|  |  |
| --- | --- |
|  |  |
| **Statistics and Data Preprocessing** |  |
|  |  |
|  | DATE – 10/09/2024  **Statistics and Data Preprocessing** |
|  | Zodex |

Table of Contents

[1. Introduction 1](#_Toc181022357)

[1.1 Core Concepts in AI 1](#_Toc181022358)

[1.2 Prerequisites 1](#_Toc181022359)

[1.3 Artificial Intelligence, Machine Learning, Deep Learning, GenAI 2](#_Toc181022360)

[1.4 Introduction to Data in Machine Learning 3](#_Toc181022361)

[1.5 Best Python libraries for Machine Learning (Link) 4](#_Toc181022362)

[1.5.1 NumPy 4](#_Toc181022363)

[1.5.2 SciPy 4](#_Toc181022364)

[1.5.3 Scikit-learn 4](#_Toc181022365)

[1.5.4 TensorFlow 4](#_Toc181022366)

[1.5.5 Keras 4](#_Toc181022367)

[1.5.6 PyTorch 5](#_Toc181022368)

[1.5.7 Pandas 5](#_Toc181022369)

[1.5.8 Matplotlib 5](#_Toc181022370)

[1.5.9 Most Used Library 5](#_Toc181022371)

[2. Statistics 6](#_Toc181022372)

[2.1 Tensor 6](#_Toc181022373)

[2.1.1 Tensor Ranks or No of Dimension or No of Axis 6](#_Toc181022374)

[2.2 Statistics 9](#_Toc181022375)

[2.3 Mean, Variance and Standard Deviation 9](#_Toc181022376)

[2.4 Types of Statistics 10](#_Toc181022377)

[2.5 Descriptive Statistic 10](#_Toc181022378)

[2.5.1 Types of Descriptive Statistics 10](#_Toc181022379)

[2.5.2 Applications of Descriptive Statistics 11](#_Toc181022380)

[2.6 Inferential Statistics 11](#_Toc181022381)

[2.6.1 Uses cases of Inferential Statistics 11](#_Toc181022382)

[2.6.2 Applications of Inferential Statistics 11](#_Toc181022383)

[2.7 Bias and Variance 11](#_Toc181022384)

[2.7.1 Bias 11](#_Toc181022385)

[2.7.2 Variance 12](#_Toc181022386)

[2.7.3 Different Combinations of Bias-Variance 13](#_Toc181022387)

[2.7.4 Bias Variance Tradeoff 13](#_Toc181022388)

[2.8 Hypothesis Testing 14](#_Toc181022389)

[2.8.1 Defining Hypotheses 14](#_Toc181022390)

[2.9 Covariance and Correlation 15](#_Toc181022391)

[2.9.1 Covariance 15](#_Toc181022392)

[2.9.2 Correlation 15](#_Toc181022393)

[2.9.3 Difference between Covariance and Correlation 19](#_Toc181022394)

[2.9.4 Applications of Covariance and Correlation 20](#_Toc181022395)

[3. Data and It’s Processing (Link) 21](#_Toc181022396)

[3.1 Data Type 21](#_Toc181022397)

[3.2 Data Cleaning vs Data Preprocessing 22](#_Toc181022398)

[3.3 Data Cleaning (Link) 22](#_Toc181022399)

[3.4 Data Preprocessing steps (Link) 23](#_Toc181022400)

[3.5 Exploratory Data Analysis (EDA) 23](#_Toc181022401)

[3.5.1 Type of EDA - Univariate, Bivariate and Multivariate data analysis 24](#_Toc181022402)

[3.6 Feature Engineering 24](#_Toc181022403)

[3.7 Feature Scaling Technique (Link)(Normalization or Standardization) 25](#_Toc181022404)

[3.7.1 Types of Feature Scaling: 25](#_Toc181022405)

[3.7.2 Normalization or Min-Max Scaling 25](#_Toc181022406)

