



# Generative AI Academy

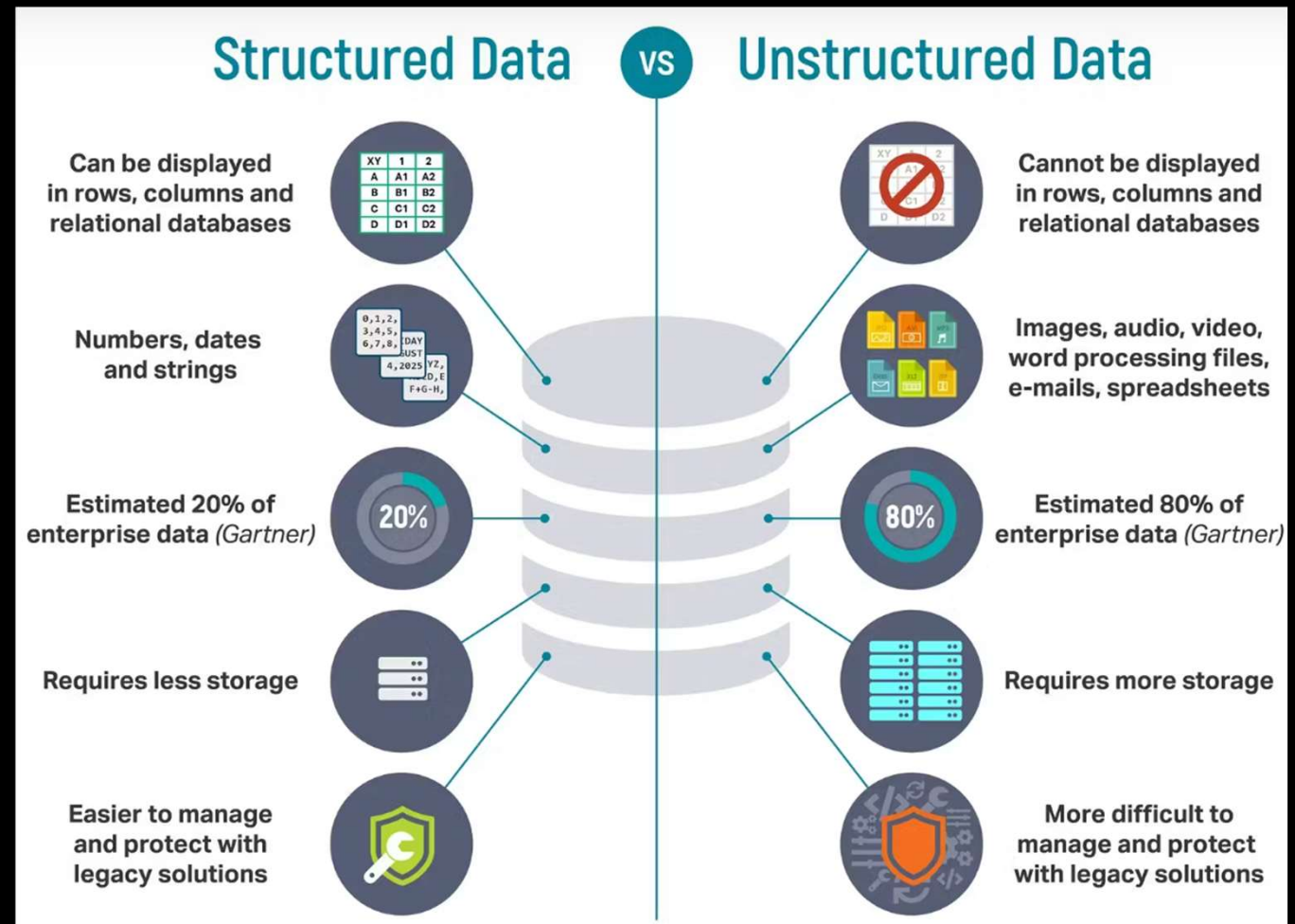
Our focus for this module:

