Generative AI Academy

Introductions

- Name
- Background / Education / Role(s)
- Technical skills
- Any experience with Data Science/Machine Learning/Deep learning/Gen AI
- Area of interest
  (Programming/Application Development, Data Engineering/Cloud, Algorithms/Maths/Stats or any other)

About me

Vijay Agrawal

- Over 30 years of experience in IT in senior positions with 14+ years in Silicon Valley, USA.
  Worked at several fortune 500 companies in software product development, design and consultancy, innovation leadership roles

- Expertise in GenerativeAI, Machine Learning, Data Science, Data Engineering, Automation.
  - Partner, Data & AI practice at Baker Tilly, India
  - VP Innovation & Business Enablement at Kroll
  - Sr Consultant, Cloud and Data Engineering at Red Hat, India

- Past 1.5 years focused on training and consulting in Data/AI, Gen AI
  - Delivered 1000+ hours of workshops on AI and Generative AI across the spectrum from leadership/strategy to execution and implementation of Gen AI solutions

https://linkedin.com/in/agrawalvijay

Program flow

- EDA
- Data Science, Machine Learning
- Machine Learning with Azure
- Deep Learning, NLP, Computer Vision
- Generative AI
- Building Gen AI Apps: Prompt Engineering, RAG, Fine Tuning
- Graph RAG, Agents
- Governance, Compliance, Risks
- Future of Gen AI

What is AI?

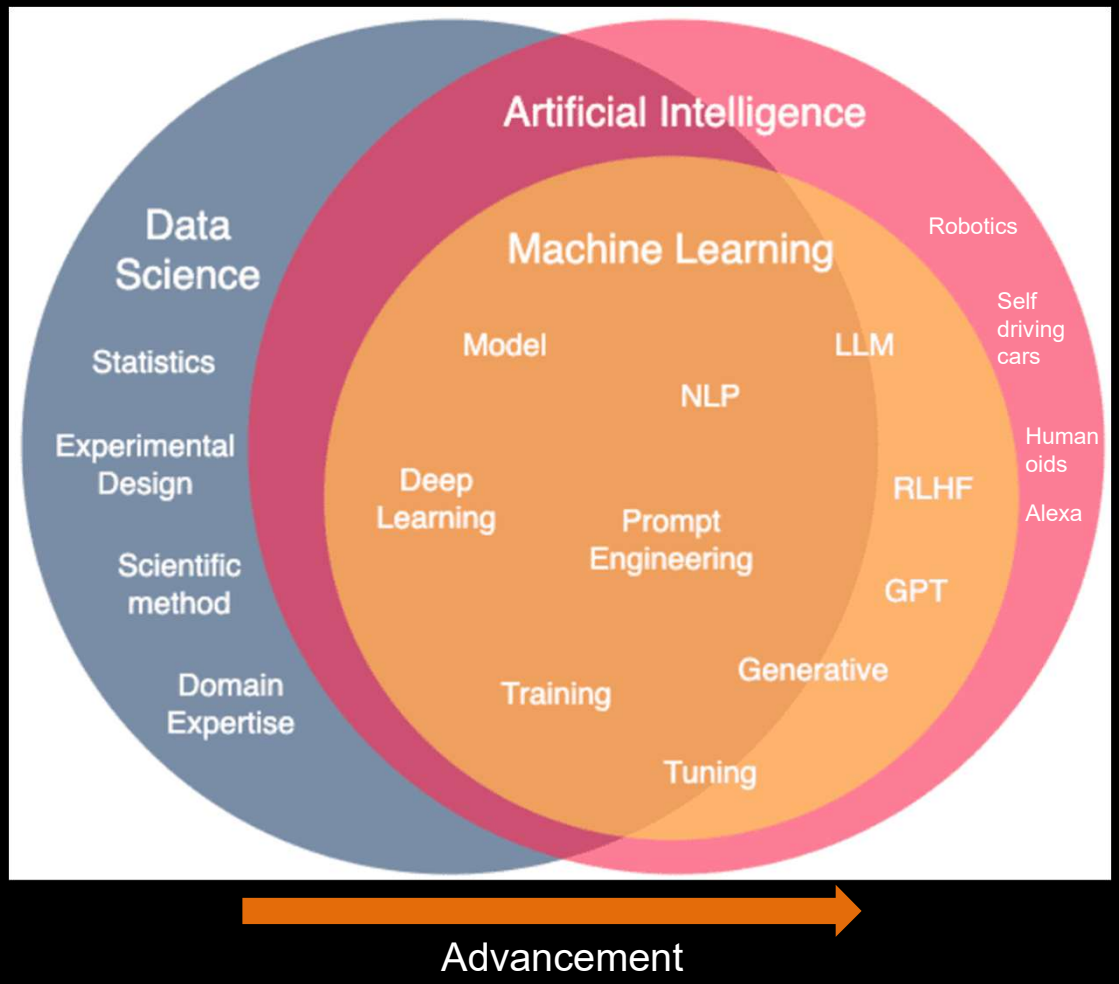
Artificial Intelligence (AI) is the field of computer science focused on creating machines or systems that can perform tasks that would typically require **human intelligence**.

These tasks often include:

- Reasoning
- Learning
- Problem-solving
- Understanding natural language
- Recognizing patterns
- Making decisions
- Generating new stuff based on it’s learning

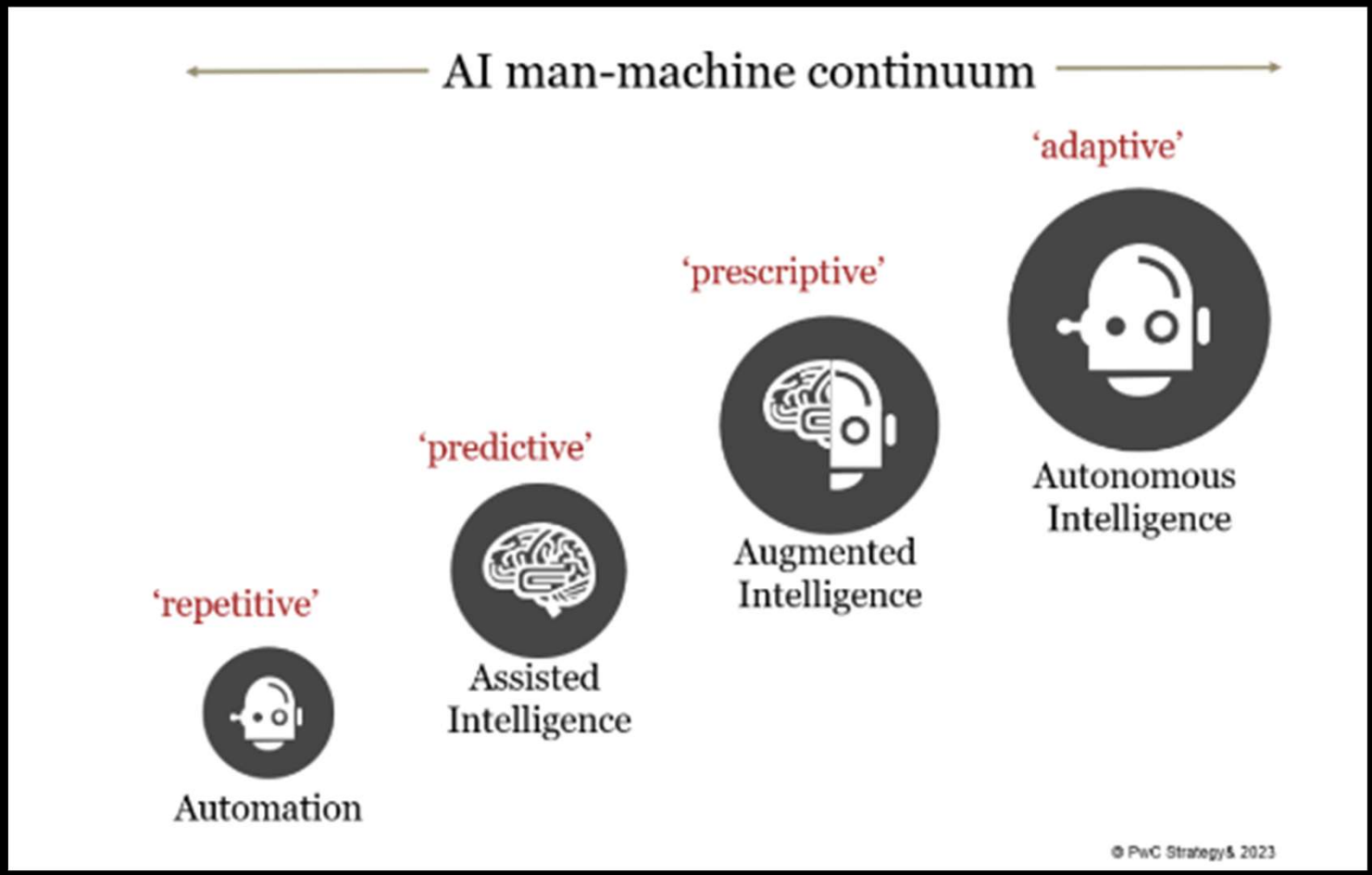
AI systems can adapt and improve over time, especially those utilizing advanced techniques like machine learning (ML) and deep learning.

Data science, Machine Learning, Deep Learning, AI

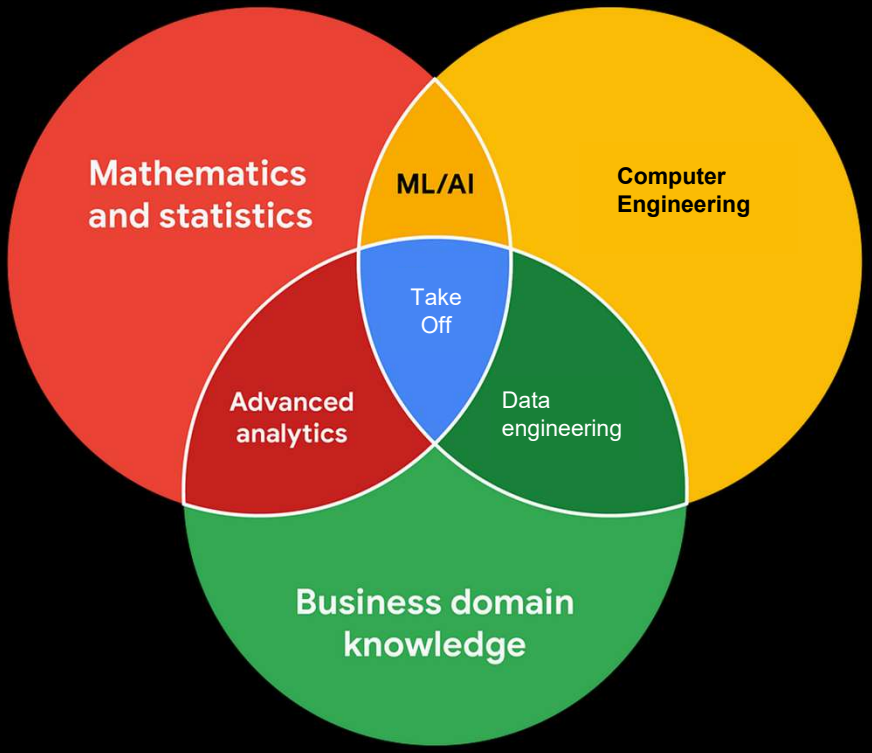


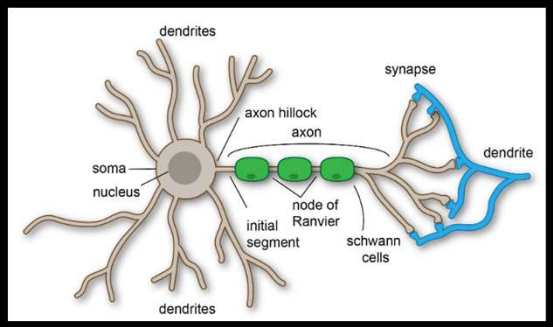
Data Science, Machine Learning, AI overlap and complement each other while still having their own distinct capabilities

