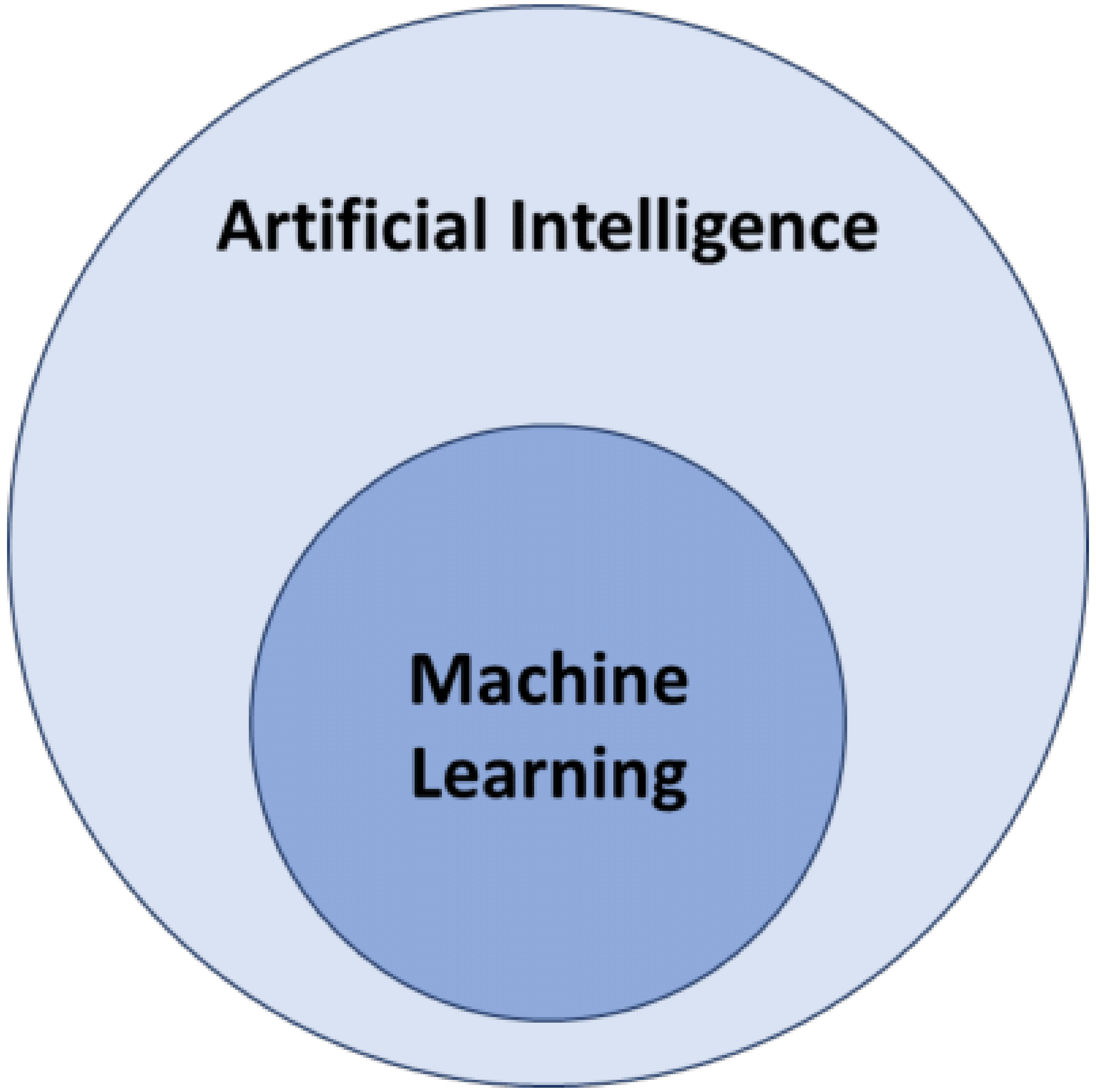
What is AI?

- The ability of computer systems to perform tasks that typically require human intelligence

- Examples: Pathfinding algorithms, adverserial search,   chat bots, image recognition, recommendation systems.

Machine Learning

- Subset of AI
- Focuses on developing algorithms and models that learn from data
- **Data-driven**
- No explicit programming required
- Examples : chat-bot, image recognition, spam email filtering, language translation



ML v traditional programming

ML algorithms:

- Learn pattern from data
- Training data is a must
- Generally, improves itself automatically
- Often a blackbox (hard to interpret)

Learning

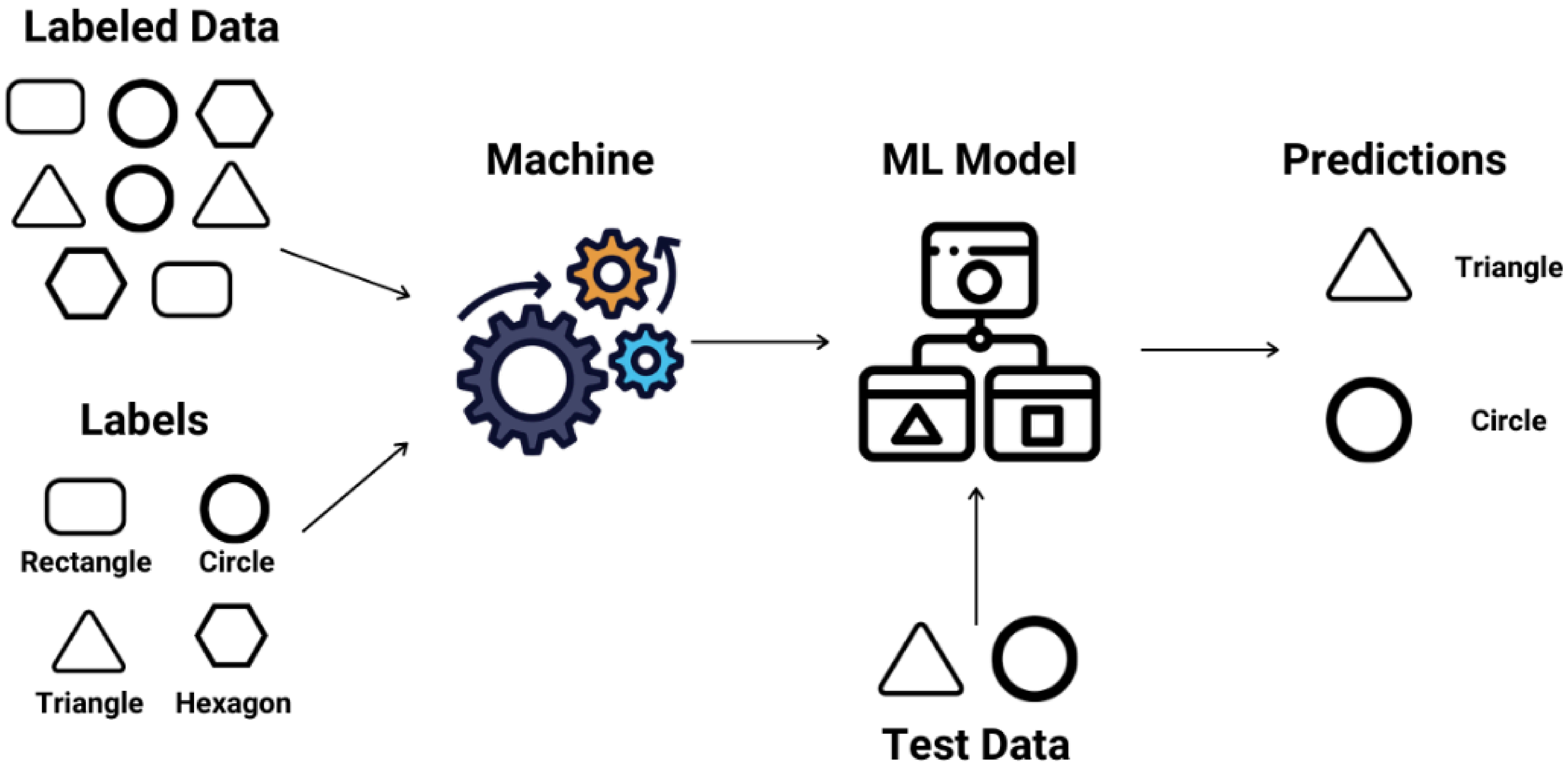
- Looking at data and finding patters. (adjusting parameters)