Data Ingestion

Statistical Analysis

EDA

# Types of data



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

# Structured Data

## Examples of structured data



Pricing data



CRM data



Dates & times



Financial transactions



Customer account data



Medical information

# Examples of unstructured data



Text files and documents



Server, website, and applications logs



Sensor data



Image files



Video files



Audio files



Emails

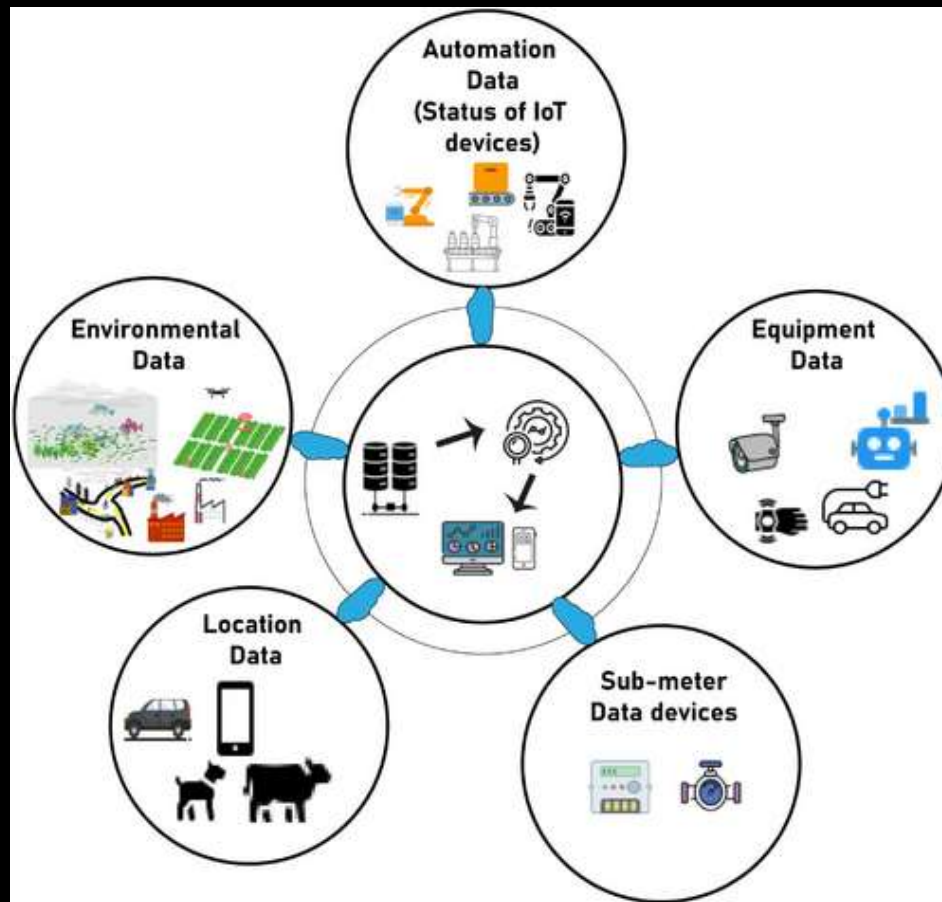


Social media data

# 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

```
def test():
    """Test function"""
    print("Hello, world!")
    return 1
```

```
def test():
    """Test function"""
    print("Hello, world!")
    return 1
```

```
def test():
    """Test function"""
    print("Hello, world!")
    return 1
```

## 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?

# 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 <b>new investment strategy</b> outperforms the old one.
Healthcare	Determining if a <b>new drug</b> is more effective than an existing one.
Marketing	Testing if <b>email marketing</b> improves customer retention.
Manufacturing	Verifying if a <b>new machine</b> reduces defects in production.
Economics	Checking if <b>inflation</b> affects stock market returns.
E-commerce	Testing if <b>free shipping</b> 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?

$${}^3C_2 = \frac{3!}{2!(3-2)!} = 3$$

Probability:  $\frac{3}{8}$

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?**

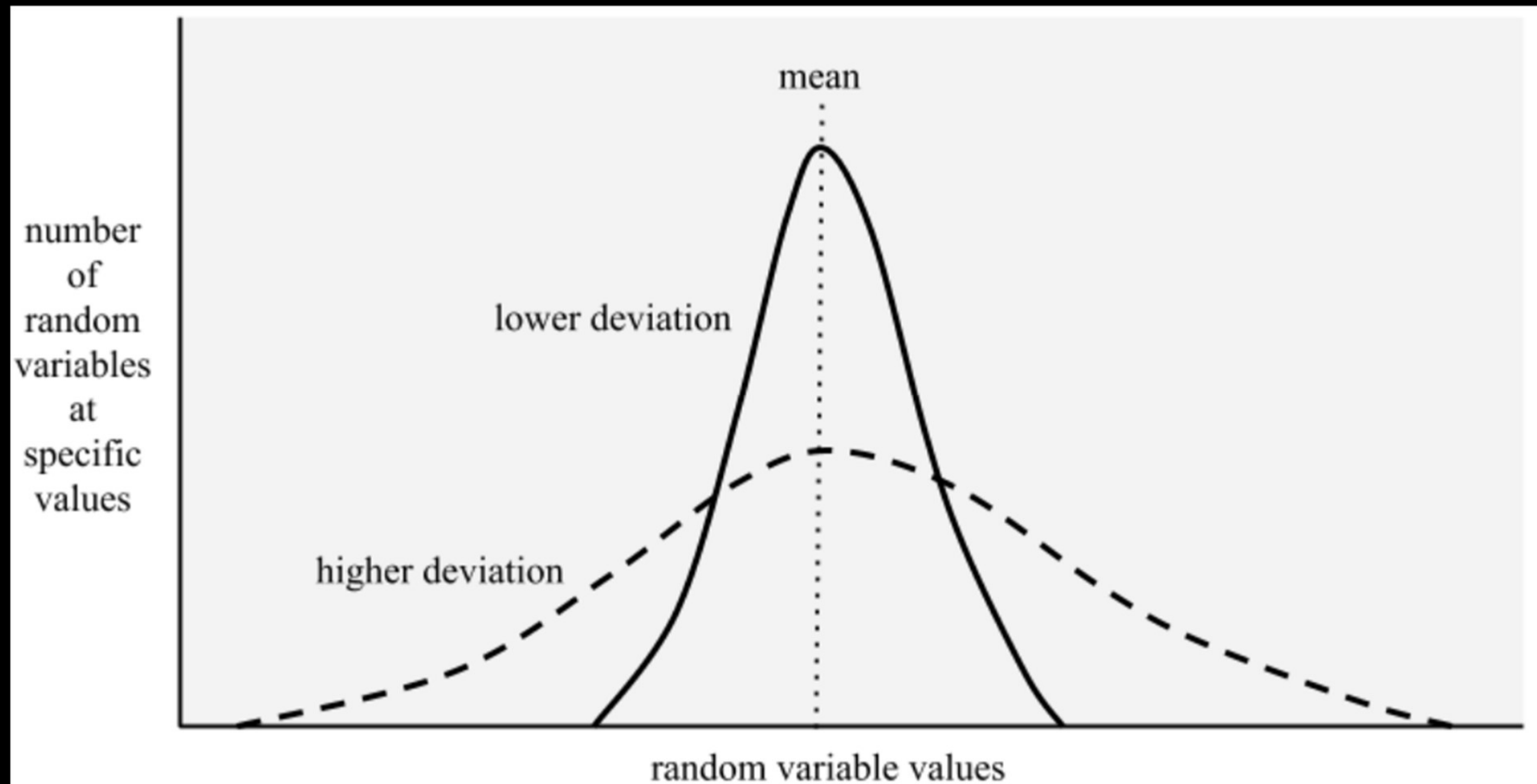
**High Variance: Noisy or overly complicated data**