[3.7.3 Standardization or Z-Score Normalization 26](#_Toc181022407)

[3.8 Encoding Technique 27](#_Toc181022408)

[3.8.1 Label Encoding 27](#_Toc181022409)

[3.8.2 One-Hot Encoding 28](#_Toc181022410)

[3.8.3 Dummy Encoding 29](#_Toc181022411)

[3.8.4 Ordinal Encoding 29](#_Toc181022412)

[3.8.5 Binary Encoding 30](#_Toc181022413)

[3.8.6 Frequency Encoding (Count Encoding) 30](#_Toc181022414)

[3.8.7 Target Encoding (Mean Encoding) 31](#_Toc181022415)

[3.8.8 Leave-One-Out Encoding 31](#_Toc181022416)

[3.9 Dummy variable trap in Regression Models 32](#_Toc181022417)

[3.10 Data Transformation 32](#_Toc181022418)

[3.10.1 Why do we need to use transformation on our data? 33](#_Toc181022419)

[3.10.2 Data Transformation Techniques: 33](#_Toc181022420)

[3.11 Outliers 35](#_Toc181022421)

[3.11.1 Outlier detection 35](#_Toc181022422)

[3.11.2 Outlier Removal 37](#_Toc181022423)

[3.11.3 Outlier vs Anomaly 37](#_Toc181022424)

[3.12 Feature Construction and Feature Splitting 37](#_Toc181022425)

[3.13 Dimensionality Reduction 38](#_Toc181022426)

[3.13.1 Different Techniques for Dimensionality Reduction 38](#_Toc181022427)

[3.14 Feature Selection 39](#_Toc181022428)

[3.14.1 Filter Method 40](#_Toc181022429)

[3.14.2 Wrapper Method: 41](#_Toc181022430)

[3.14.3 Intrinsic Method (Embedded Methods) 42](#_Toc181022431)

[3.15 Feature extraction 43](#_Toc181022432)

[3.15.1 Principal Component Analysis (PCA) 44](#_Toc181022433)

[3.16 Handling Imbalanced Data with SMOTE and Near Miss Algorithm 46](#_Toc181022434)

[3.16.1 SMOTE (Synthetic Minority Oversampling Technique) 46](#_Toc181022435)

[3.16.2 NearMiss Algorithm – Undersampling 47](#_Toc181022436)

[3.17 Best practices for pre-processing data 47](#_Toc181022437)