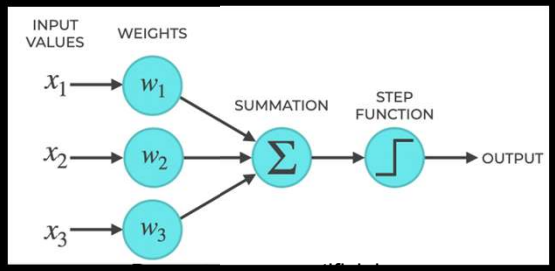
Evolution of AI

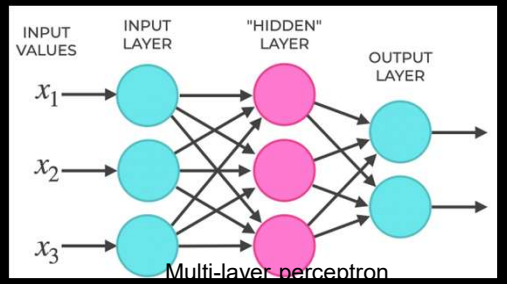


End goal: Mimic human brain



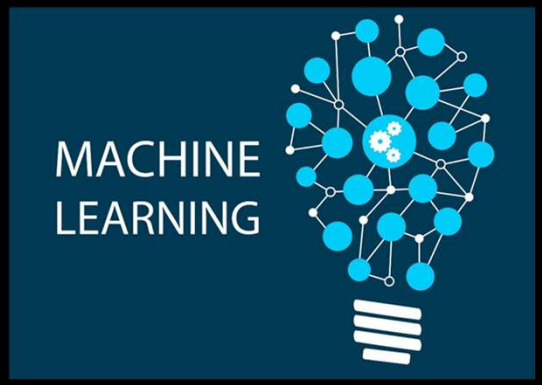




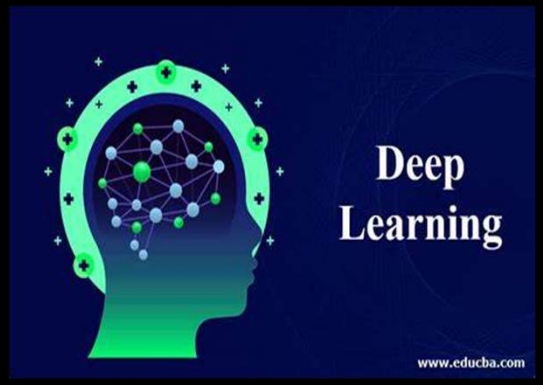


Multi-layer perceptron

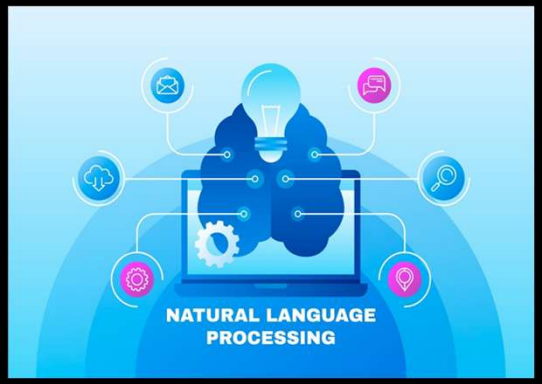
Key building blocks of modern AI



Machine Learning



Deep Learning



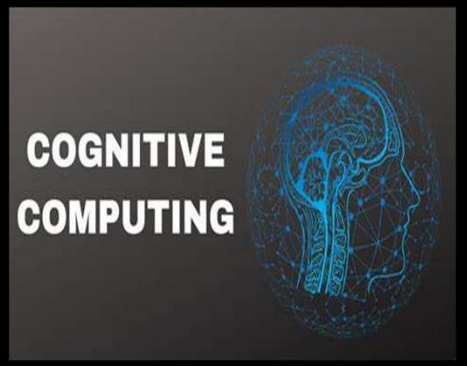
Natural Language Processing

Computer Vision

Cognitive computing

Cognitive computing represents a broader system that leverages both machine learning and deep learning to achieve human-like intelligence and decision-making capabilities.

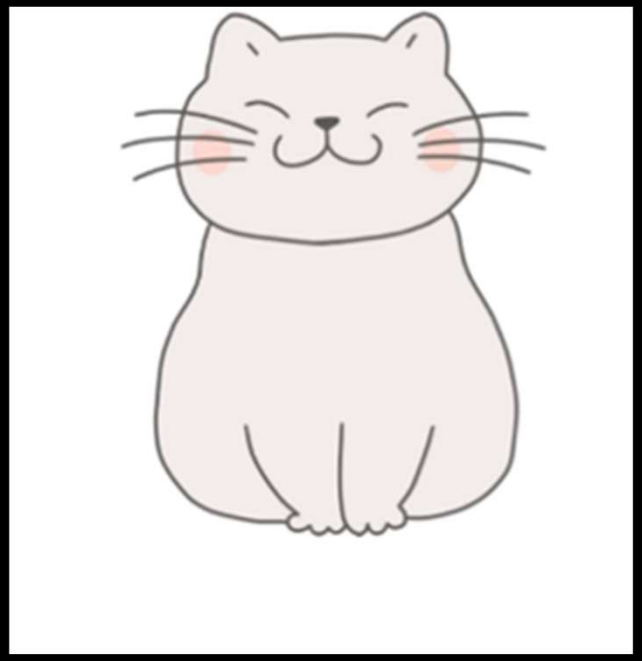
Feature	Machine Learning	Deep Learning	Cognitive Computing
Goal	Identify patterns, predict outcomes	Automate feature extraction, handle unstructured data	Simulate human thought, assist decision-making
Data	Structured	Structured and unstructured	Structured and unstructured
Scope	Predictive tasks	Complex recognition tasks	Decision support, reasoning
Technology	Algorithms (SVM, trees, etc.)	Neural networks	AI, ML, NLP, DL
Human Interaction	Limited	Limited	High



Cognitive Computing

How to make the machine predict the animal from its picture?

Is this a cat or a dog?

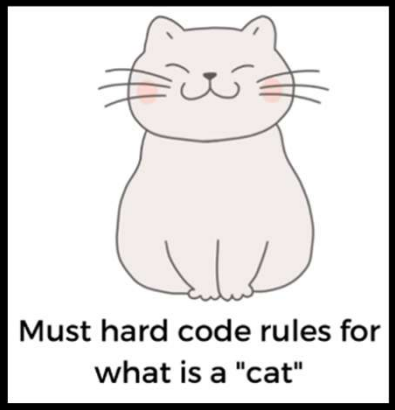


It’s all about learning to find patterns and relationships in the data

Traditional programming Vs Machine Learning

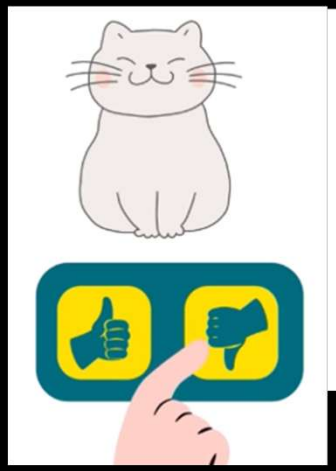
To Answer, "Is this a cat?"

Traditional Programming



Must hard code rules for what is a "cat"

With Machine Learning



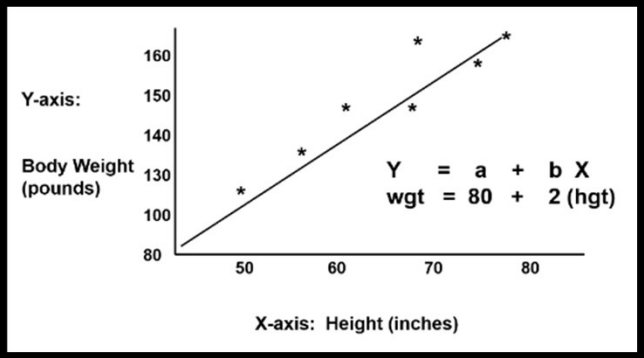
- Must give the network pictures of cats and dogs
- Makes a prediction based on training

Machine learning (ML) is defined as a discipline of artificial intelligence (AI) that provides machines the **ability to automatically learn** from data and past experiences to identify patterns and make predictions with minimal human intervention ,operating **without explicit programming**

End to end – predict weight of a person based on height

1. Start with training dataset

2. Determine the model to use
   Linear regression in this case



y = mx + c

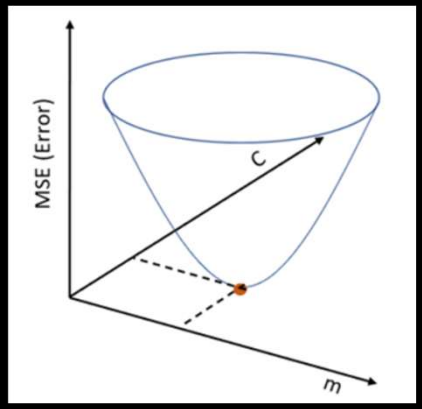
3. Measure error / accuracy

$$Cost\ Function(MSE) = \frac{1}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))^2$$

Cost function (or loss function) measures the error.
Our goal is to minimize the cost function to find the best fit line

4. Tune the model - Minimise error (improve accuracy)

Gradient Descent is a mathematical function that can help compute the model parameters (m, c) that will minimize the loss function in minimum number of iterations.



Relies on derivatives and calculus

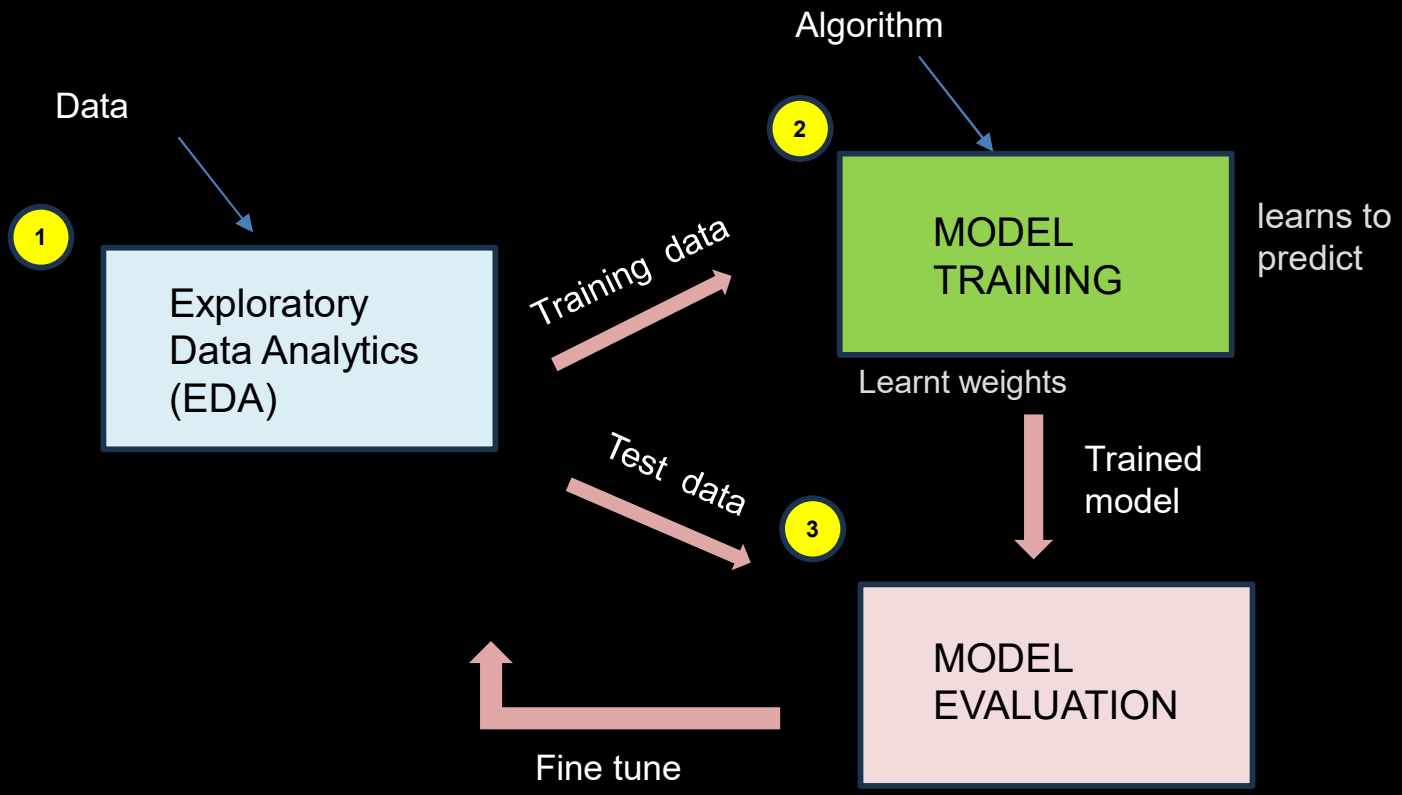
$$D_m = \frac{\partial(Cost\ Function)}{\partial m} = \frac{\partial}{\partial m}\left(\frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2\right)$$

$$D_c = \frac{\partial(Cost\ Function)}{\partial c} = \frac{\partial}{\partial c}\left(\frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2\right)$$