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





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.

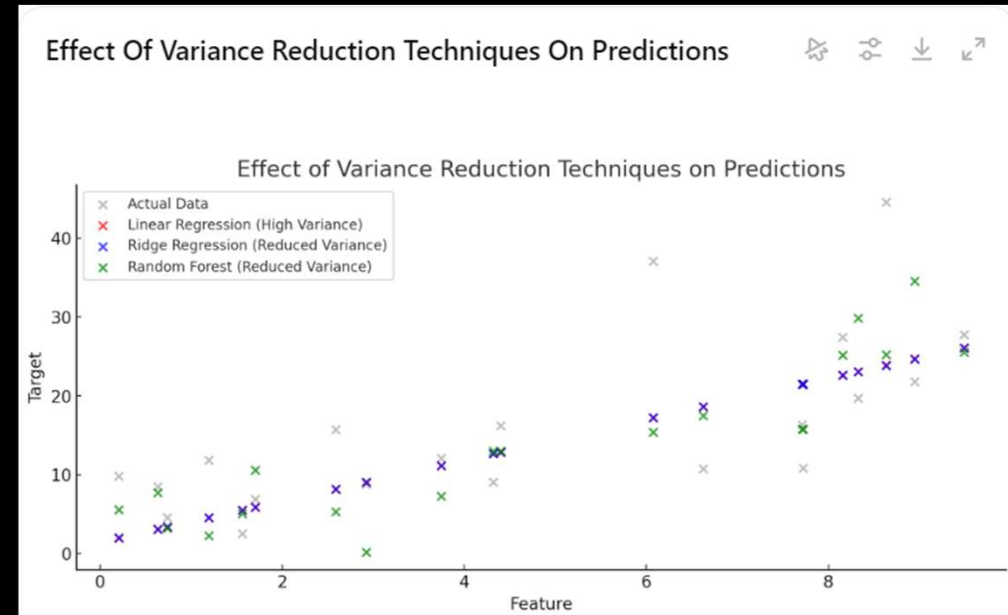
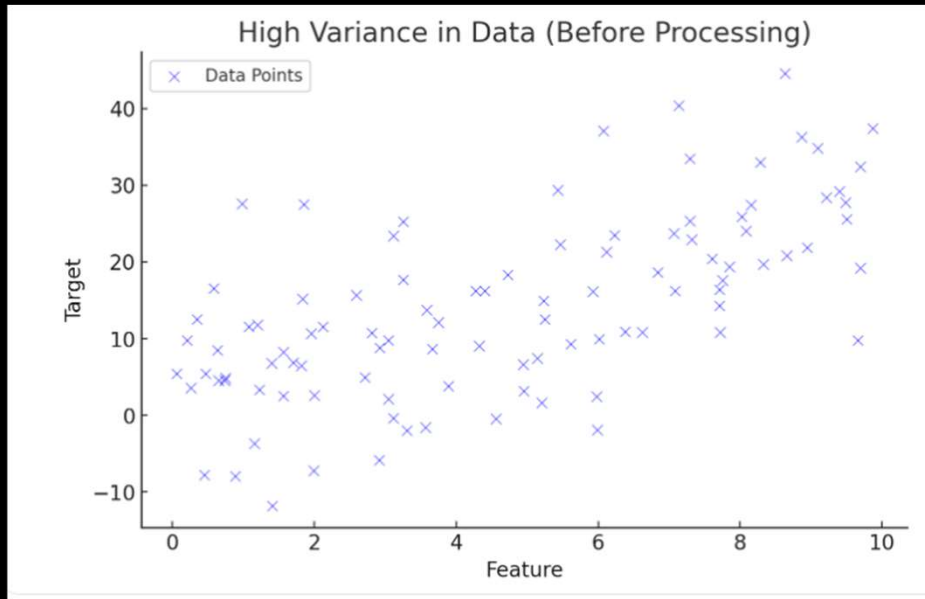
Data augmentation in computer vision (e.g., flipping, rotating images).