# Introduction

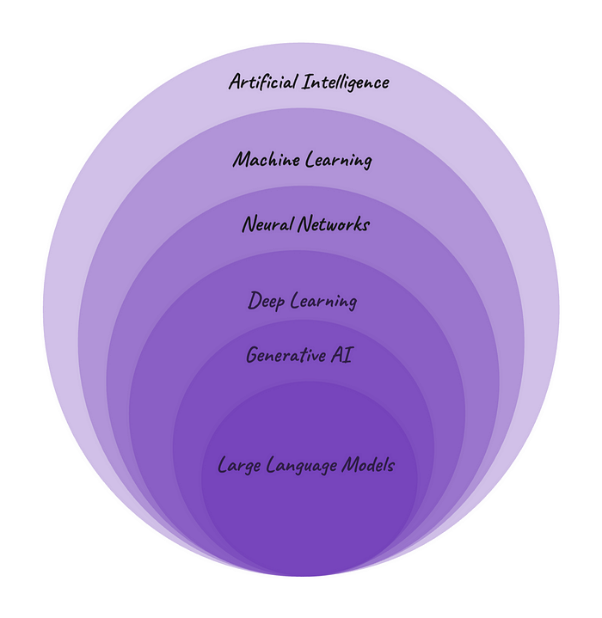
## Core Concepts in AI

1. [**Machine Learning (ML)**:](https://www.geeksforgeeks.org/machine-learning/) This is the backbone of AI, where algorithms learn from data without being explicitly programmed. It involves training an algorithm on a data set, allowing it to improve over time and make predictions or decisions based on new data.
2. [**Neural Networks**](https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/): Inspired by the human brain, these are networks of algorithms that mimic the way neurons interact, allowing computers to recognize patterns and solve common problems in the fields of AI, machine learning, and[deep learning.](https://www.geeksforgeeks.org/deep-learning-tutorial/)
3. [**Deep Learning**](https://www.geeksforgeeks.org/introduction-deep-learning/): A subset of ML, deep learning uses complex neural networks with many layers (hence “deep”) to analyse various factors of data. This is instrumental in tasks like image and speech recognition.
4. **Natural Language Processing (NLP)**:[NLP](https://www.geeksforgeeks.org/natural-language-processing-nlp-tutorial/) involves programming computers to process and analyse large amounts of natural language data, enabling interactions between computers and humans using natural language.
5. **Robotics**: While often associated with AI, robotics merges AI concepts with physical components to create machines capable of performing a variety of tasks, from assembly lines to complex surgeries.
6. [**Cognitive Computing**:](https://www.geeksforgeeks.org/cognitive-computing/) This AI approach mimics human brain processes to solve complex problems, often using pattern recognition, NLP, and[data mining.](https://www.geeksforgeeks.org/introduction-to-data-mining/)
7. **Expert Systems**: These are AI systems that emulate the decision-making ability of a human expert, applying reasoning capabilities to reach conclusions.

## Prerequisites

1. **Mathematics and Statistics:**
   * [**Linear Algebra:**](https://www.geeksforgeeks.org/linear-algebra/)
     + Learn vectors, matrices, and operations (addition, multiplication, inversion).
     + Study Eigenvalues and Eigenvectors.
   * [**Calculus**](https://www.geeksforgeeks.org/math-calculus/)**:**
     + Understand differentiation and integration.
     + Study partial derivatives and gradient descent.
   * [**Probability**](https://www.geeksforgeeks.org/probability-in-maths/)**and**[**Statistics**](https://www.geeksforgeeks.org/statistics-with-python/)**:**
     + Learn probability distributions (normal, binomial, Poisson).
     + Study Bayes’ theorem, expectation, variance, and hypothesis testing.
2. **Programming Skills:**
   * [**Python Programming**](https://www.geeksforgeeks.org/python-programming-language/)**:**
     + Basics: syntax, data structures (lists, dictionaries, sets), control flow (loops, conditionals).
     + Intermediate: functions, modules, object-oriented programming.
   * **Python Libraries for Data Science:**
     + [NumPy](https://www.geeksforgeeks.org/python-numpy/)for numerical computations.
     + [Pandas](https://www.geeksforgeeks.org/pandas-tutorial/)for data manipulation and analysis.
     + [Matplotlib](https://www.geeksforgeeks.org/python-introduction-matplotlib/)and [Seaborn](https://www.geeksforgeeks.org/introduction-to-seaborn-python/)for data visualization.
     + [Scikit-Learn](https://www.geeksforgeeks.org/what-is-python-scikit-library/) for machine learning algorithms.

