8.22x_2$$

PCA : Example (… contd.)

Simplifying:

$$5.3x_1 = 3.67x_2 \qquad \text{eq. (1)}$$

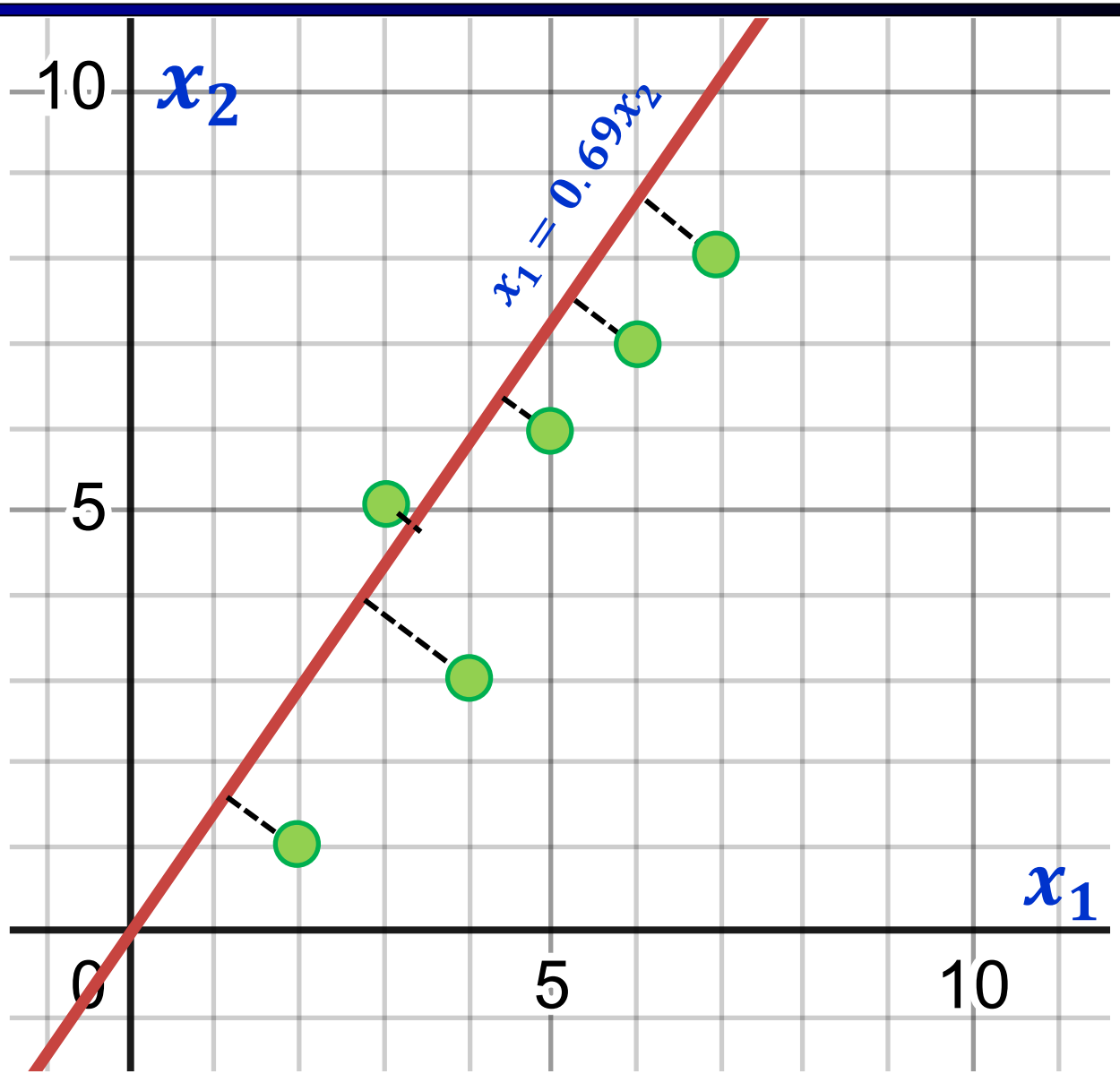
$$3.67x_1 = 2.55x_2 \qquad \text{eq. (2)}$$

From above equations:

$$x_1 = 0.69x_2$$

From eq. (2), the Eigen Vector is:

$$\Rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}$$



PCA : Example 2

Using the PCA algo., transform the given pattern (2,1) onto the Eigen vector obtained in prev. example.

Given

Eigen Vector:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}$$

Mean Vector: $= \begin{bmatrix} 4.5 \\ 5 \end{bmatrix}$

Feature Vector: $= \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

PCA : Example 2

Using the PCA algo., transform the given pattern (2,1) onto the Eigen vector obtained in prev. example.

To transform the feature vector:

Transpose of Eigen Vector x (Feature Vector – Mean Vector)

$$= \begin{bmatrix} 2.55 \\ 3.67 \end{bmatrix}^T * \left(\begin{bmatrix} 2 \\ 1 \end{bmatrix} - \begin{bmatrix} 4.5 \\ 5 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 2.55 & 3.67 \end{bmatrix} * \begin{bmatrix} -2.5 \\ -4 \end{bmatrix}$$

$$= -21.055$$