### **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, under sampling)

## **Improve Feature Representation**

Create More Informative Features: polynomial, quadratic, domain specific: margins instead of raw revenues, combine 2 features (use  $\text{age} * \text{income}$ )

## **Use More Expressive Models**

Switch to a More Complex Model:

If linear regression underfits, try decision trees or neural networks.

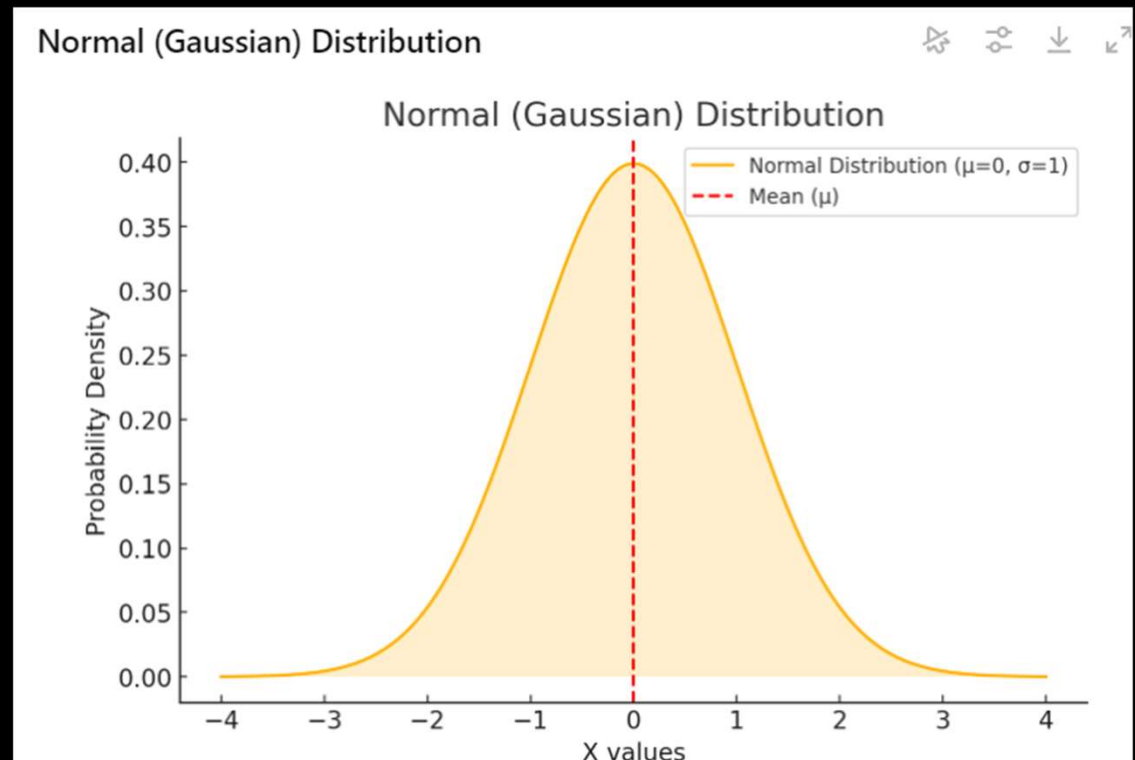
If a simple neural network underfits, add more layers or neurons.

# 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)

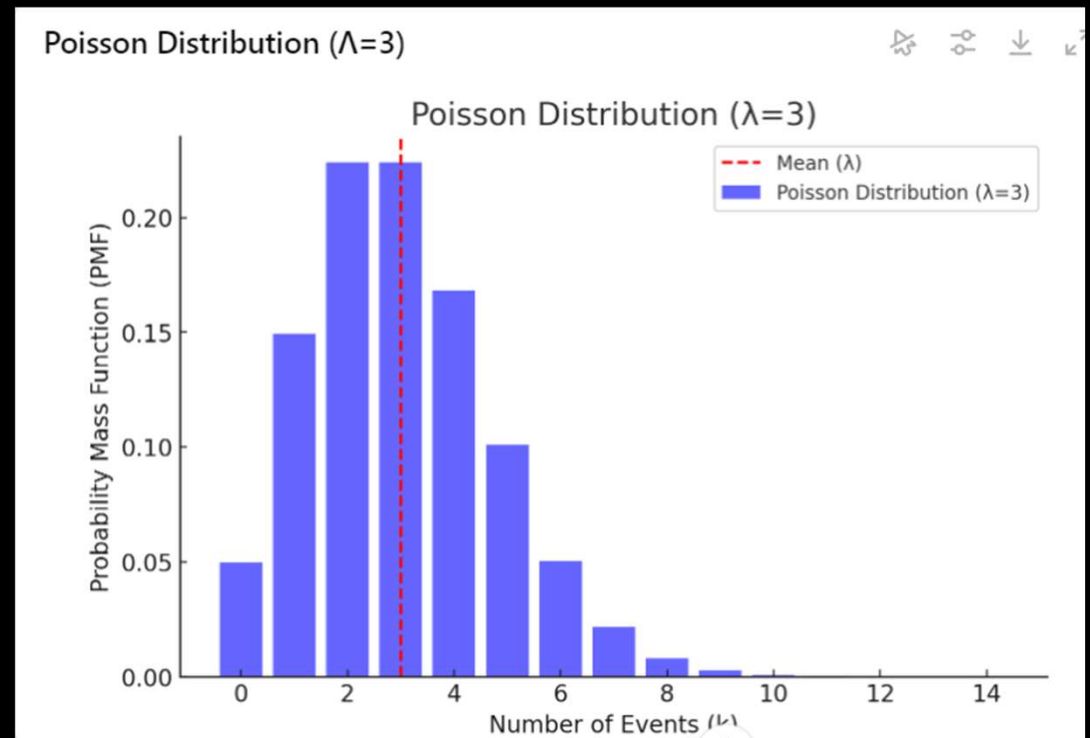


# Key concepts: Distribution

*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)