Types:

- Supervised
- Unsupervised
- Reinforcement

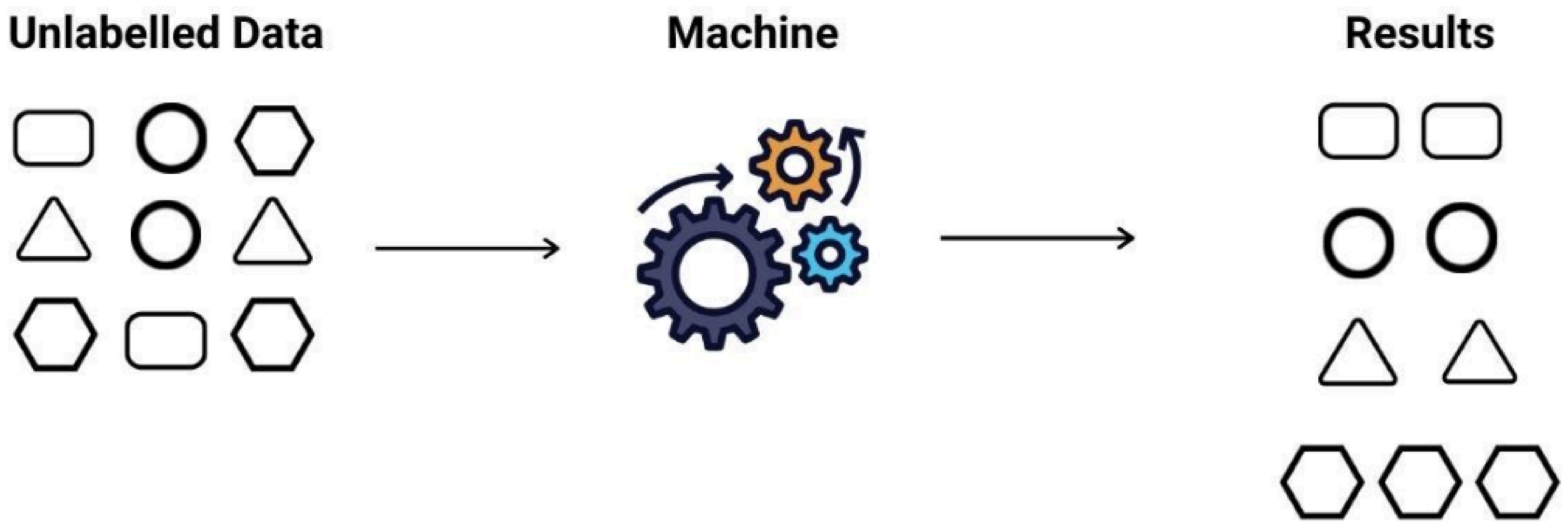
Supervised Learning

- Uses labelled data
- Goal: Map inputs to outputs by minimizing prediction errors.
- Uses: Image classification, spam detection, price prediction.
- Algorithms: Linear regression, decision trees, neural networks, KNN.


Labeled Data
Machine
ML Model
Predictions
Triangle
Circle
Labels
Rectangle
Circle
Triangle
Hexagon
Test Data

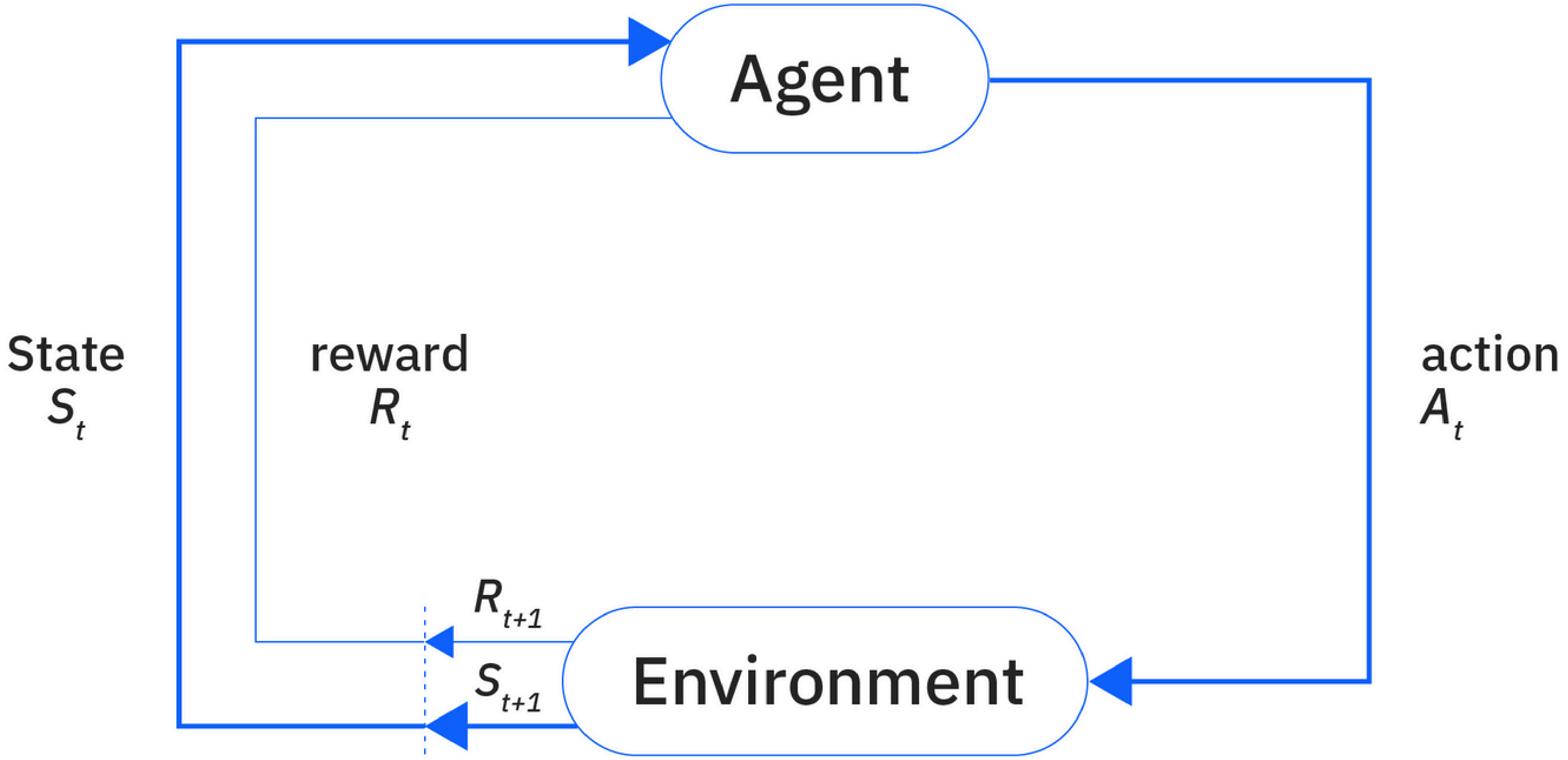
Unsupervised Learning

- Uses unlabelled data
- Goal: Organize or transform data based on similarities or statistical properties
- Uses: Customer segmentation, dimensionality reduction
- Algorithms: K-Means clustering, PCA


Unlabelled Data
Machine
Results

Reinforcement learning

- Model learns by interacting with environment
- Based on rewards / penalties mechanism
- Goal: Learn a policy that maximizes reward
- Uses: Gameplaying, robotics
- Algorithms: Q-Learning, PPO


Agent
State
S_t
reward
R_t
action
A_t
R_{t+1}
S_{t+1}
Environment

Data

- Raw facts fed into models
- Numbers, text, images, sounds,etc.
- Collected from the real world
- The essence of machine learning
- Better data > better algorithm

Basic Dataset Terminolgies

- Feature (Column): An individual variable or attribute
- Observation (Row): A single data entry or record
- Target (Label): The value we want to predict (in supervised learning)
- Categorical Data: Data with discrete categories (e.g., colors, cities)
- Numerical Data: Data with numeric values (e.g., age, price)
-

What makes Data “Good”?