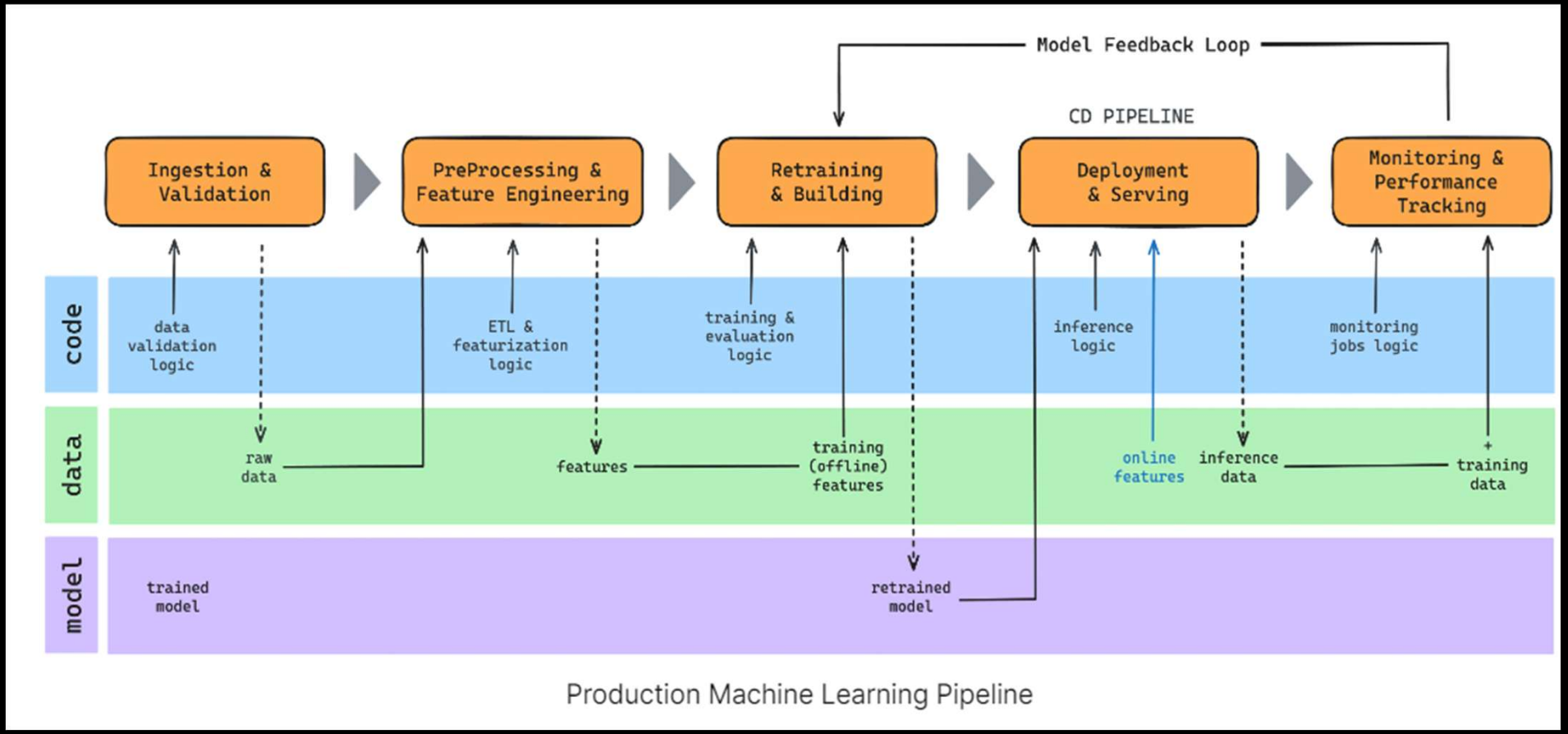
$$m = m - LD_m$$

$$c = c - LDc$$

Data Science part of the solution



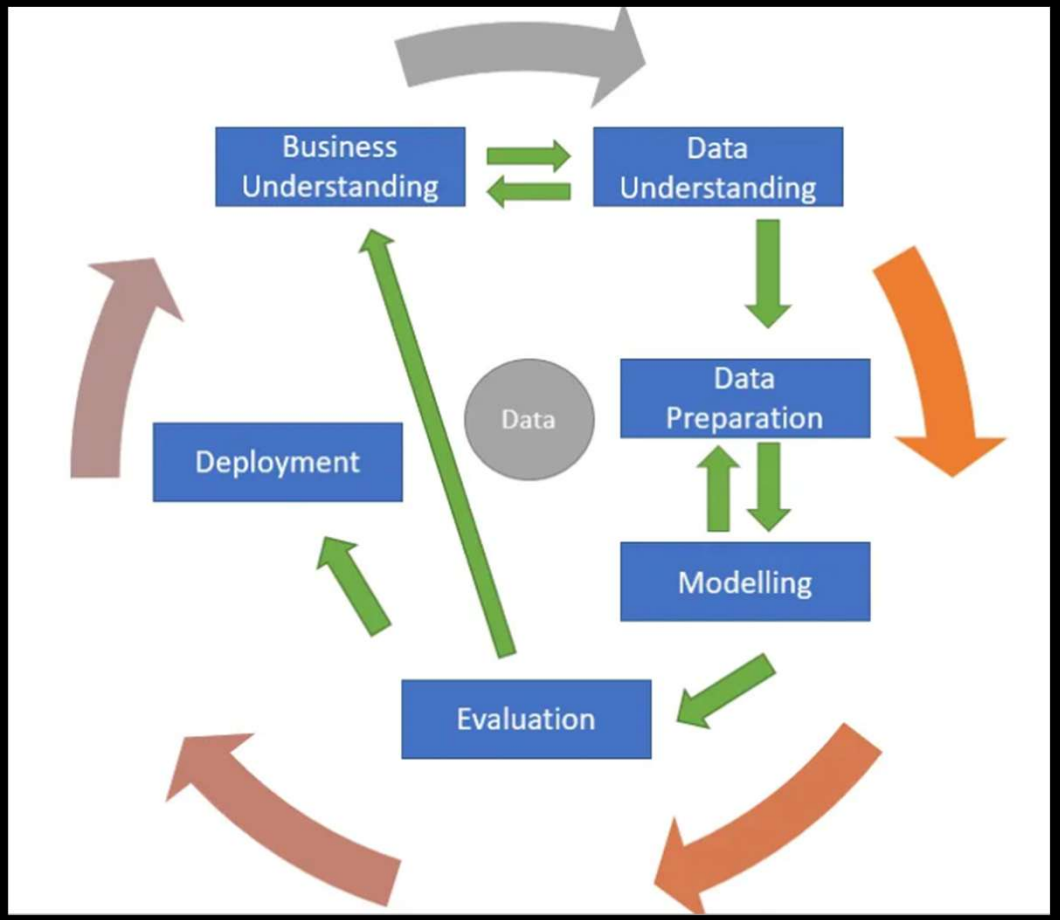
Machine learning pipeline



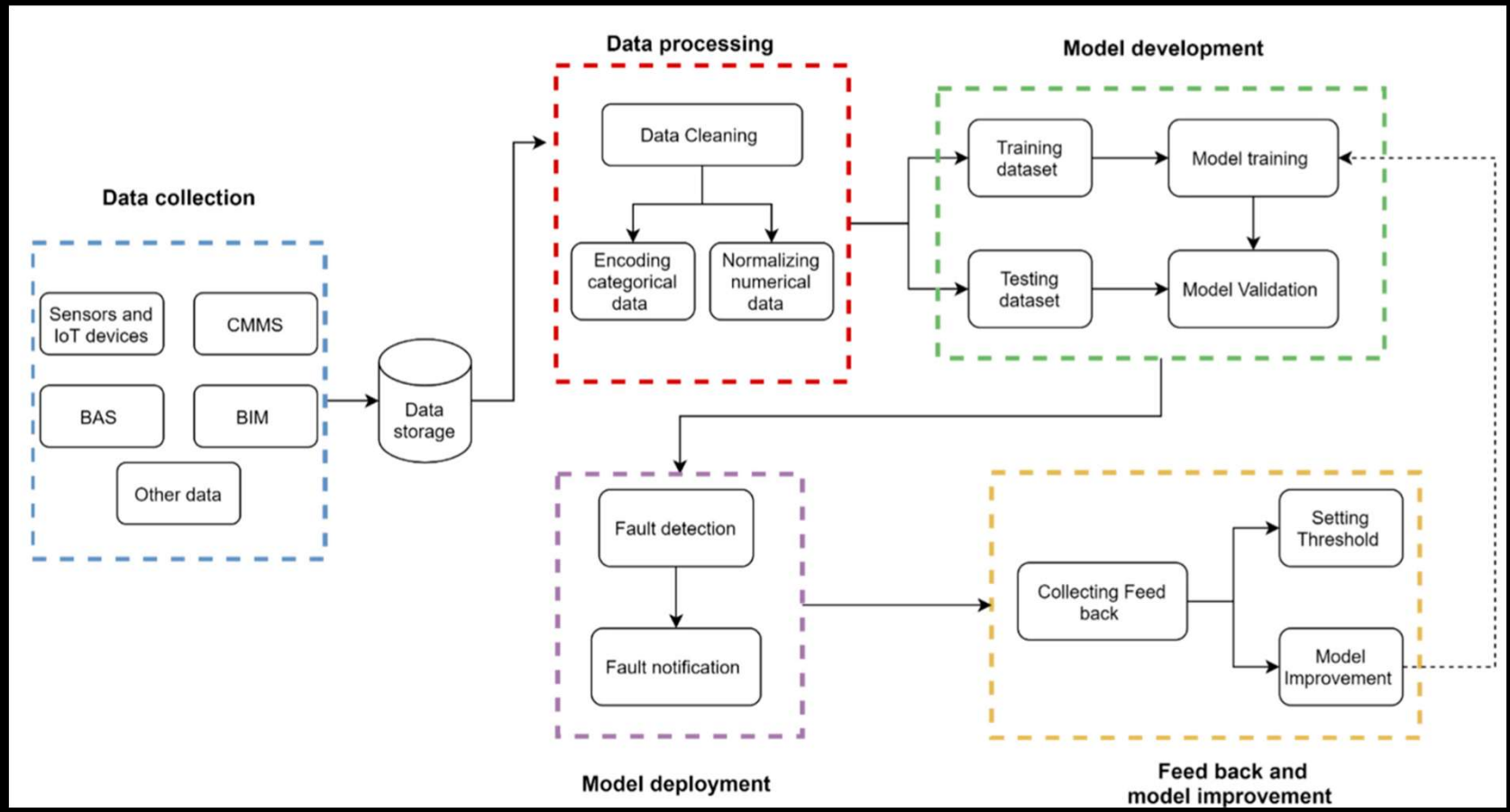
Production Machine Learning Pipeline

CRISP-DM Framework

Cross-Industry Standard Process for Data Mining



Detailed flow and steps to execute a data science project



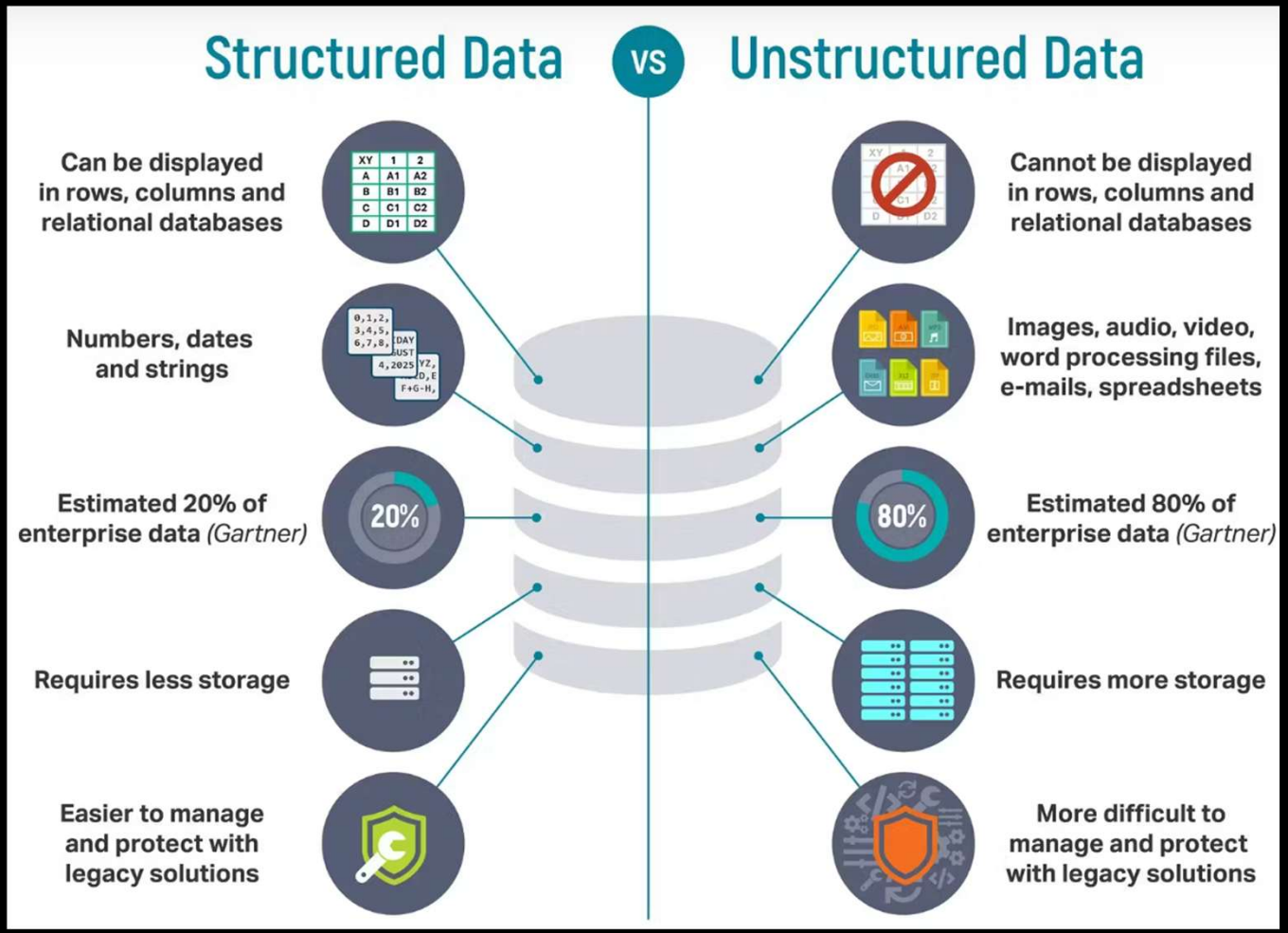
Our focus for this module:

Data Ingestion

Statistical Analysis

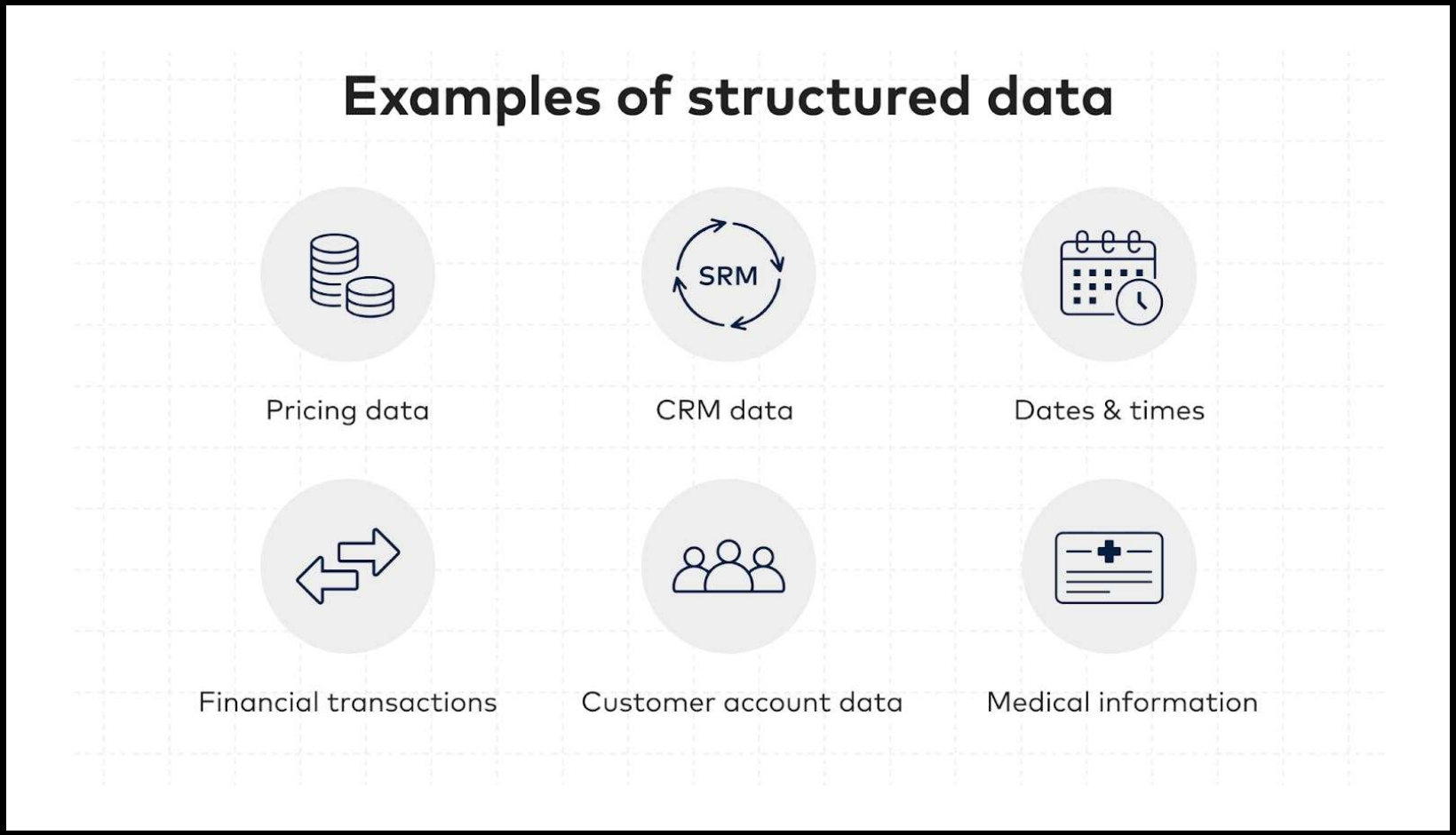
EDA

Types of data

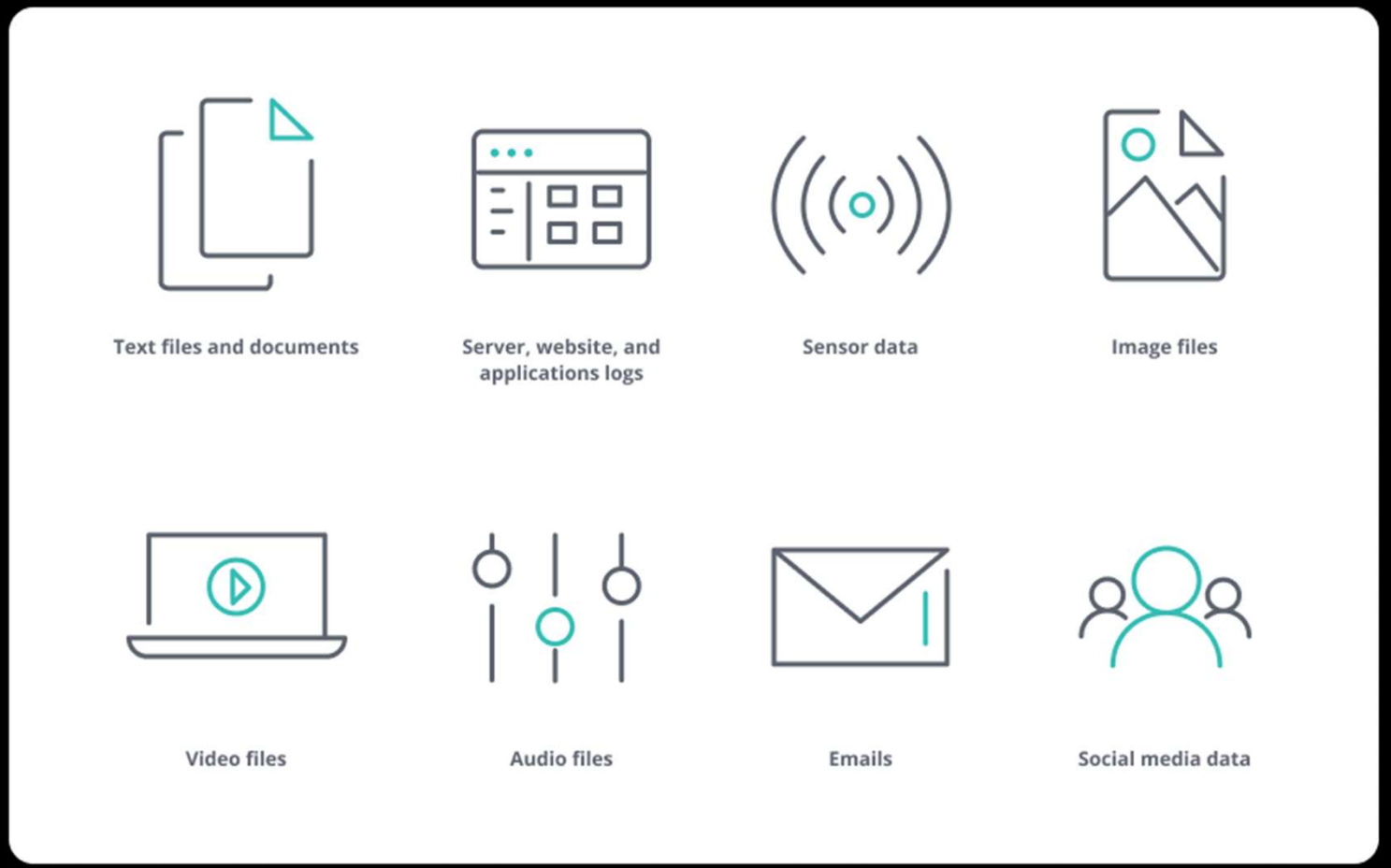


Data in JSON format will be classified as structured or unstructured?

Structured Data



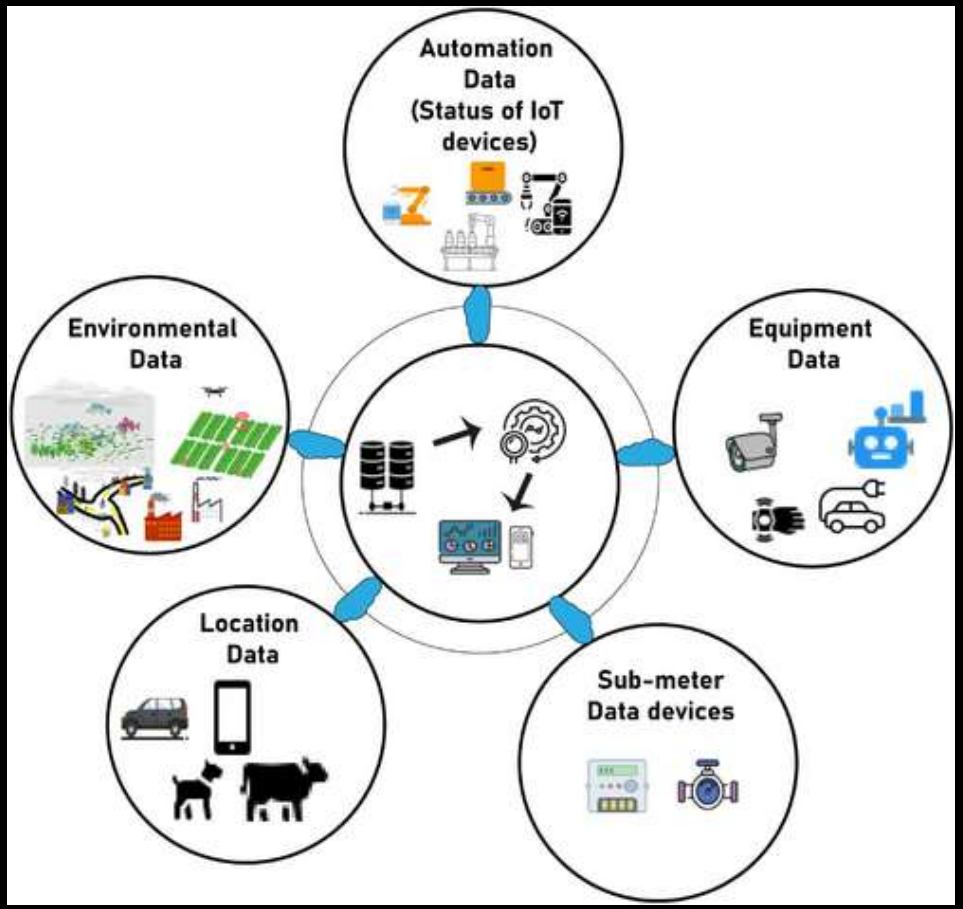
Examples of unstructured data



Examples of data sources and types of data

Data Source	Data Type	Structured/ Unstructured?	Key Intelligence that can be derived?
Logs from computer systems / Applications	Text	Unstructured / semi structured	Anomalies, error detection, performance degradation
Excel/CSV files, Databases	Text	Structured	Insights, forecasting, patterns and trends, recommendations
Audio data from calls, podcasts, conferences, meetings	Audio	Unstructured	Summarization, classification, prediction, insights (question answering)
Photo, video data from video conferences, calls, meetings	Images, Video	Unstructured	Summarization, classification, prediction, insights (question answering)
APIs: weather, stocks, business data	Text	Semi structured	Summarization, classification, prediction, insights (question answering)
Documents	Multi modal	Unstructured	Summarization, classification, prediction, insights (question answering)

Data Source example: iOT data



Data Handling in Python

How to represent student demographic information in python?