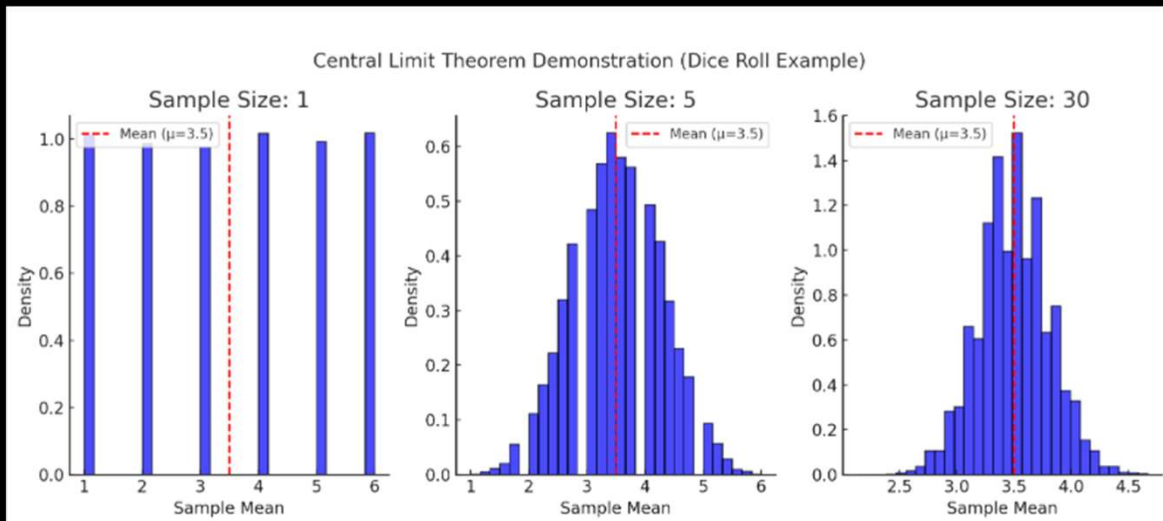


# Key concepts: Central Limit Theorem

The Central Limit Theorem (CLT) states that

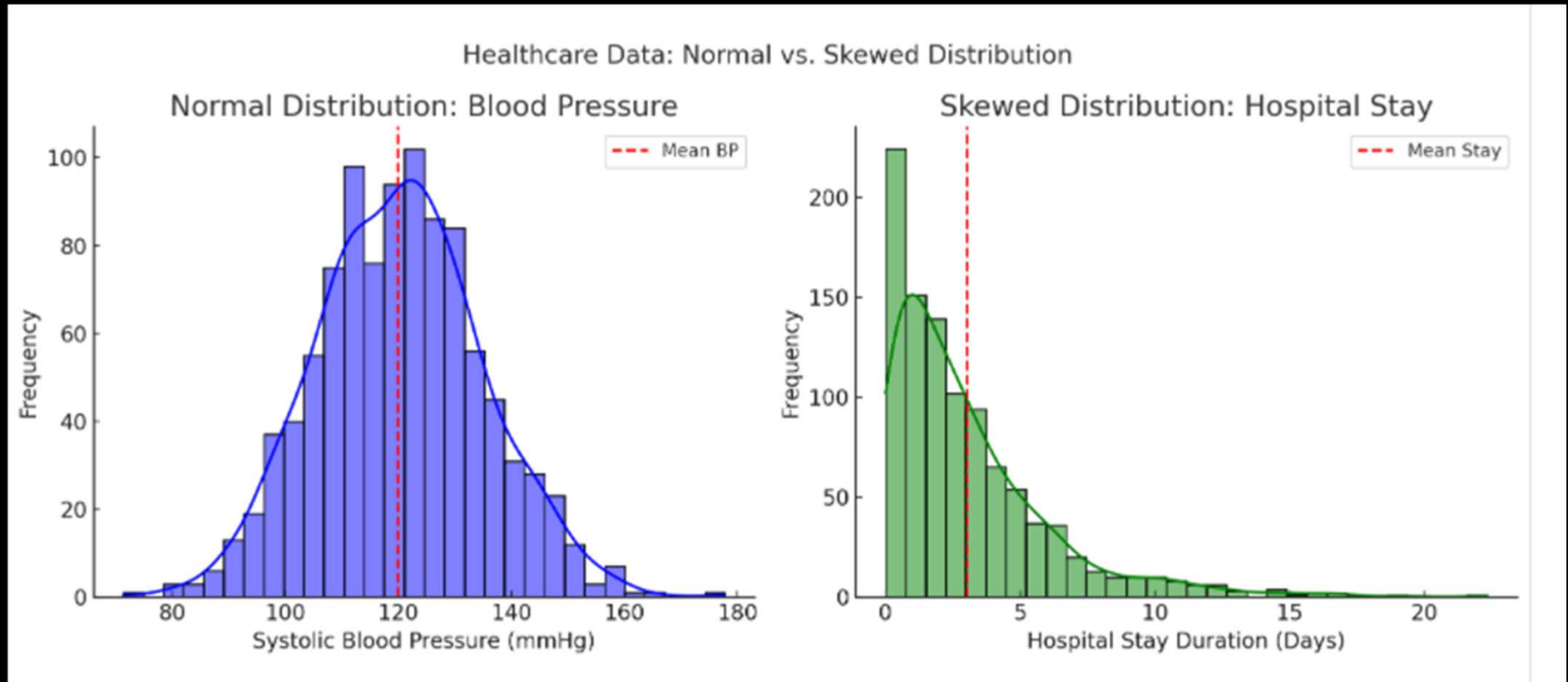
Regardless of the original distribution of a population, the distribution of the sample mean approaches a normal (Gaussian) distribution as the sample size increases, provided that:

- 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

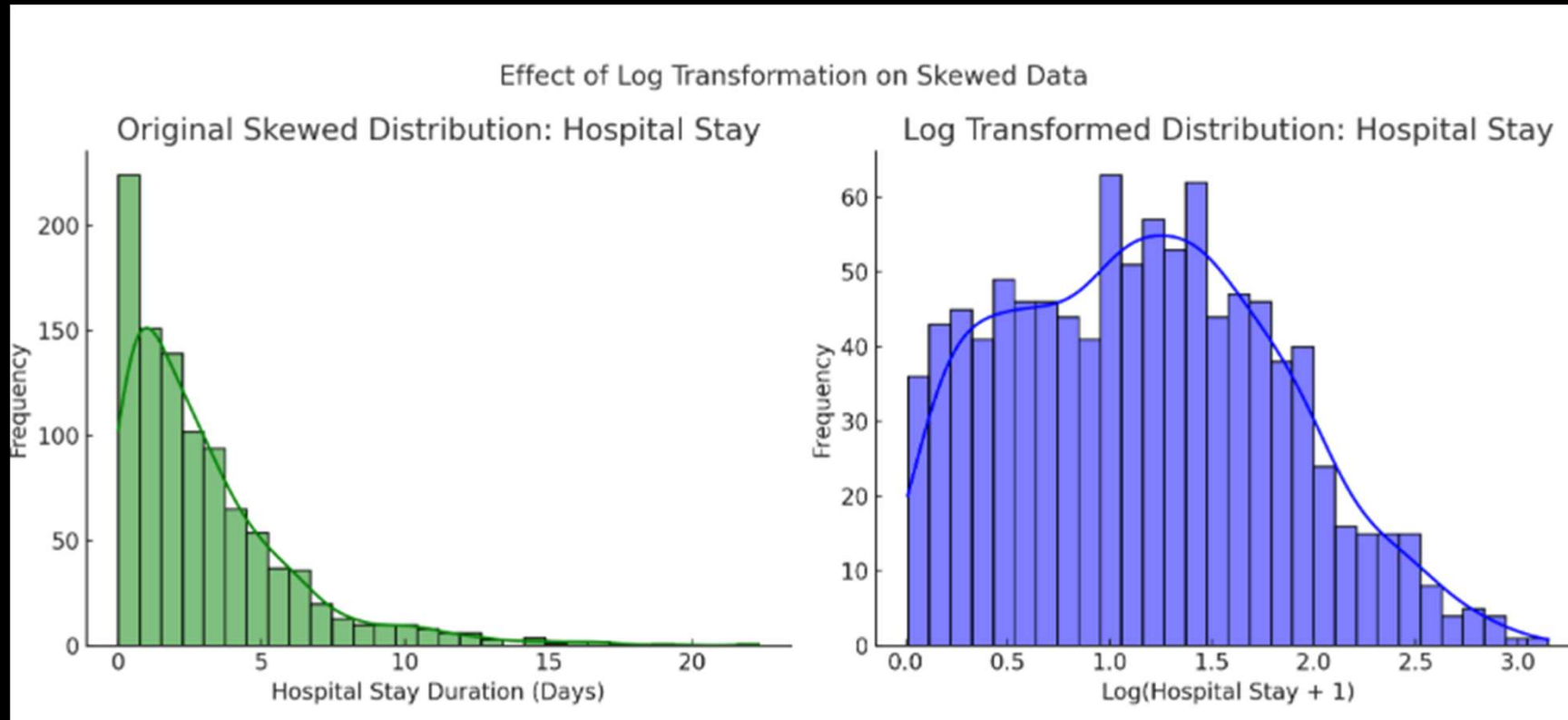




# Normalizing skewed distribution

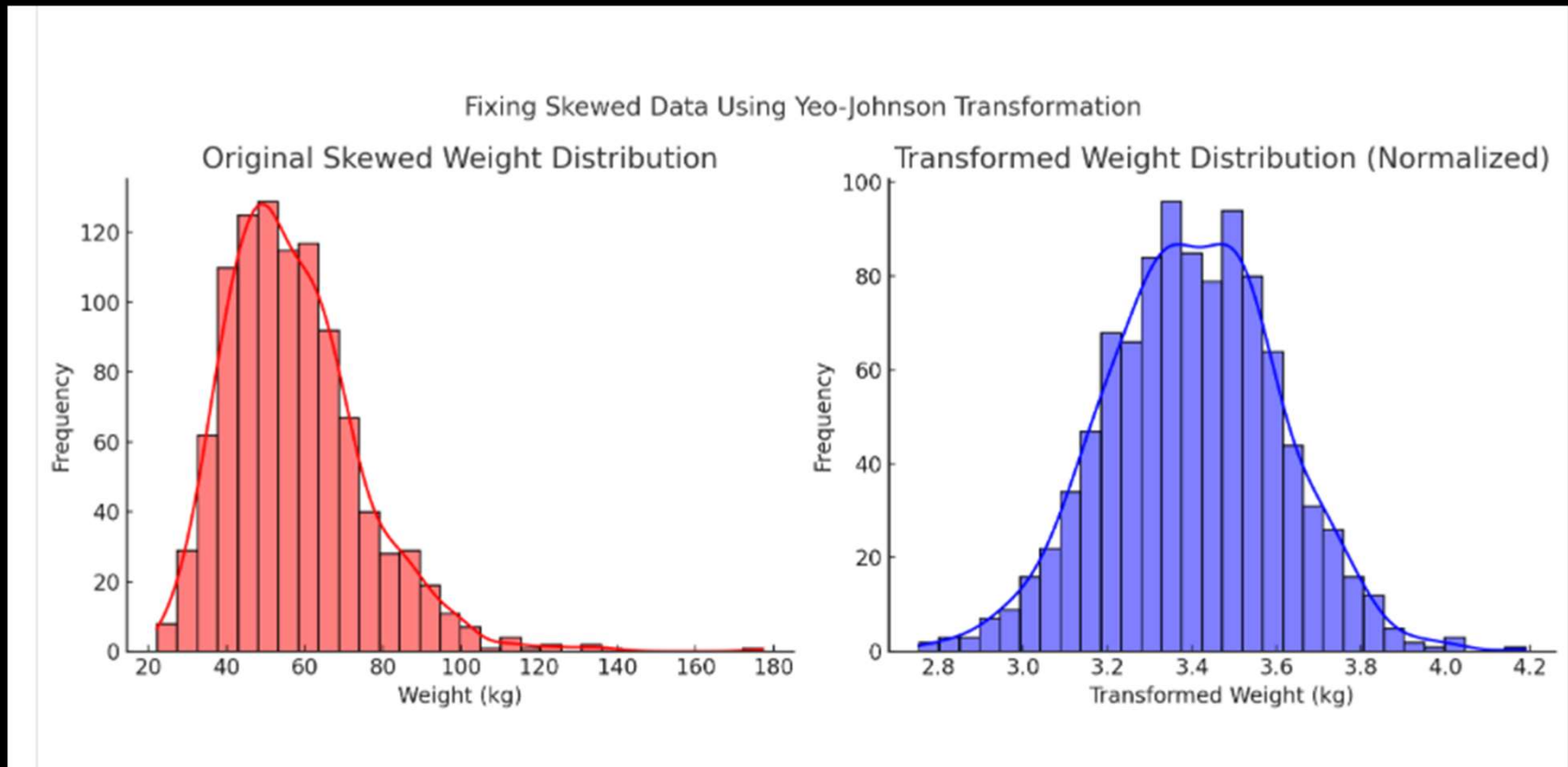
- **Log transformation**  
Used when data is right skewed, compresses large values, reduces right skewedness
- **Square root transformation**  
Right skewed data with non negative values. Reduces variance
- **Box Cox transformation**  
Used only for positive data. Adjusts skewness dynamically
- **Yeo-Johnson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

# Applying log transformation to skewed data



Compresses large values, reducing right skewness.  