- Complete and accurate (no missing or wrong values)
- Consistent (no duplicates or errors)
- Relevant(contains features that help solve the problem)
- Balanced classes (for fair learning)

Data Inspection

- First step after loading data
- Understanding data structure and quality
- Identifying issues like missing values, duplicates, outliers, and incorrect data types

Missing Values

- Data entries that are empty or null
- Incomplete information affects the accuracy and dependability of model
- isnull() function returnsTrue for NaN value.

	Height	Weight	Country	Place	Number of days	Some column
0	12.0	35.0	India	Bengaluru	1.0	NaN
1	NaN	36.0	US	New York	2.0	NaN
2	13.0	32.0	UK	London	NaN	NaN
3	15.0	NaN	France	Paris	4.0	NaN
4	16.0	39.0	US	California	5.0	12.0
5	NaN	NaN	NaN	Mumbai	NaN	NaN
6	NaN	NaN	NaN	NaN	6.0	NaN

Duplication

- Repeated data entries causing redundancy
- Increases the processing time and storage
- Can bias analysis and reduce model accuracy

Duplication

	A	B	C	D	E
1	Year	Sport	Athlete	Country	Medal
2	2012	Wrestling	Artur TAYMAZOV	UZB	Gold
3	2012	Wrestling	Davit MODZMANASHVILI	GEO	Silver
4	2012	Wrestling	Komeil GHASEMI	IRI	Bronze
5	2012	Volleyball	Martins PLAVINS	LAT	Bronze
6	2012	Volleyball	Julius BRINK	GER	Gold
7	2012	Volleyball	Alison CERUTTI	BRA	Silver
8	2012	Volleyball	Martins PLAVINS	LAT	Bronze
9	2012	Wrestling	Davit MODZMANASHVILI	GEO	Silver