## Artificial Intelligence, Machine Learning, Deep Learning, GenAI



* **Artificial Intelligence (AI)** is a discipline, a branch of computer science, that deals with the creation and development of machines that think and act like humans. AI powered technologies have been around for a while and some everyday examples are Siri and Alexa and customer service chatbots that pop up on websites.
* **Machine Learning (ML)** is a subfield of AI. It is a program or system that trains a model from input data and then that trained model can make useful predictions from new or never before seen data. So, ML gives the computer the ability to learn without explicitly programming. While in traditional programming, developers write explicit instructions for a computer to execute, in ML, algorithms learn patterns and relationships from data to make predictions or decisions. Unsupervised, Supervised and Reinforcement learning are the most common ML models. Google smart reply is an example of ML.
* **Neural Networks (NNs)**— also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning. The name and structure inspired by the human brain, mimicking the way that biological neurons signal to one another. NN consist of interconnected artificial neurons organised in layers: an input layer, one or more hidden layers, and an output layer. NN are at the heart of deep learning algorithms.
* **Deep Learning (DL)**is a subset of NN. The word deep here refers to the depth of layers in a neural network. Any neural network with more than three hidden layers can be considered a deep learning algorithm. Having a higher number of hidden layers, DL models are well-suited for tackling complex real-world problems. Everyday examples of technologies using NN and DL are: image recognition or object detection in smartphone cameras — such as Facial Recognition and Autofocus- and online language translation services like Google Translate.
* **Generative AI (GenAI)**is subset of DL, a type of artificial intelligence technology that can generate different types of content - such as text, imagery, audio, video - based on what has learnt from existing content.
* **Large Language Model (LLM)**is a form of generative AI, which focuses on generating human-like text based on the patterns learned from vast amounts of textual data during the training process. Note the difference between LLM and ML. Large Language Model is a specific type of machine learning model specialised in natural language processing, while ML is a broader field that encompasses various algorithms and techniques used across diverse domains to enable computers to learn from data and make predictions or decisions. ChatGPT is possibly the most famous example of technologies using LLM right now.

## Introduction to Data in Machine Learning

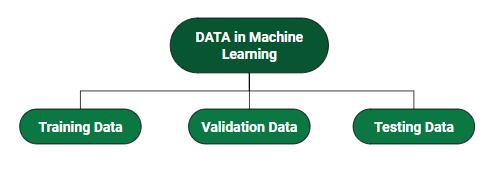
Data is typically divided into two types:

* Labeled data
* Unlabeled data

**DATA:** It can be any unprocessed fact, value, text, sound, or picture that is not being interpreted and analysed. Data is the most important part of all Data Analytics, Machine Learning, and Artificial Intelligence. Without data, we can’t train any model and all modern research and automation will go in vain. Big Enterprises are spending lots of money just to gather as much certain data as possible.

**Example:** Why did Facebook acquire WhatsApp by paying a huge price of $19 billion?





## Best Python libraries for Machine Learning ([Link](https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/))

Python libraries that are used in Machine Learning are:

### ****NumPy****

* **Purpose:** NumPy is a fundamental package for numerical computing in Python. It provides support for arrays (especially multi-dimensional arrays) and various mathematical operations like linear algebra, statistical functions, and more.
* **Usage:** Commonly used for matrix operations, handling large datasets, and providing support for scientific computing.

### ****SciPy****

* **Purpose:** SciPy builds on NumPy and provides additional utilities for scientific and technical computing, including modules for optimization, integration, interpolation, eigenvalue problems, and more.
* **Usage:** Primarily used in scientific computing fields like physics, biology, and engineering for solving mathematical problems.

### ****Scikit-learn****

* **Purpose:** Scikit-learn is a powerful machine learning library that provides simple and efficient tools for data analysis and machine learning, including classification, regression, clustering, dimensionality reduction, and model evaluation.
* **Usage:** Widely used for implementing traditional machine learning algorithms like decision trees, random forests, SVMs, and more.

### ****TensorFlow****

* **Purpose:** TensorFlow is an open-source machine learning framework developed by Google. It is widely used for building and deploying deep learning models, offering support for both high-level and low-level operations.
* **Usage:** Used for developing both deep learning models (like CNNs, RNNs) and large-scale machine learning applications, particularly in production environments.

### ****Keras****

* **Purpose:** Keras is a high-level deep learning API that runs on top of TensorFlow (and was also compatible with Theano in the past). It allows for easy and fast prototyping of neural networks with an intuitive API.
* **Usage:** Popular among beginners and practitioners for building and training deep learning models due to its simplicity and ease of use.

### ****PyTorch****

* **Purpose:** PyTorch is an open-source deep learning framework developed by Facebook. It provides dynamic computation graphs (as opposed to static ones like TensorFlow), making it popular for research.
* **Usage:** Widely used for research in deep learning due to its flexibility and popularity in academia. It is also gaining popularity in production environments.