Works only for positive values

# Applying Yeo-Johnson transformation to skewed data

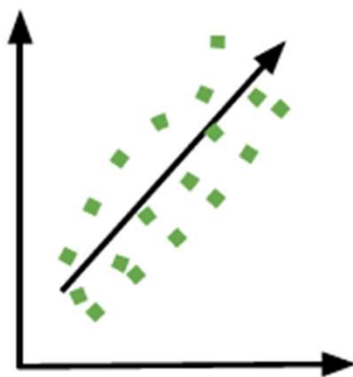


Box Cox transformation: Works only for positive data

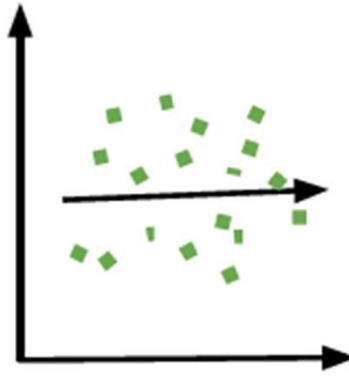
Yeo Johnson transformation: Works even when data has zero or negative values

# Key concept: Correlation

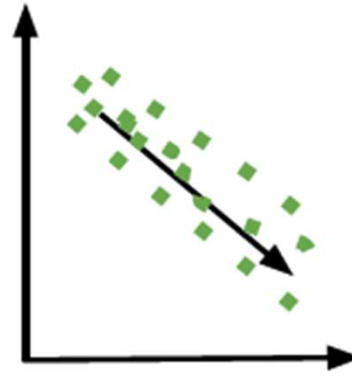
## CORRELATION



Positive  
Correlation



Zero  
Correlation



Negative  
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  $H_0$ :

(Status Quo)

No significant difference between the mean return of small-cap and large-cap stock.