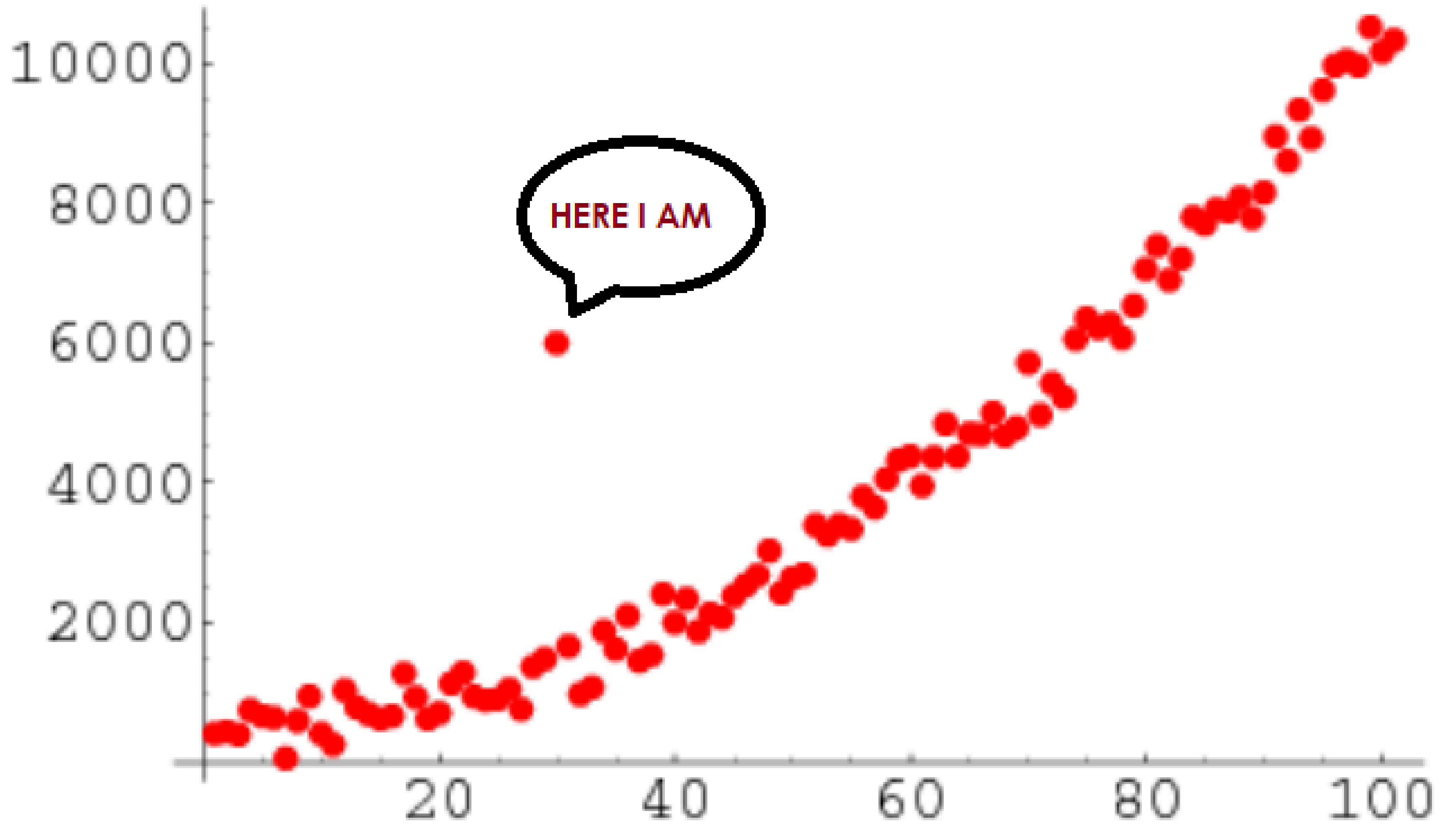
Data Type Issues & Inconsistent Values

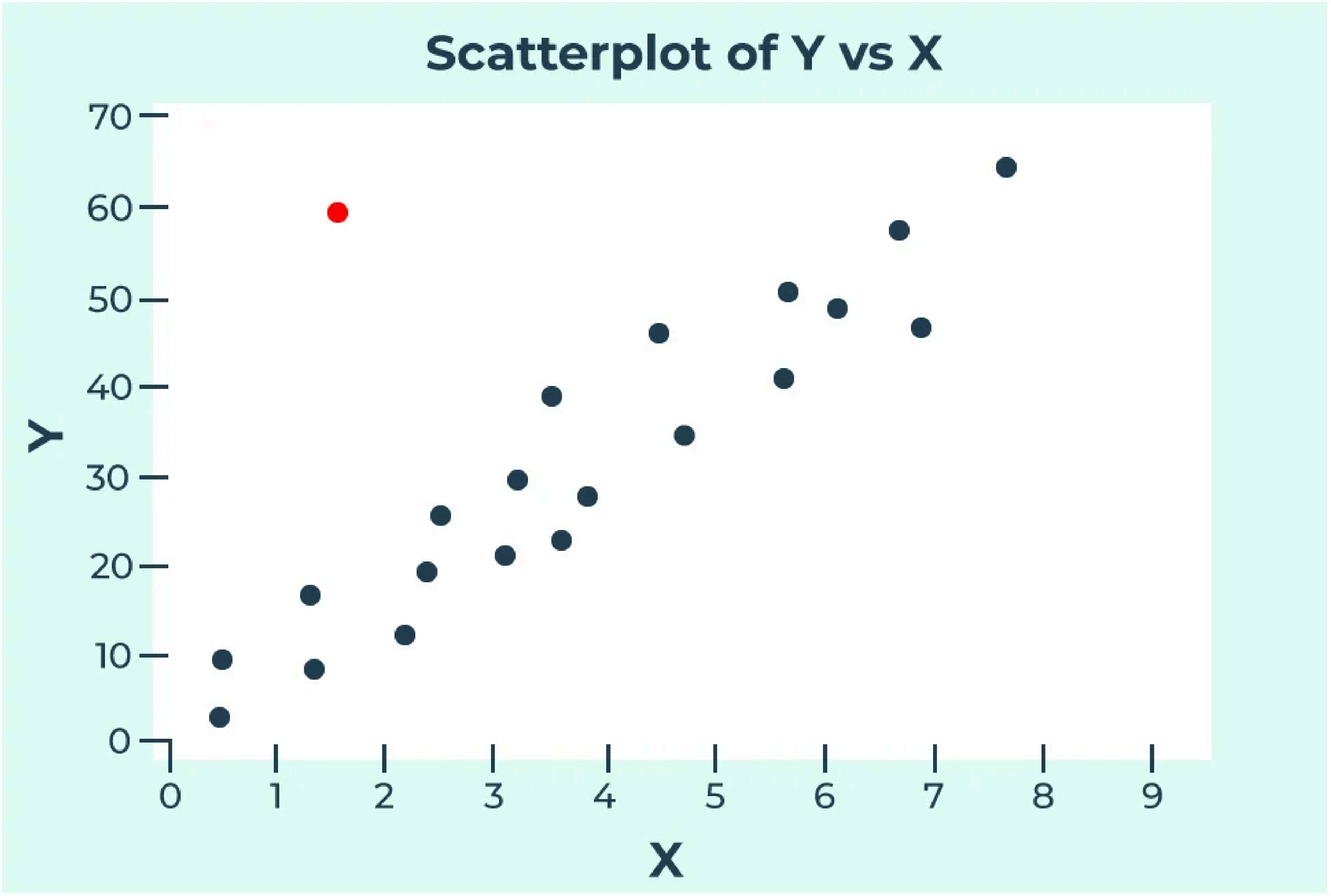
- Wrong data types (e.g., dates as text, numbers as strings) cause processing and analysis errors
- Same category recorded in different formats (e.g., “Male”, “M”, “male”) causes inconsistencies
- Lead to faulty analysis, unreliable results, and data management issues

Outliers

- Extreme values that differ form most of the other data points in the dataset.
- Often caused by errors, unusual behavior, or rare events
- Can skew summary statistics and negatively affect model performance
- Four ways to identify outliers
    - Sorting method
    - Data visualization method
    - Statistical tests (z scores)
    - Interquartile range method

Outliers


10000
8000
6000
4000
2000
20
40
60
80
100
HERE I AM

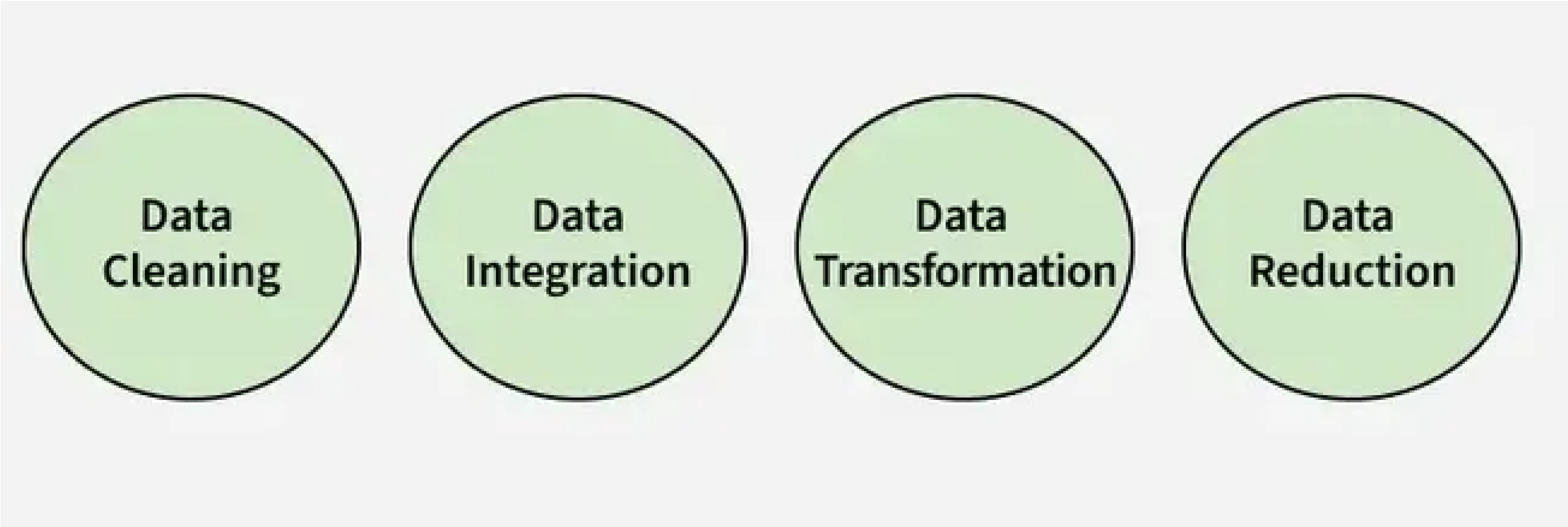

Scatterplot of Y vs X
70
60
50
40
30
20
10
0
Y
0
1
2
3
4
5
6
7
8
9
X

Exploratory Data Analysis

- Analyzing and visualizing data to uncover patterns and identify relations

- The types of analysis are:
    - Univariate Analysis
    - Bivariate Analysis
    - Multivariate Analysis

Data Preprocessing



Data Cleaning

Why Clean Data?