Option 1: List of dicts

```
students = [
    {'name': 'Vijay', 'age': 40},
    {'name': 'Sunita', 'age': 37},
    {'name': 'Megha', 'age': 33}
]
```

Option 2: Dicts with name as key

```
students = {
    'Vijay': {'age': 40},
    'Sunita': {'age': 37},
    'Megha': {'age': 33}
}
```

Option 3: Using pandas dataframe

df = pd.DataFrame(students)

- Which option is better?
- What are pros and cons of each?
- With many students, what techniques could be used to optimize search and retrieval?

Search and retrieval optimization techniques

Approach	Best For	Search Speed	Setup Complexity	Lookup Time Complexity (O notation)
Linear Search	Small datasets	Slow	Easy	O(n) (scans entire list)
Dictionary Indexing	Unique key lookups	Very Fast	Moderate	O(1) (average case)
Binary Search	Sorted datasets	Fast	Moderate	O(log n) (halves search space)
Pandas Indexing	Large structured datasets	Very Fast	High	O(1) (for indexed columns)
Elasticsearch	Large-scale text search	Very Fast	Very High	O(log n) (depends on index structure)

For <1000 records, a dictionary or Pandas DataFrame is enough.

For millions of records, consider Elasticsearch or SQL indexing.

Python refresher

```
students = [
  {'name': ‘Vijay', 'age': 40},
  {'name': ‘Sunita', 'age': 37},
  {'name': ‘Megha', 'age': 33}
]
```

```
sorted(students, key=lambda item: item['age'], reverse=True)
```

```
import pandas as pd

df = pd.DataFrame(students)

df.head()
```

Python – guess the output

```
text = "Python is awesome!“
result = text.replace("is", "was").split()
print(result[1])


data = ["a", "b", "c"]
result = [x.upper() for x in data if "b" not in x]
print(result)

text = "mississippi“
unique = set(text)
print(len(unique))
```

numpy refresher

```
# Importing Numpy package
import numpy as np

# Creating a 3-D numpy array using np.array()
org_array = np.array(
        [[23, 46, 85],
         [43, 56, 99],
         [11, 34, 55]])

# Printing the Numpy array
(org_array)

# Printing the Numpy array
np.sort(org_array)
```

Data ingestion

We will learn following techniques:

- Data collection from a csv file
- Data collection via API
  - Weather data
  - Stock data
  - Google search data
  - Twitter data
- Data collection via web scraping

Walkthrough and demos of data collection

Data ingestion Hands-on

1. Execute the supplied data ingestion demo notebooks in Google colab

2. Write a program (Python notebook or in VS Code) to
   a) fetch twitter tweets for a given topic
   b) Create a dataframe of top 100 tweets
   c) Save the dataframe in
      i. Csv format
      ii. Parquet format

Data ingestion – which format to save?

CSV
- Easy to read

Parquet
- Columnar format, enabling efficient retrieval of specific columns (features), making it more suitable for ML
- Compresses data more efficiently than CSV

- Enforces strong datatypes

Database/Data Warehouse

Exercise: Let’s save our dataframe in parquet format

Using pandas

```
import pandas as pd

df = pd.read_csv("data.csv")
df.to_parquet("data.parquet", engine="pyarrow", compression="snappy")
```

Or using pyspark

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("CSVtoParquet").getOrCreate()
df = spark.read.csv("data.csv", header=True, inferSchema=True)
df.write.parquet("data.parquet", mode="overwrite")
```

Exercise: reading from parquet file and creating pandas dataframe

Using pandas

Or using pyspark

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("CSVtoParquet").getOrCreate()
df = spark.read.csv("data.csv", header=True, inferSchema=True)
df.write.parquet("data.parquet", mode="overwrite")
```

Data ingestion at scale

Imagine a typical enterprise use-case where lots of data is being collected.

For example, a investment bank makes 10000+ API calls at 9am to a stock market service to obtain information about stocks in a large number of portfolios

1. How can the job be triggered?
2. How can it be done at high speed and efficiency?
3. What format should the ingested data be saved in?

Data ingestion at scale: Azure Data Factory

- Low code/no code tool:
  Visual pipeline designer, control flows, event triggers, scheduling
- Data ingestion, ETL at scale
- Integrates with Databricks, Synapse
- For async http calls, Python asyncio, aiohttp are suitable
- Use PySpark Python library and PySpark dataframes for distributed processing

Data ingestion at scale: Azure Data Factory

Code-Free ETL as a service

Ingest

- Multi-cloud and on-premise hybrid copy data
- 100+ native connectors
- Serverless and auto-scale
- Use wizard for quick copy jobs

Control Flow

- Design code-free data pipelines
- Generate pipelines via SDK
- Utilize workflow constructs: loops, branches, conditional execution, variables, parameters, ...

Data Flow

- Code-free data transformations that execute in Spark
- Scale-out with Azure Integration Runtimes
- Generate data flows via SDK
- Designers for data engineers and data analysts

Schedule

- Build and maintain operational schedules for your data pipelines
- Wall clock, event-based, tumbling windows, chained

Monitor

- View active executions and pipeline history
- Detail activity and data flow executions
- Establish alerts and notifications

https://learn.microsoft.com/en-us/azure/data-factory/introduction

Key steps in EDA

Data Cleaning
Handling missing values, outliers, duplicates, and inconsistencies.

Descriptive Statistics
Data understanding, Hypothesis testing
Data Visualization:
Using plots like histograms, scatter plots, box plots, and correlation matrices to visualize relationships between variables.

Feature Engineering
Creating new features or transforming existing ones to make data optimized for ML

Why is EDA important?

- Reduces uncertainty and bias
- Validates Assumptions – Ensures data follows expected distributions before applying models.
- Aids Feature Selection – Identifies important variables that impact predictions.
- Prevents Garbage-In, Garbage-Out – Ensures clean, relevant data for accurate models.
- Avoids costly mistakes – patterns and relationships uncover important business intelligence. For example, A company may assume higher ad spending leads to higher revenue, but hypothesis testing might show otherwise.
- Improves Model Performance – Provides insights to engineer better features and transformations.
- Reduces Overfitting – Detects spurious correlations and unnecessary complexity in data.
- Supports Business Decision-Making – Provides data-driven insights before deploying ML models.

Cleaning the data

- Removing duplicates
- Remove irrelevant data
- Standardize capitalization
- Convert data type
- Handling outliers
- Fix errors
- Language Translation
- Handle missing values
- Handling null or n/a values

  Imputation: Replacing missing values with a specific value (mean, median, mode, etc.).
  Dropping: Removing rows or columns with null/N/A values.
  Default Values: Assigning a default value to handle missing data.

Cleaning the data

Demo and hands-on

STATISTICAL ANALYSIS OF DATA
a.k.a. DESCRIPTIVE ANALYTICS

Why is statistical analysis important?

- Validate assumptions, business and research hypothesis
- Draw inferences about data distributions, relationships, and patterns.
- Feature selection and importance
- Helps with data driven decisions rather than intuition
- Model explainability
- Bias detection: test if model's predictions are **unfairly skewed** toward a specific group.
  Example: A Chi-Square test can check if a loan approval model disproportionately rejects applicants from a certain demographic.
- Model comparison and evaluations
- Avoids costly mistakes

Example: A company wants to check if a new marketing strategy increases sales. Hypothesis testing determines if the increase is significant or just due to chance.

The type of hypothesis testing varies depending on whether the data is **structured** (tabular, numerical, categorical) or **unstructured** (text, images, audio, etc.).

Examples of statistical analysis

Field	Example
Finance	Checking if a **new investment strategy** outperforms the old one.
Healthcare	Determining if a **new drug** is more effective than an existing one.
Marketing	Testing if **email marketing** improves customer retention.
Manufacturing	Verifying if a **new machine reduces defects** in production.
Economics	Checking if **inflation affects stock market returns**.
E-commerce	Testing if **free shipping increases sales**.

Key concepts: Probability

3 coins tossed. what is probability of 2 being heads?

How many total possible outcomes?
8 (2^3)

Is it a permutation problem or a combination problem?

3C2 = 3*2/(2!)*(3-2)! = 3
Probability: 3/8

Permutation: Order matters (arrange 3 books out of 5 books in a shelf)

Combination: Order does not matter (choose 3 books from 5 books)

Key concepts: Variance

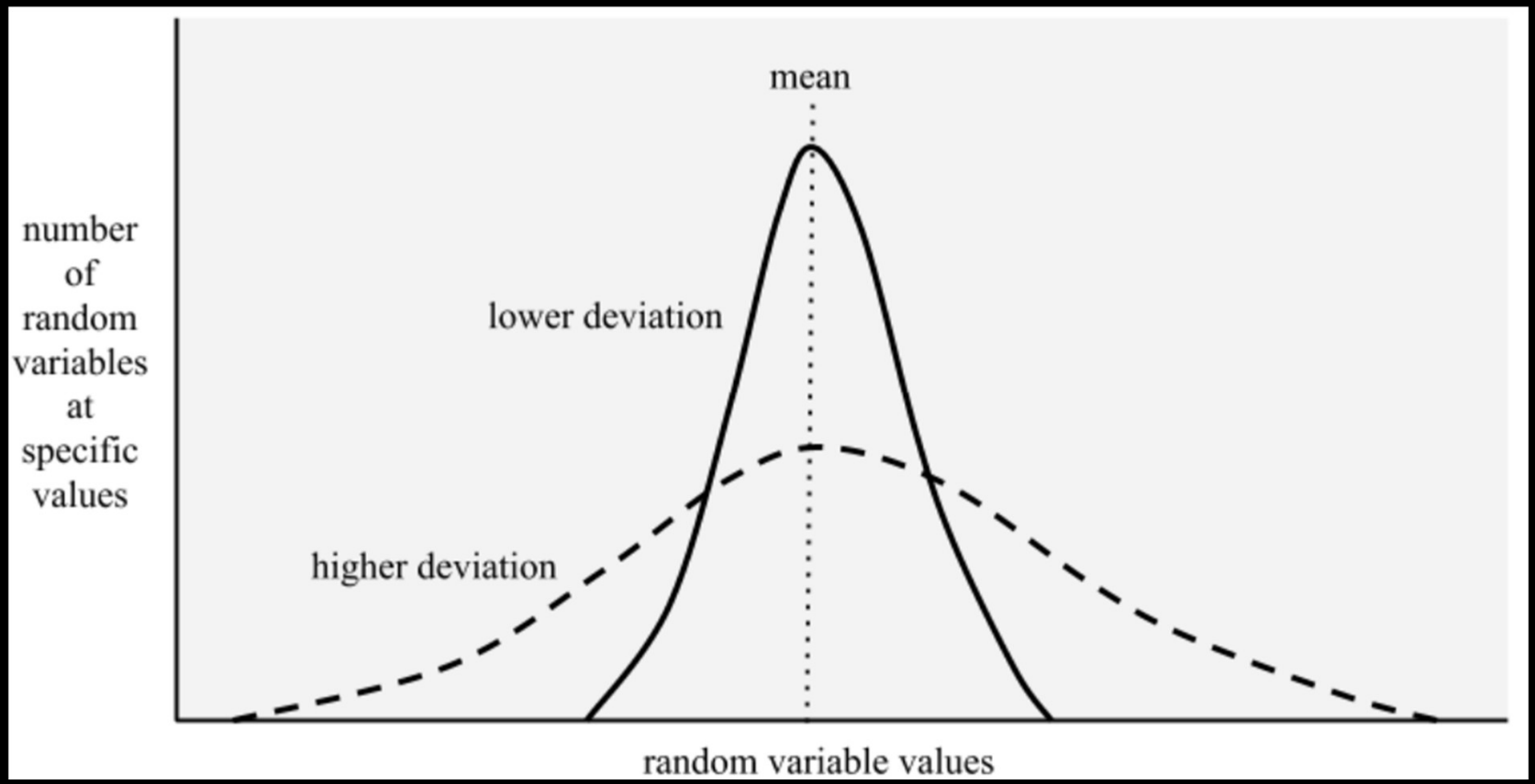
Variance *is a statistical measure that quantifies the spread or dispersion of a set of data points. It indicates how much the individual data points in a dataset differ from the mean (average) of the dataset*

Why is variance important in machine learning?