### ****Pandas****

* **Purpose:** Pandas is a library that provides high-level data structures and functions designed to make data manipulation and analysis easy, particularly for working with structured data (like tables and time series).
* **Usage:** Commonly used for data cleaning, transformation, and analysis. It is often the first step in any data science or machine learning pipeline.

### ****Matplotlib****

* **Purpose:** Matplotlib is a 2D plotting library that enables users to create static, animated, and interactive visualizations in Python.
* **Usage:** Frequently used for generating plots, charts, histograms, and other data visualizations.

### ****Most Used Library****

* **Pandas** and **NumPy** are the most universally used libraries because they form the foundation of almost any data science or machine learning project.
* For machine learning, **Scikit-learn** is a go-to for traditional algorithms, while **TensorFlow** and **PyTorch** are the dominant players in deep learning.

# Statistics

## Tensor

A Tensor is a **N-dimensional Matrix**:

* A Scalar is a 0-dimensional tensor
* A Vector is a 1-dimensional tensor
* A Matrix is a 2-dimensional tensor

A **Tensor** is a generalization of **Vectors** and **Matrices** to higher dimensions.

Technically, all of the above are tensors, but when we speak of tensors, we generally speak of matrices with a dimension larger than 2 (**R > 2**).

### Tensor Ranks or No of Dimension or No of Axis

The number of directions a tensor can have in a **N**-dimensional space, is called the **Rank** of the tensor.

The rank is denoted **R**.

A **Scalar** is a single number.

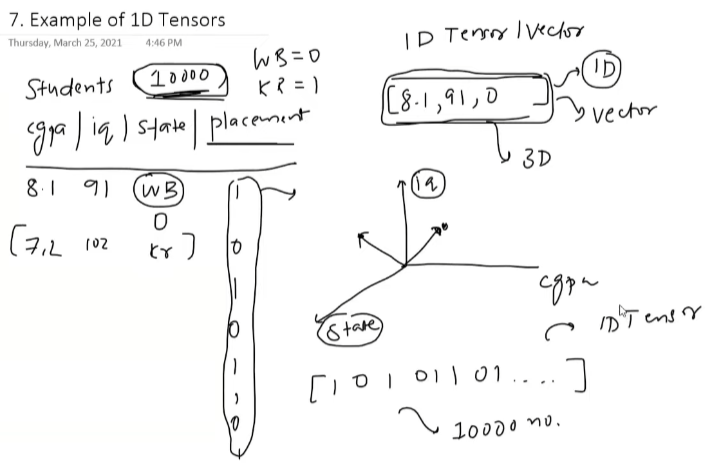
* It has 0 Axes
* It has a **Rank of 0**
* It is a 0-dimensional Tensor

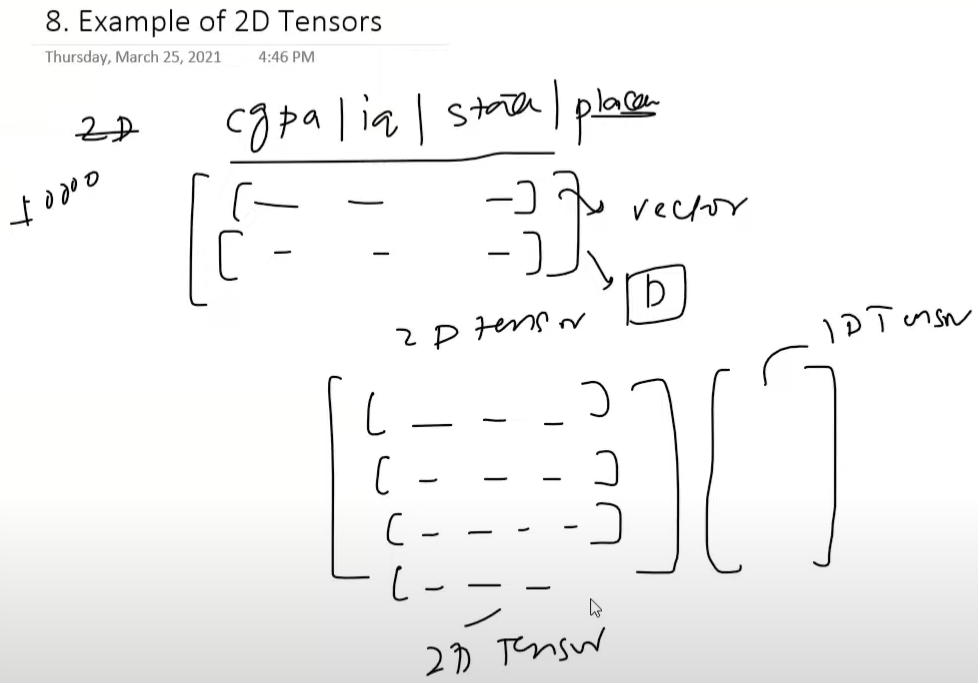
A **Vector** is an array of numbers.