- Real data is often messy and incomplete
- Dirty data causes errors and bad predictions
- Efficient data saves time and computing power
- Clean data > accurate, faster, and efficient models

Handling Duplicate Values

- Identify duplicates using df.duplicated()
- Remove them using df.drop_duplicates()
- Keep the first or last occurrence if needed

Handling Missing Values

- Drop or fill missing entries using dropna() or fillna()
- Common fill methods: mean, median, mode, forward/backward fill
- Drop when data is unimportant or few rows are affected
- Impute when data is important and a logical value can be estimated

Fixing Data Types & Inconsistencies

- Convert types using astype() or to_datetime() (for dates)
- Detect invalid numbers with .describe() or checks
- Use domain knowledge to fix, cap, or remove these values
- Find and standardize inconsistent categories (value_counts(), .str.lower().replace())

Handling Outliers

- Detect outliers using .describe(), boxplots, Z-score, or IQR
- Assess if outliers are errors or valid variations
- Remove or cap outliers based on context
- Use transformations (e.g., log) to reduce skewness

Data Integration

- Combining data from different sources into one dataset
- Ensures consistent and complete data for analysis

Data Transformation

- Converts raw data into a suitable format for analysis and modeling
- Key Processes:
    - Feature Scaling– Adjust numeric features to a common range or distribution
    - Encoding – Convert categorical data into numeric form
    - Aggregation, Discretization – Summarize or group data

Feature Scaling

- Datasets often contain features with different value ranges

(e.g., Age: 0–100 vs. Income: 0–100,000)

- Features with large values can dominate smaller ones
- Ensures all features are treated equally
- Helps models learn faster and more accurately

Types of Feature Scaling

- Normalisation(E.g. Min-Max Scaling, Robust Scaling)
- Standardisation(Z-score scaling)

Min-Max Scaling

- Scales data to a fixed range, usually[0, 1]

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Robust Scaling

- Uses the median and interquartile range (IQR)
- Effective for datasets with outliers

Standard Scaling

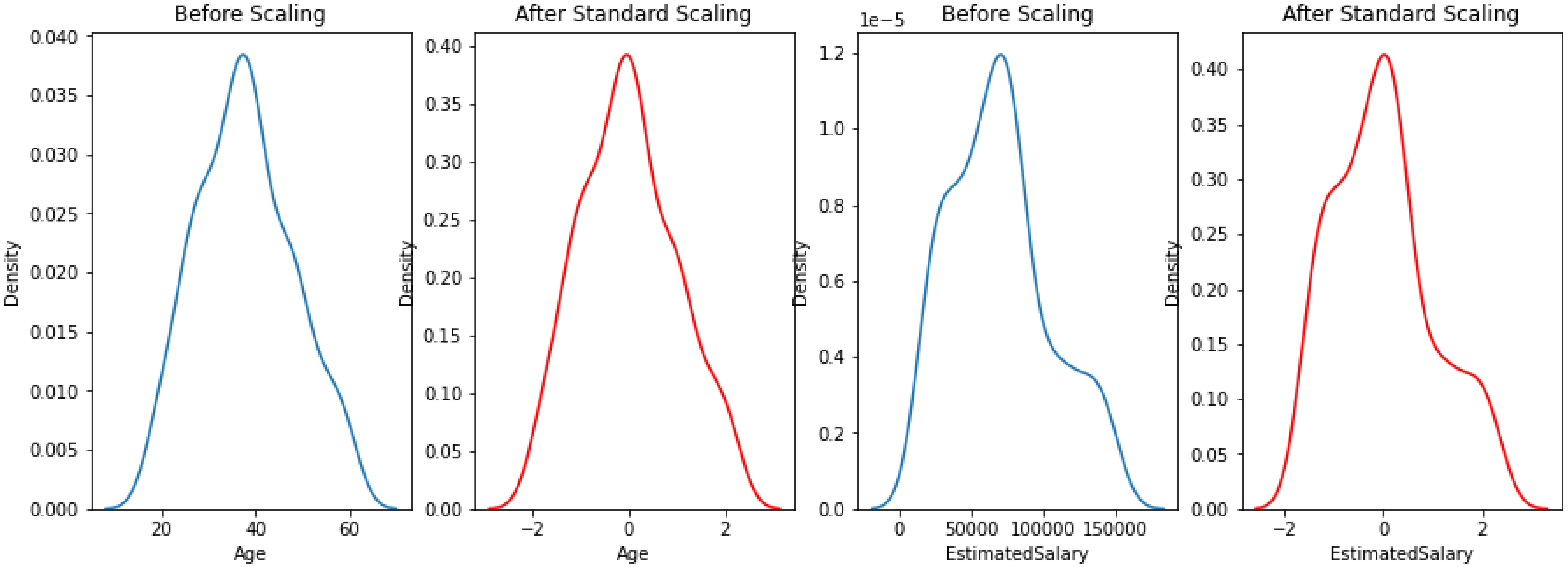
- Centers data around mean = 0, with standard deviation = 1
- Works well with algorithms that assume data is normally distributed

$$z = \frac{x - \mu}{\sigma}$$

$\mu =$ Mean

$\sigma =$ Standard Deviation

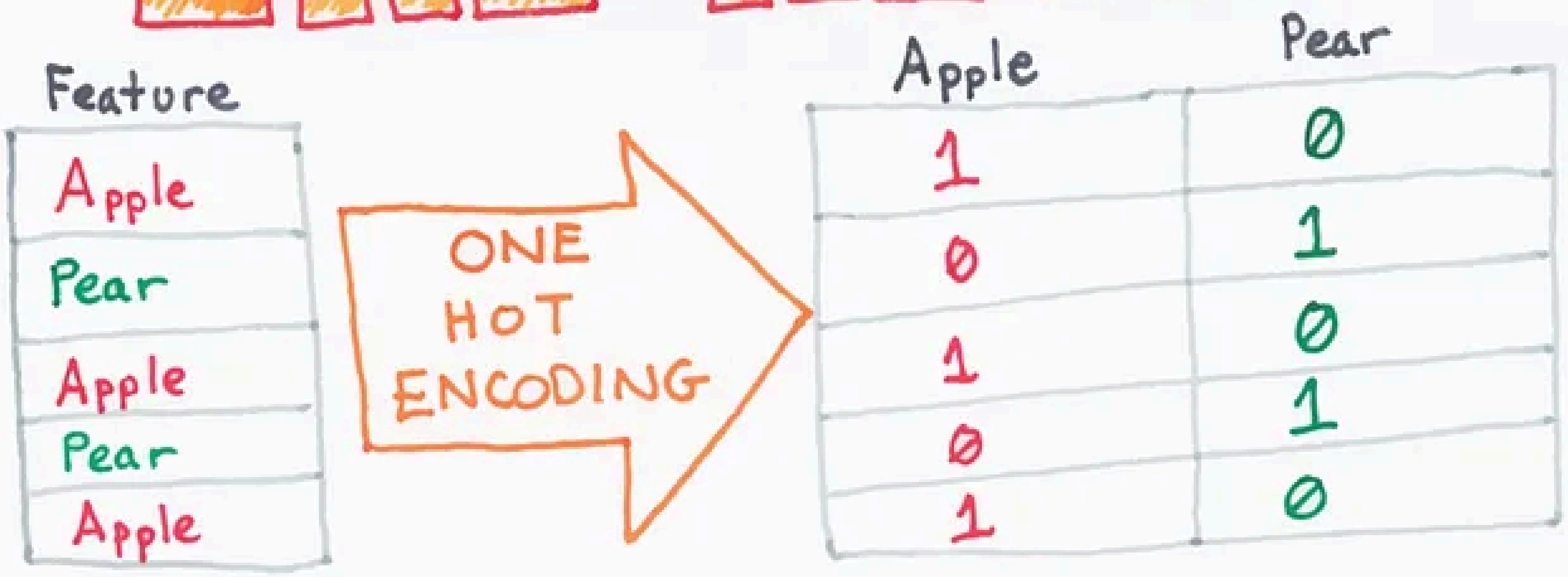
Standard Scaling



Categorical Data Encoding

- Converting categorical columns into numerical representation
- Types:
  - Ordinal Encoding
  - Nominal Encoding(E.g. One-hot encoding)

ONE-HOT ENCODING



One-hot encoding allows us to turn nominal categorical data into features with numerical values, while not mathematically imply any ordinal relationship between the classes.