High Variance: **Noisy or overly complicated data**
Model captures noise instead of relationships. Can lead to overfitting – model performs well on training data but poorly on test data

Low Variance: **Lacks enough diversity**
Lacks variability, Leads to underfitting – model performs poorly in both train and test scenarios

Variance


mean
number of random variables at specific values
lower deviation
higher deviation
random variable values

Dealing with high variance

Data Cleaning & Preprocessing
Remove Outliers, Reduce Noise in Data:

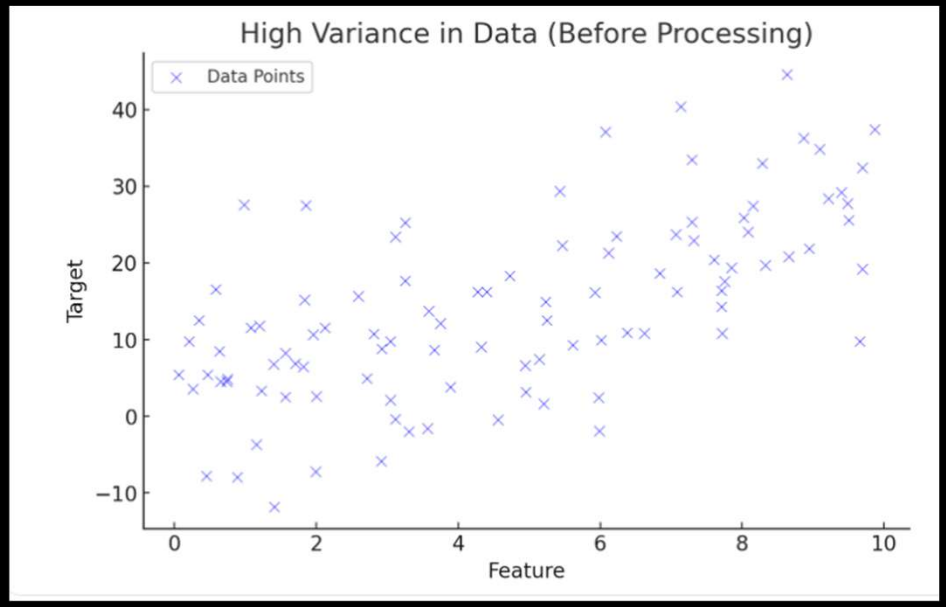
Feature Engineering & Selection
Remove redundant dimensions.
Feature Transformation, Normalization

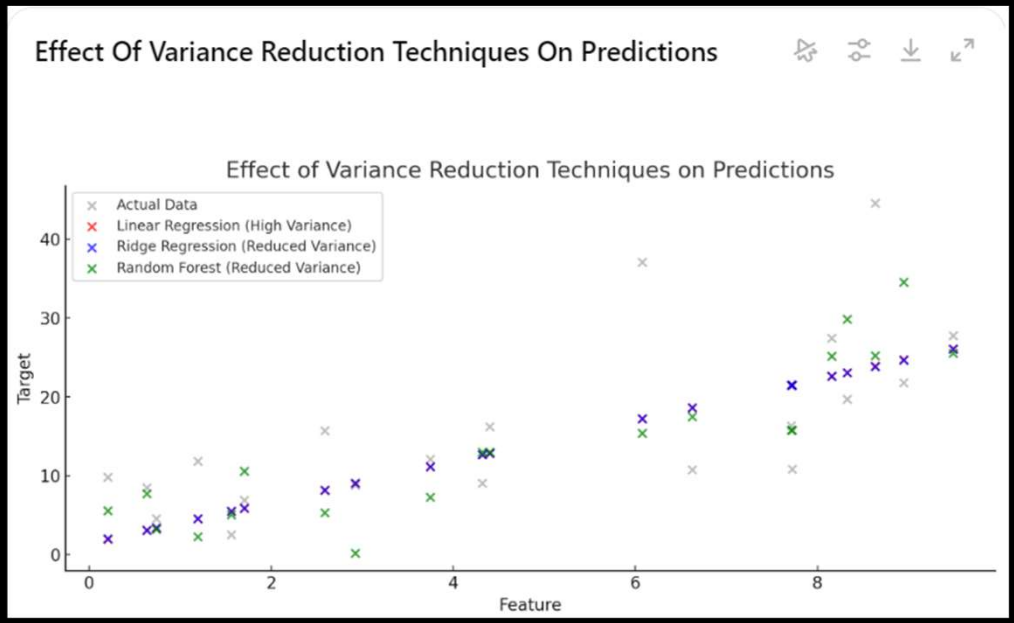
Increase Training Data

Augment Data (Synthetic Data Generation):
SMOTE (Synthetic Minority Over-sampling Technique) for imbalanced classification.
Data augmentation in computer vision (e.g., flipping, rotating images).

Model Regularization
L1 (Lasso), L2 (Ridge), Dropout
Ensemble Learning

Variance handling example





Dealing with low variance

Increase Data Diversity
Collect More Diverse Data:
Balance Class Distribution (oversampling, undersampling)

Improve Feature Representation
Create More Informative Features: polynomial, quadratic, domain specific: margins instead of raw revenues, combine 2 features (use age*income)

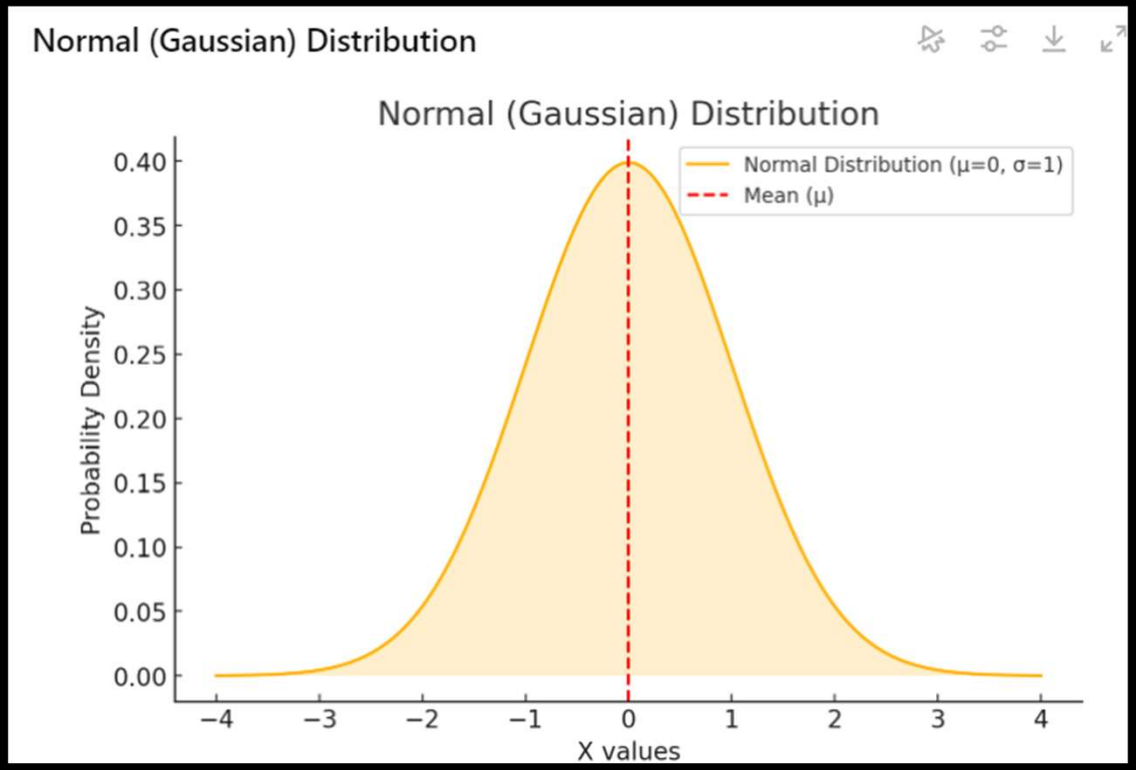
Use More Expressive Models
Switch to a More Complex Model:
If linear regression underfits, try decision trees or neural networks.
If a simple neural network underfits, add more layers or neurons.

Key concepts: Distribution

Distribution *describes how data is distributed in the population*

Normal Distribution

- Bell-shaped curve, symmetric around the mean
- 68% of values fall within $\pm 1\sigma$, 95% within $\pm 2\sigma$
- Examples in ML: heights of people, exam scores, stocks
- Many ML algorithms assume linearity (Linear and Logistic regression, for example)

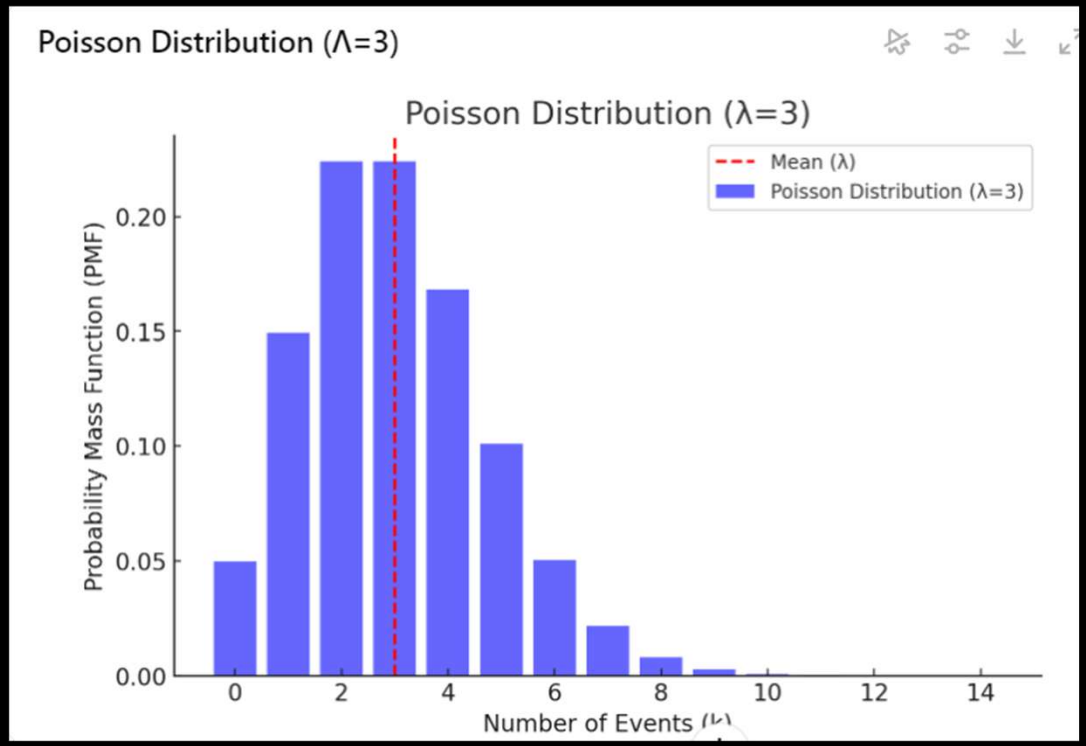


Key concepts: Distribution

Poisson Distribution *used for events occurring over time and space*

Normal Distribution

- Only non negative int values
- Examples in ML: number of website visitors, insurance claims, purchases
- Many ML algorithms assume linearity (Linear and Logistic regression, for example)