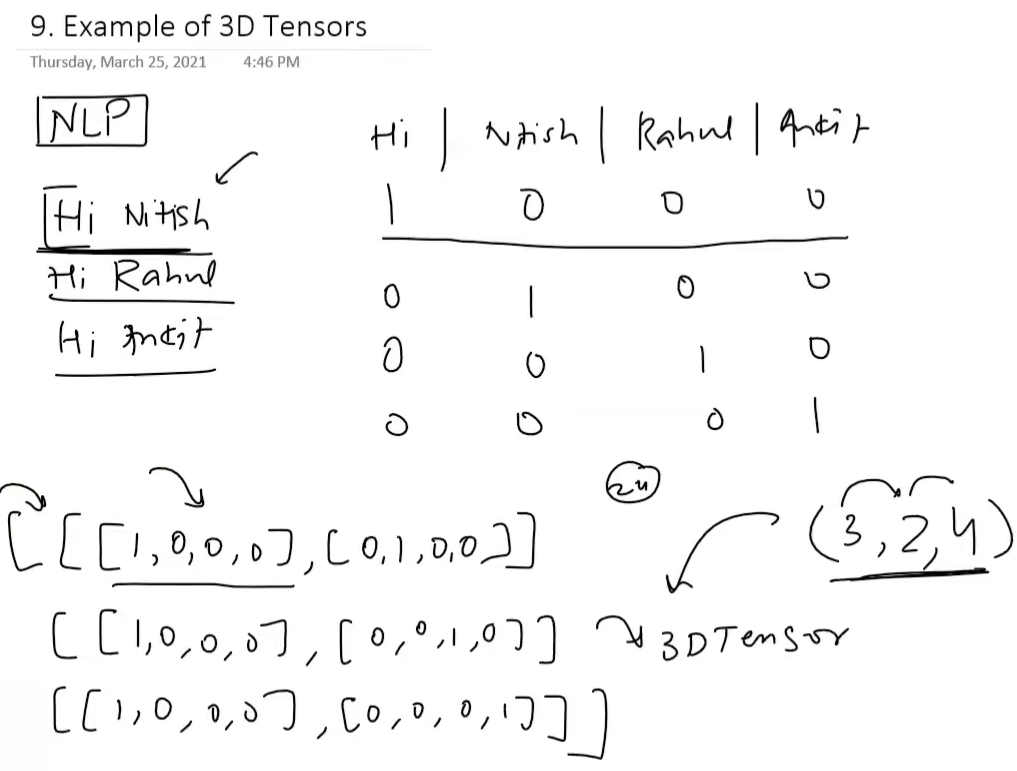
* It has 1 Axis
* It has a **Rank of 1**
* It is a 1-dimensional Tensor