ChrisAlbon

Ordinal Encoding

Btech
Master's
High School
PHD

PHD	4
Master's	3
Btech	2
High School	1

Other CommonTransformations

- Binning/Discretization: Convert continuous values into categories (E.g., Age → Teen, Adult, Senior)
- Log Transformation: Reduce skewness in data with large ranges
- Aggregation: Summarize data (e.g., avg. sales per region)

Data Reduction

- Reduces data size while keeping important information
- Speeds up processing and lowers memory usage
- Common Techniques:
    - Sampling
    - Aggregation
    - Feature Selection
    - Dimensionality Reduction

Feature Engineering

- Creating new features from existing data (e.g., total sales = price × quantity)
- Transform features (scaling, binning, log transforms)
- Select most relevant features to reduce noise
- Extract features from dates, text, or categories

Pandas

- Python library for data analysis and manipulation
- Key features:
  - Data cleaning, preprocessing
  - Filtering, grouping, reshaping datasets
  - Supports various formats (CSV, JSON,XLSX)
  - Compatible with other libraries like matplotlib, scikit-learn

code section

Data Visualization

- A graphical representation of information

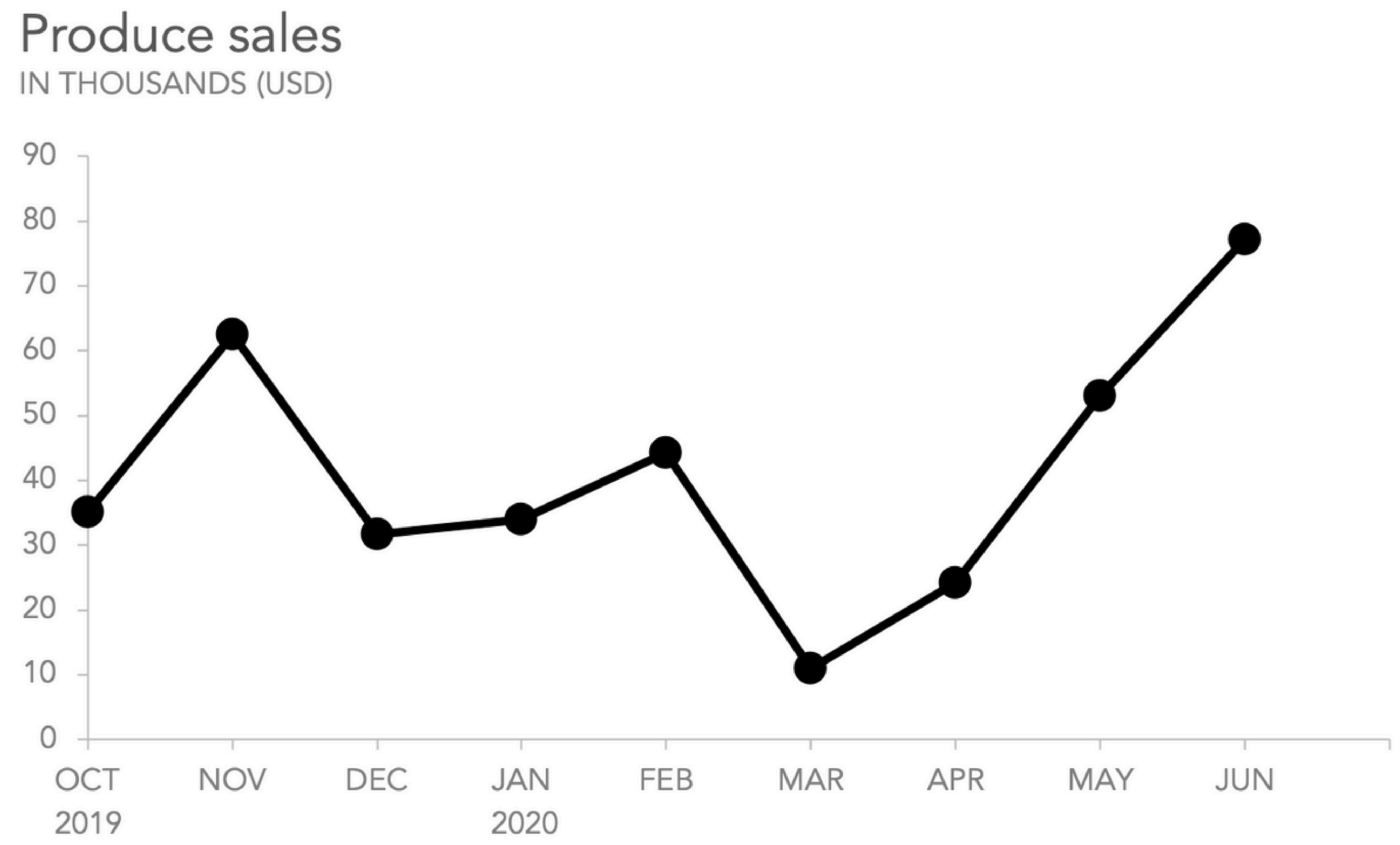
Why?

- Reveal hidden patterns
- Enhanced communication
- Faster insights
- Model evaluation

Common types of graphs

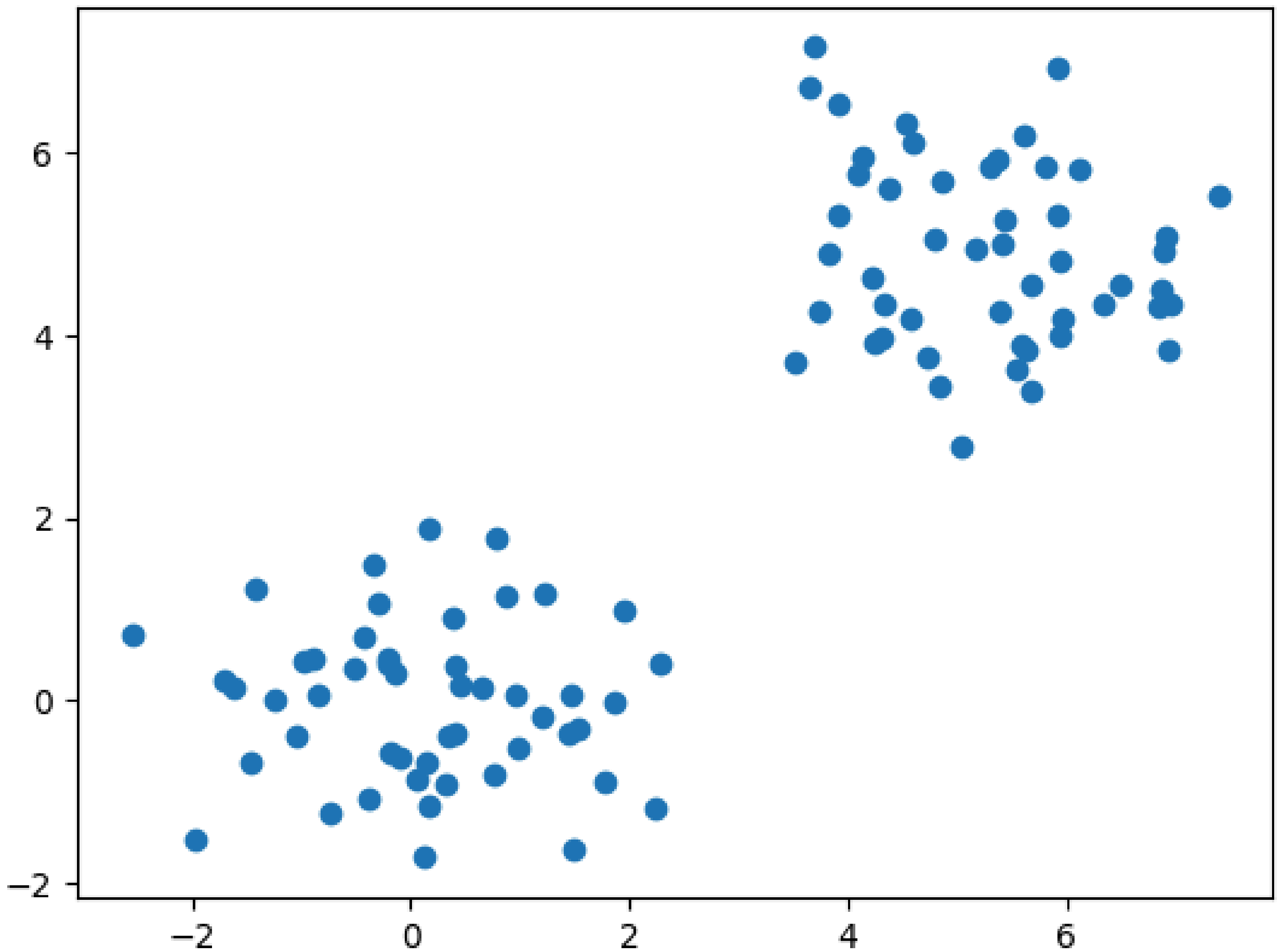
1. **Line plot**

- A basic plot that connects data points with a continuous line.
- Commonly used to visualize trends over time (e.g., model loss during training)