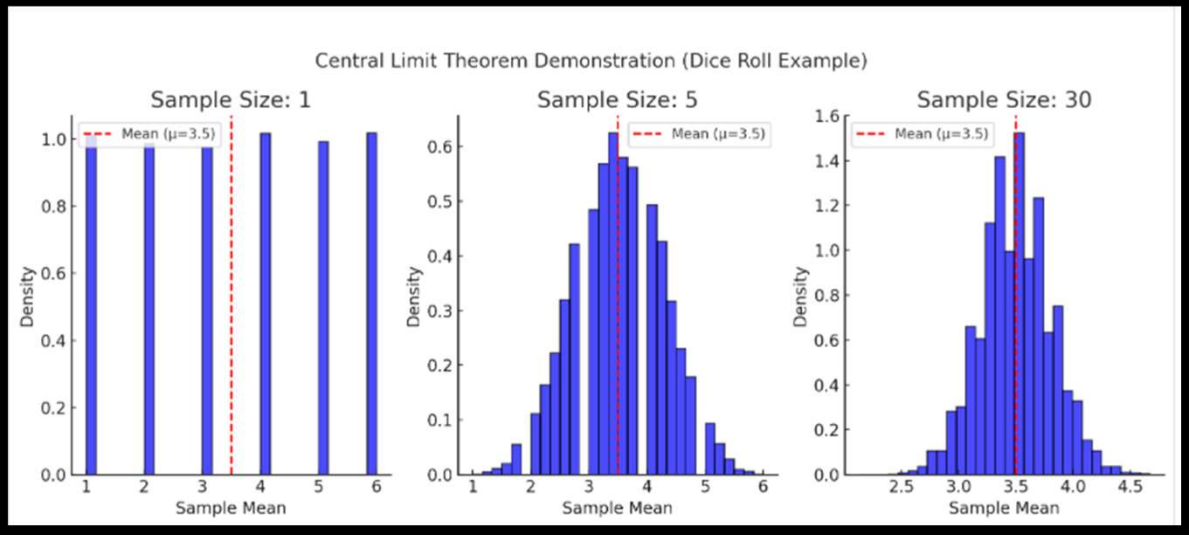


Key concepts: Central Limit Theorem

The Central Limit Theorem (CLT) states that

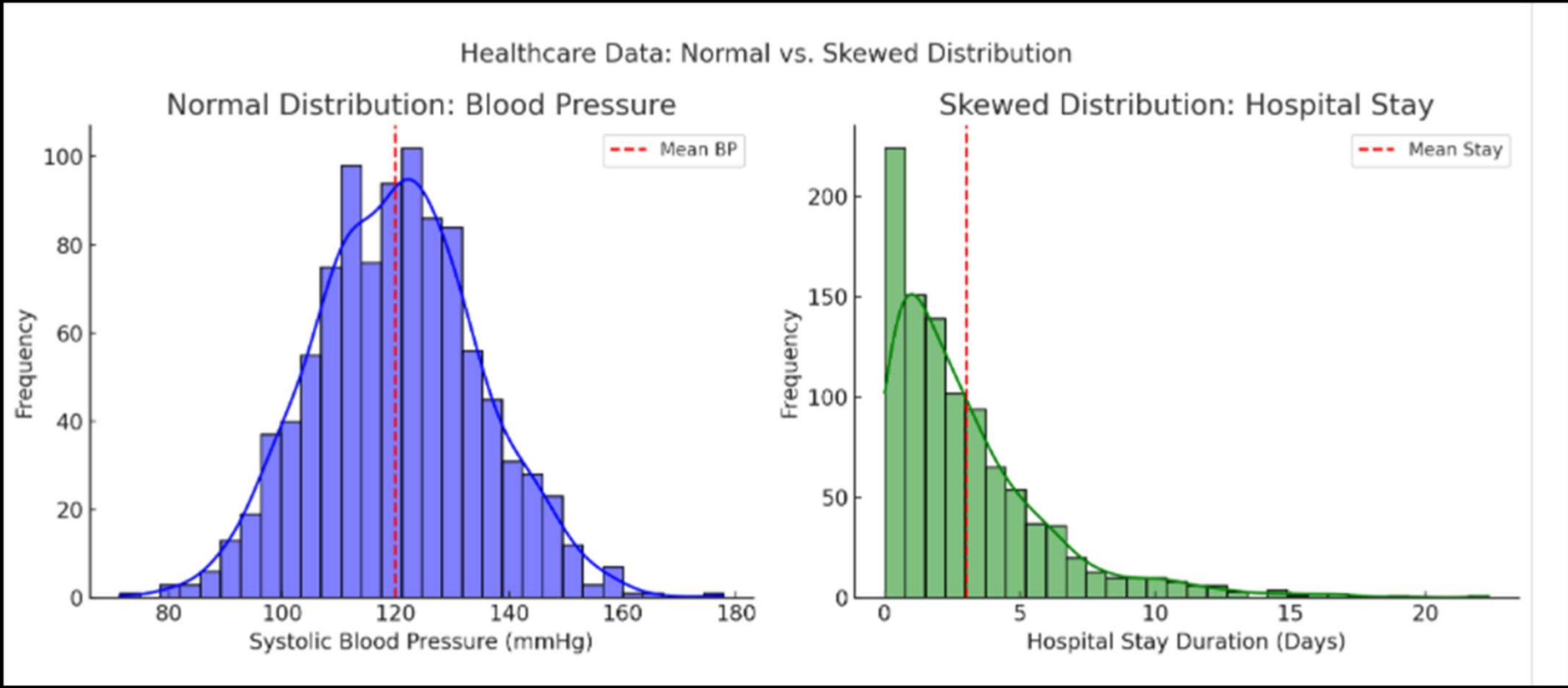
Regardless of the original distribution of a population, the distribution of the sample mean approaches a normal (Gaussian) distribution as the sample size increases, provided that:

- The samples are independent.
- The sample size is sufficiently large (typically $n \geq 30$).
- The population has a finite mean and variance.



Dice toll example

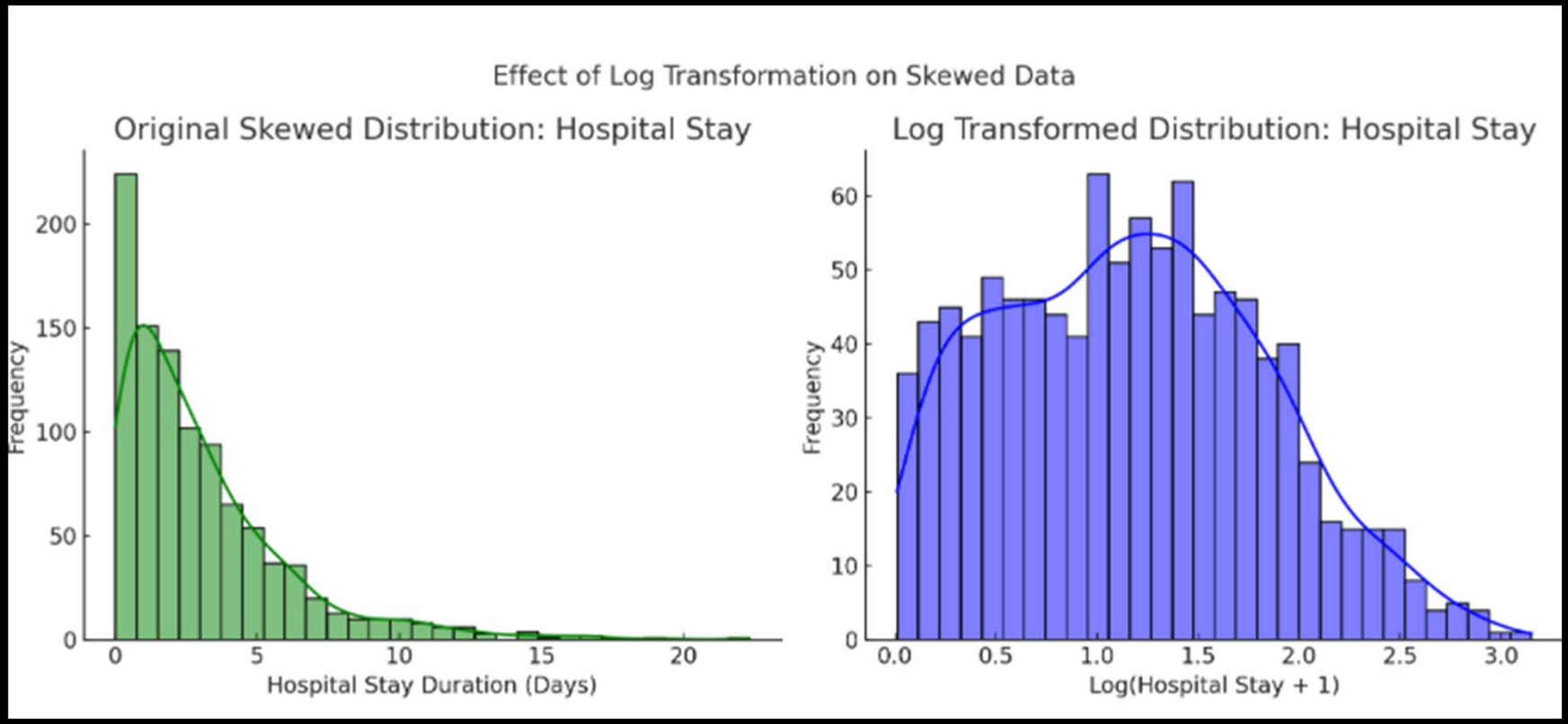
Skewed distribution



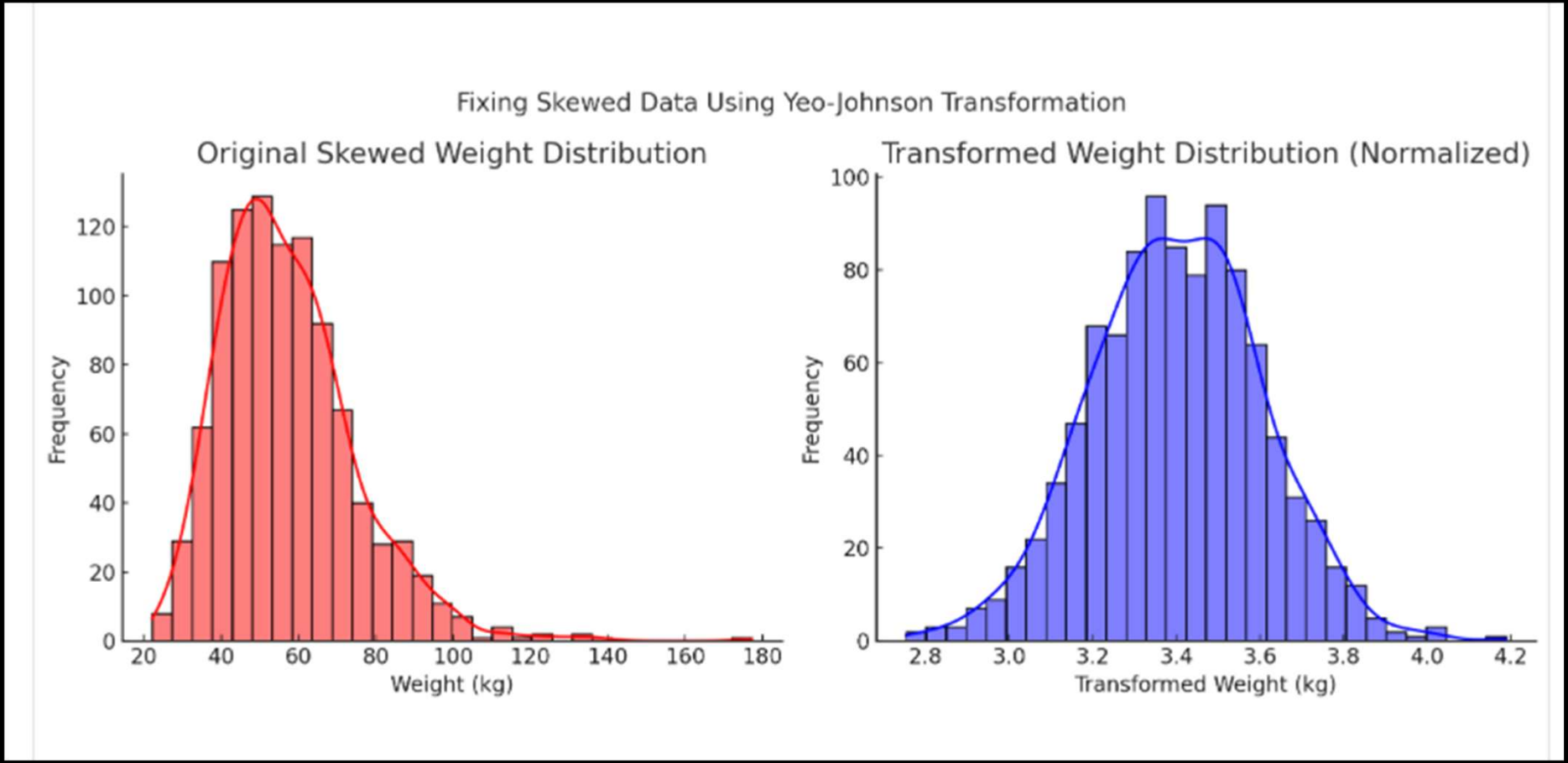
Normalizing skewed distribution

- Log transformation
  Used when data is right skewed, compresses large values, reduces right skewedness
- Square root transformation
  Right skewed data with non negative values. Reduces variance
- Box Cox transformation
  Used only for positive data. Adjusts skewness dynamically
- Yeo-Johnson transformation
  Modified Box Cox that supports zero and negative values as well
- Z-scale transformation, min-max scaling