Alternate Hypothesis  $H_1$ :

(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,  $P(\text{Disease}|\text{Test positive})$  using Bayes' theorem.

# Descriptive Statistics and data visualization

## Data Visualization

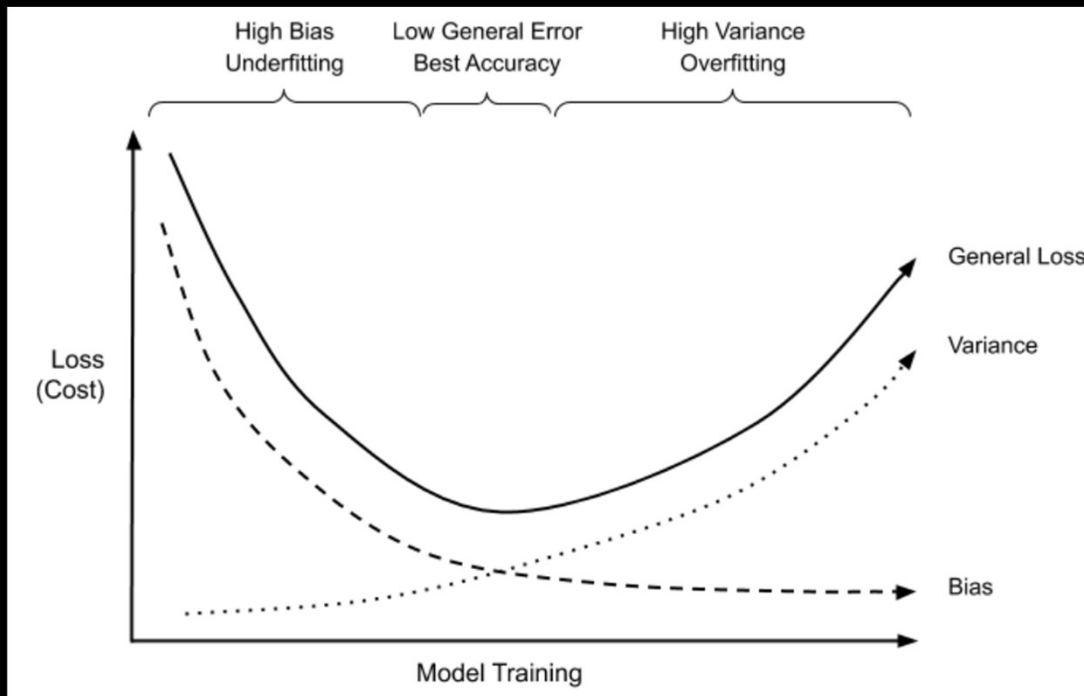
- **Histogram:** Plot the distribution of credit scores to see if they are normally distributed or skewed.
  - Example: A histogram might reveal that most applicants have a credit score between 600-800, but there are a few with scores below 500, indicating high-risk profiles.
- **Scatter Plot:** Plot 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.

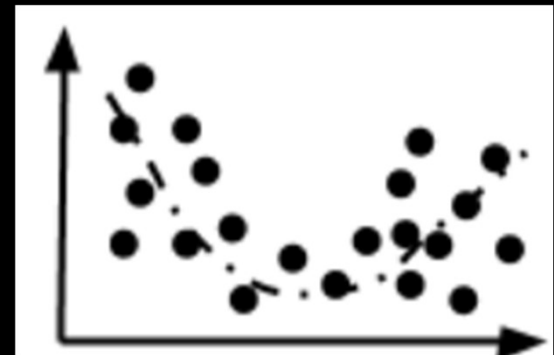
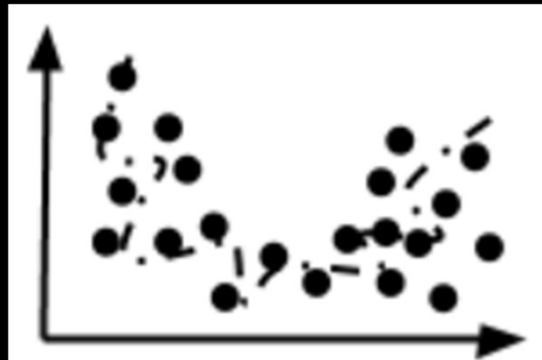
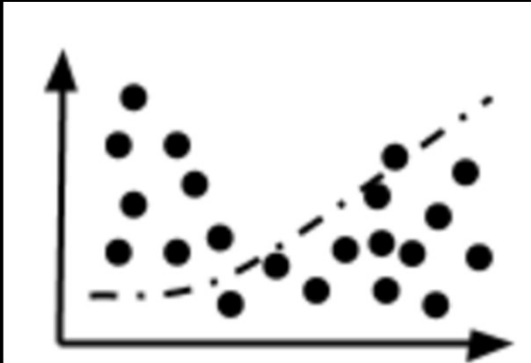
Effect of bias and variance on loss



# 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

## **Loan Defaulters Prediction | EDA | Lending Club Study**

Python · [Loan Classification Dataset](#)

<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.