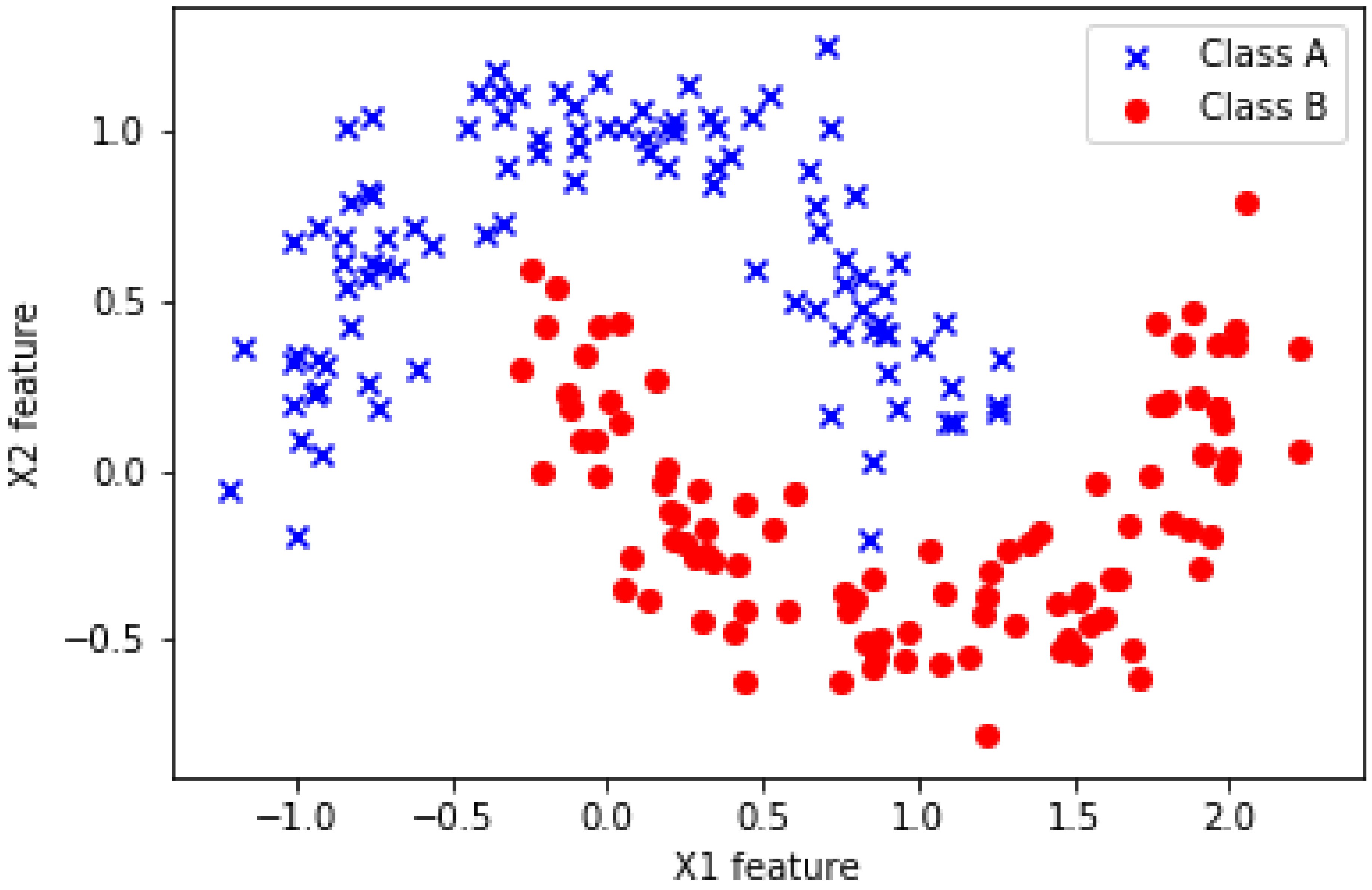

Produce sales
IN THOUSANDS (USD)
90
80
70
60
50
40
30
20
10
0
OCT
2019
NOV
DEC
JAN
2020
FEB
MAR
APR
MAY
JUN

2. Scatter plot

- Plots individual data points on an X-Y plane.
- Commonly used to visualize relationship between any two variables or visualizing clusters/groups