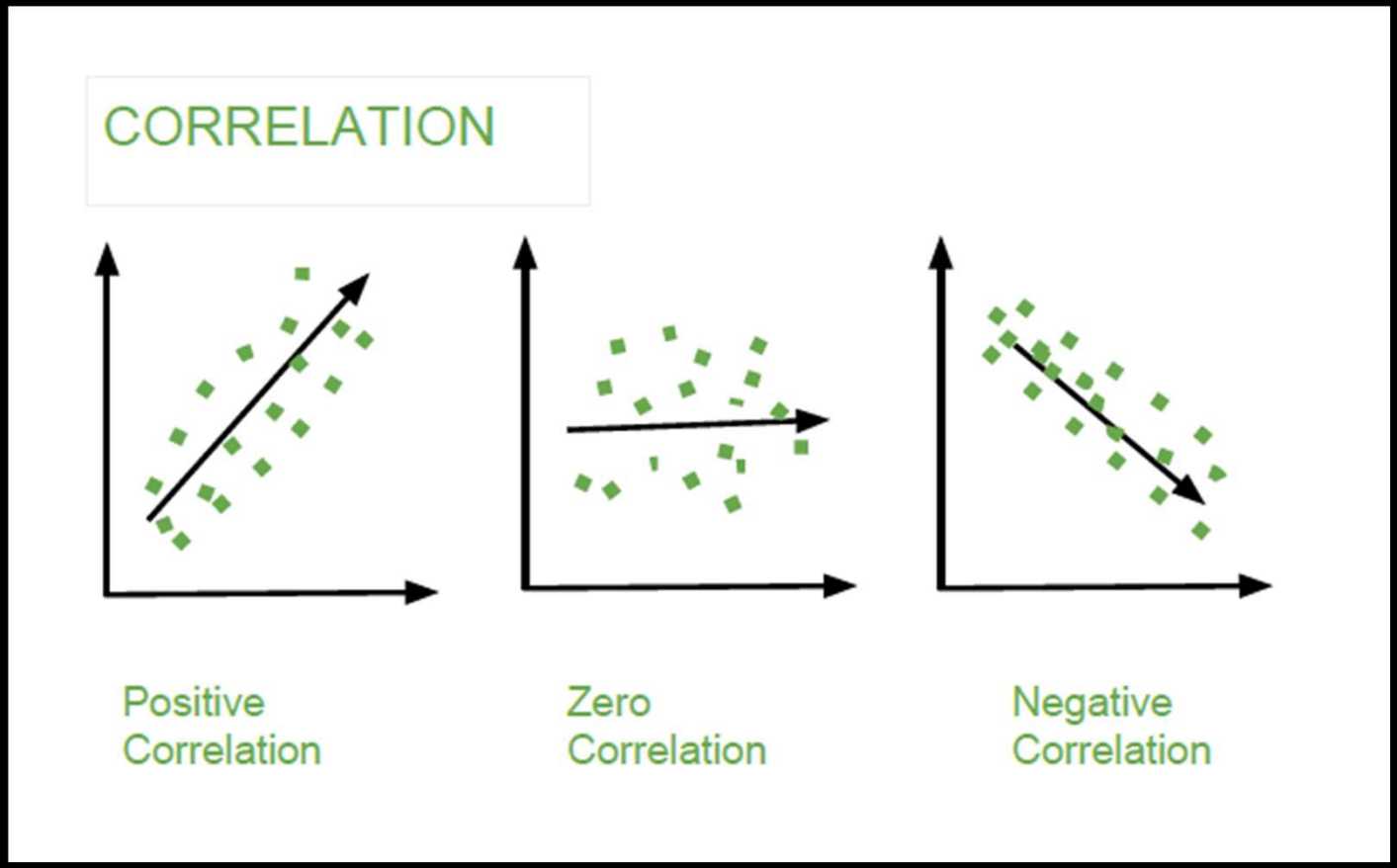
Applying log transformation to skewed data



Applying Yeo-Johnson transformation to skewed data



Key concept: Correlation



Key concept Hypothesis testing

A hypothesis is an assumed statement about a population's characteristics, often considered an opinion or claim about an issue.

Example:

The mean return of small-cap stock is higher than that of large-cap stock.

Null Hypothesis H0:

(Status Quo)
No significant difference between the mean return of small-cap and large-cap stock.

Alternate Hypothesis H1:

(Status Quo)
Mean return of small-cap stock is higher than large-cap stock.

Hypothesis and null hypothesis - example

Scenario:

Determine if there is an association between treatment type and the occurrence of side effects.

```
# Assume we have the following counts:
#               Side Effect: Yes   Side Effect: No
# Treatment A:           20                30
# Treatment B:           35                15
```

Null Hypothesis:
There is no significant association between treatment type and occurrence of side effects

Test: chi-squared

If p-value: <0.05
reject the null hypothesis. There is an association between treatment type and occurrence of side effects

Hypothesis testing for structured data

T-Test (Comparing Means)
Use case: Compare the means of two groups.
Example: Checking if average customer spending differs between two cities.

ANOVA (Analysis of Variance)
Use case: Compare means across multiple groups.
Example: Evaluating if sales performance differs across different regions.

Chi-Square Test for Independence
Use case: Determines if two categorical variables are related.
Example: Checking if customer churn is dependent on gender.

Correlation Analysis (Pearson/Spearman/Kendall)
Use case: Tests relationships between two numerical variables.
Example: Checking if there is a correlation between ad spend and revenue.

Hypothesis testing for unstructured data - text

Chi-Square Test for Text Features
Use case: Tests independence between categorical text-based features.
Example: Checking if a particular word usage pattern is linked to positive or negative reviews.

TF-IDF and Statistical Significance
Use case: Identifying significant keywords in text.
Example: Comparing word importance between fraudulent and non-fraudulent insurance claims.

Sentiment Score Comparison
Use case: T-tests or Mann-Whitney U tests to compare sentiment scores across groups.
Example: Checking if sentiment scores in customer feedback are significantly different before and after a product update.

Hypothesis testing for unstructured data – images, audio

Kolmogorov-Smirnov Test for Image Features
Use case: Comparing distributions of image feature vectors.
Example: Checking if defective and non-defective product images have significantly different pixel intensity distributions.

Chi-Square Test for Categorical Image Labels
Use case: Testing if image classification labels are evenly distributed.
Example: Checking if a face detection algorithm performs equally well across different ethnicities.

Spectral Analysis for Audio Data
Use case: Hypothesis testing on frequency components.
Example: Identifying differences in frequency patterns between normal and faulty machine sounds.

Conditional Probability and Bayes' Theorem

Conditional probability is the probability of an event occurring given that another event has already occurred.

Bayes' theorem is a way to update our belief about the probability of an event based on new evidence.

In healthcare, Bayes' theorem is widely used in diagnostic testing and risk assessment.
For example, suppose you want to know the probability that a patient actually has a disease given that they have tested positive for it. This is a classic application of Bayes' theorem.

- **Prevalence (Prior Probability)** $P(\text{Disease})$: 1% of the population has the disease.
- **Sensitivity** $P(\text{Test Positive}|\text{Disease})$: 95% (The probability the test is positive if the patient has the disease.)
- **Specificity** $P(\text{Test Negative}|\text{No Disease})$: 90% (The probability the test is negative if the patient does not have the disease.)

From these, we can calculate the probability that a patient has the disease given a positive test result, P(Disease|Test positive) using Bayes' theorem.

Descriptive Statistics and data visualization

Data Visualization

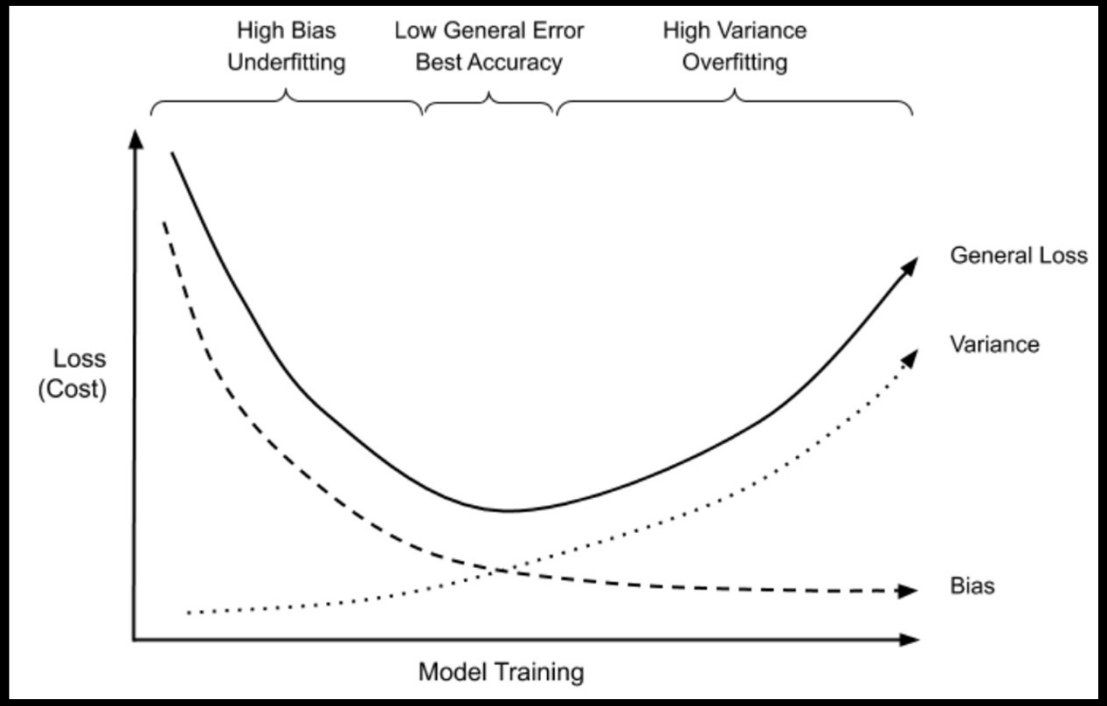
- **Histogram**: Plot the distribution of credit scores to see if they are normally distributed or skewed.
    - Example: A histogram might reveal that most applicants have a credit score between 600-800, but there are a few with scores below 500, indicating high-risk profiles.
- **Scatter Plot**: Plot Income vs. Loan Amount to see if there is a relationship between income and the loan size applicants seek.
    - Example: A scatter plot could show that higher-income applicants tend to apply for larger loans, but there are exceptions.
- **Correlation Matrix**: Create a correlation heatmap between variables like Credit Score, Income, and Loan Status to identify strong correlations.
    - Example: If you find a strong negative correlation between Credit Score and Loan Default, you can hypothesize that lower credit scores are associated with higher loan defaults.

Loss (Cost) Function

A loss function or cost function calculates the difference between true and estimated values.

Machine Learning models are trained to minimize a loss function.

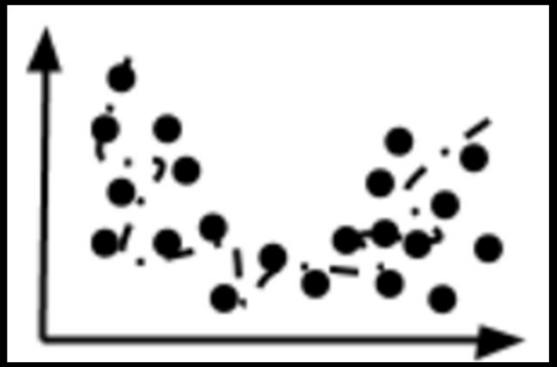
Effect of bias and variance on loss



Overfitting

Overfitting and underfitting

What are these diagrams depicting?



Descriptive Statistical Analytics – Demo and Hands On

Feature Engineering

- Dimensionality Reduction
- One hot encoding (convert categorical features into numerical)
- Normalization (Min-max scaling, Log transformation)
- Binning
- Adding new columns

Feature Engineering

Demo and hands-on

Case study – loan defaulters prediction EDA

https://www.kaggle.com/code/abhishek14398/loan-defaulters-prediction-eda-lending-club-study

Key outcomes of EDA

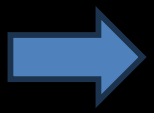
Insights from EDA:

- **Outliers**: You might discover that loans issued to applicants with very low incomes or credit scores have higher default rates.
- **Trends**: You could find a trend where applicants with a higher debt-to-income ratio are more likely to default on their loans.
- **Relationships**: A correlation matrix might reveal that Credit Score is the most important factor in determining loan default.

Enterprise Machine Learning with Microsoft Azure

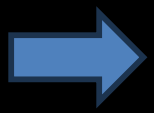
Data ingestion

- **Azure Data Factory**
  Low code/no code tool:
  Visual pipeline designer, control flows, event triggers, scheduling
- Data ingestion, ETL at scale
- Integrates with Databricks, Synapse
- For async http calls, Python asyncio, aiohttp are suitable



Data Eng., model training

- **Azure Databricks**
  data cleaning, feature engineering at scale with Apache Spark
- Distributed Model training with Apache Spark.
  Integrates with Hadoop
- Notebooks
- Metadata, Delta Tables



Data Storage

- Parquet files in Azure Data Lake or AWS S3
- Azure Synapse
  If you require complex sql queries, reporting or ad-hoc analysis on structured data