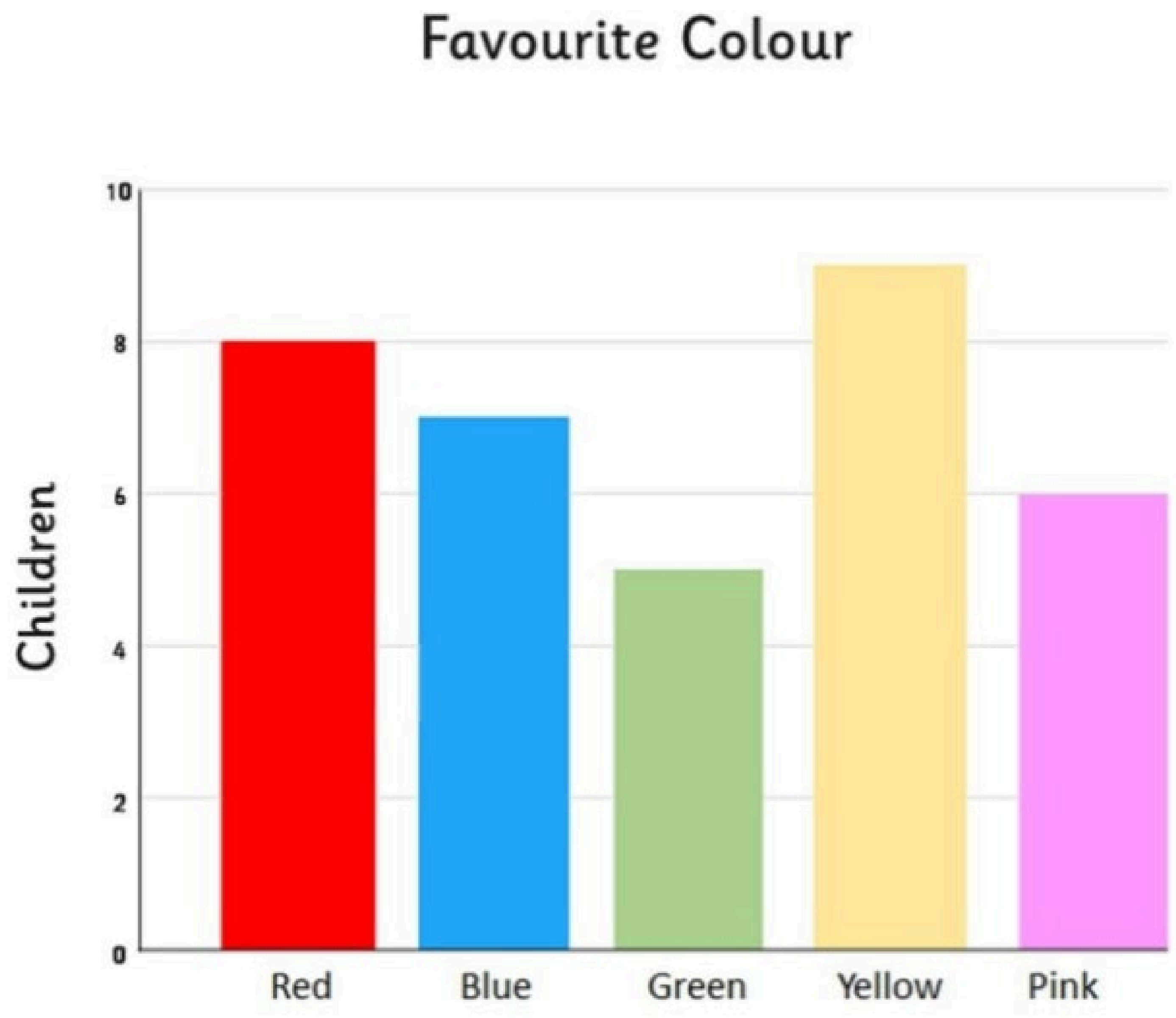
−2
0
2
4
6



Class A
Class B
X2 feature
X1 feature
1.0
0.5
0.0
−0.5
−1.0
1.5
2.0

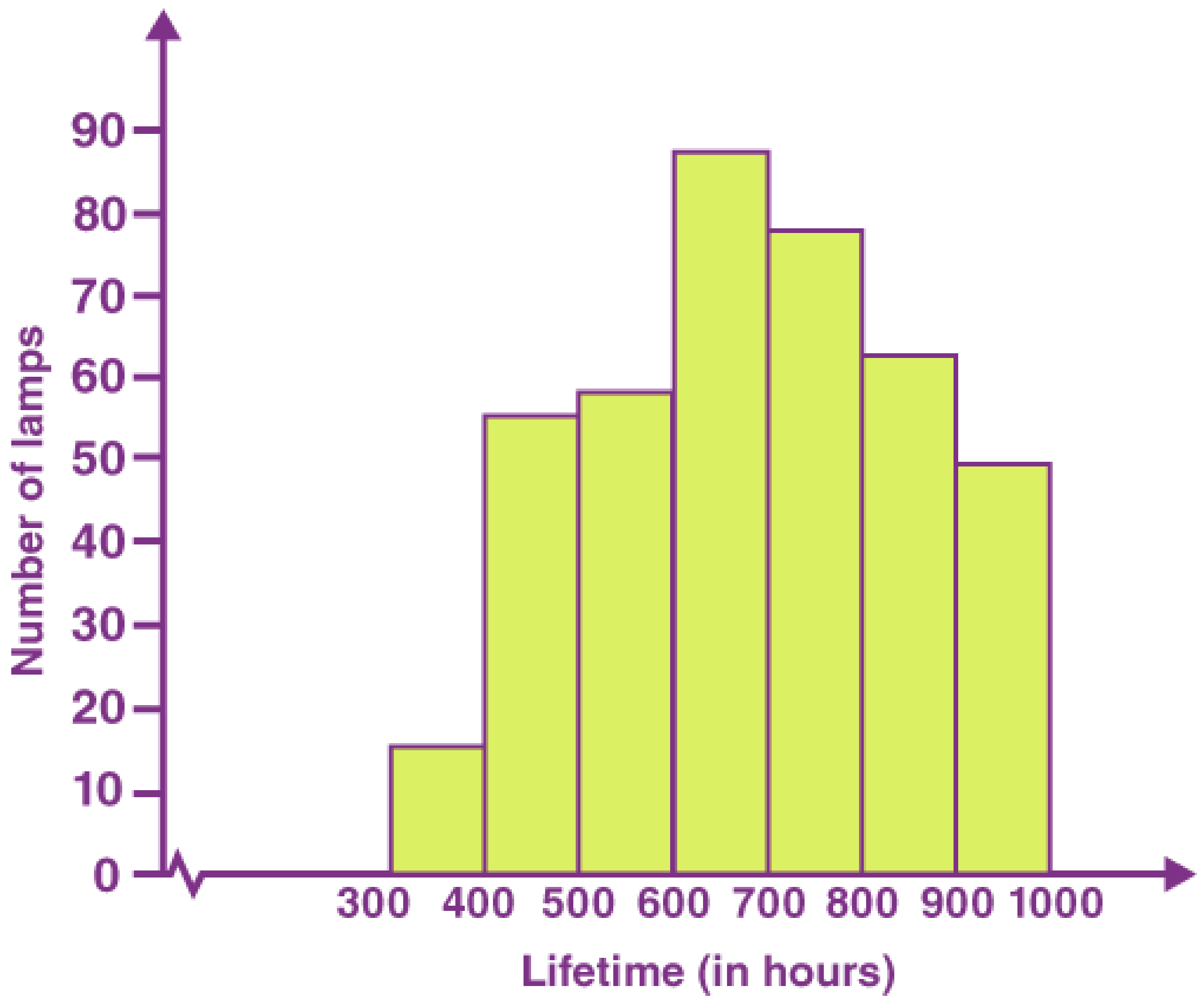
3. Bar Plot

- Displays categorical data with rectangular bars representing frequency or value.
- Useful for visualizing group statistics in classification problems


Favourite Colour
Children
10
8
6
4
2
0
Red
Blue
Green
Yellow
Pink

3. Histogram

- Groups continuous numerical data into bins and shows the frequency per bin.
- Useful for visualizing group statistics in classification problems


Number of lamps
90
80
70
60
50
40
30
20
10
0
300
400
500
600
700
800
900
1000
Lifetime (in hours)

What is Matplotlib?

- A powerful plotting library in Python
- Works well with NumPy, Pandas
- Produces static, interactive, and animated plots
- Inspired by MATLAB plotting

Key Features of Matplotlib

- Line plots
- Bar charts
- Histograms
- Pie charts
- Scatter plots

Simple Plot

```python
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]
plt.plot(x, y)
plt.show()
```

Plot Customization Basics

```
plt.title("Simple Line Graph")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.grid(True)
```

Bar Plot

```
categories = ['A', 'B', 'C']
values = [5, 7, 3]
plt.bar(categories, values)
plt.show()
```

Specific Use Case-Comparing categories or groups of data.

Advantege-Clearly shows differences between categories. Can display both positive and negative values.

Disadvantage-Becomes cluttered with too many categories.

Scatter Plot

```
x = [5, 7, 8, 7, 2, 17, 2]
y = [99, 86, 87, 88, 100, 86, 103]
plt.scatter(x, y)
plt.show()
```

Specific Use Case-Showing relationships or correlations between two continuous variables

Advantages-Effectively shows trends, clusters, outliers. Can visualize large datasets effectively.

Disadvantages-Interpretation may require statistical understanding.

Pie Chart

```
sizes = [20, 30, 50]
labels = ['Apples', 'Bananas', 'Cherries']
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.show()
```

Specific Use Case-Showing proportions or percentages of a whole.

Advantages-Visually intuitive for showing parts of a whole. Simple and familiar to most audiences.

Disadvantages-Hard to compare similar sized slices. Becomes confusing with too many categories.

Subplots

```
plt.subplot(1, 2, 1)
plt.plot([1, 2, 3], [1, 4, 9])

plt.subplot(1, 2, 2)
plt.plot([1, 2, 3], [1, 2, 3])
plt.show()
```