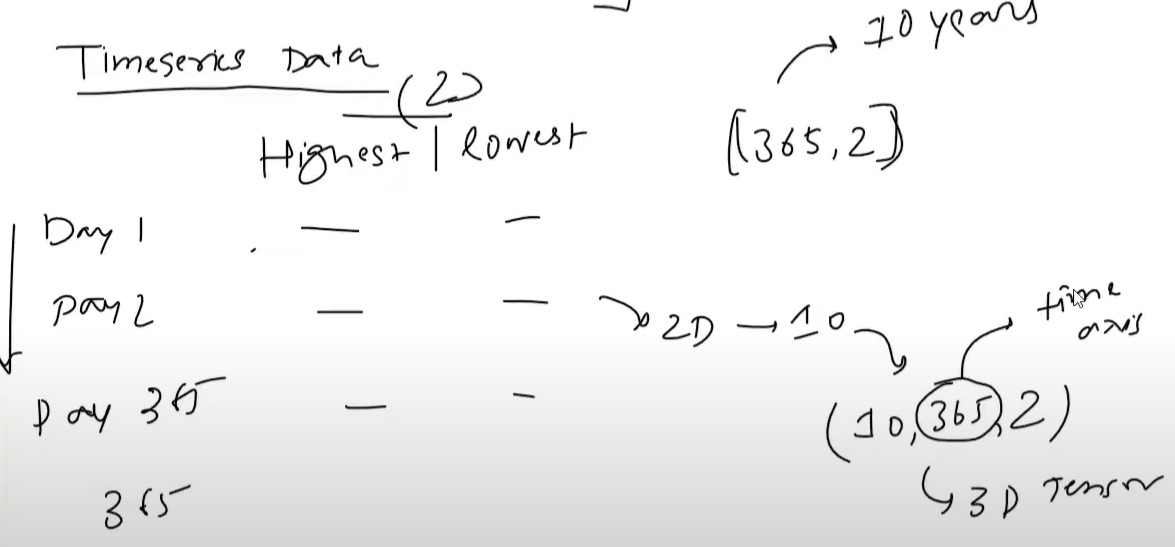
A **Matrix** is a 2-dimensional array.

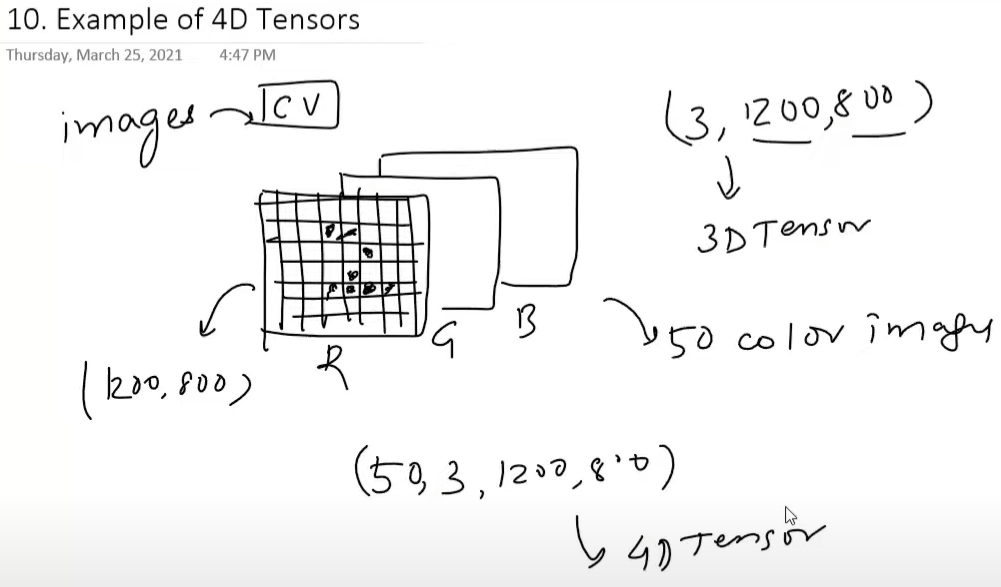
* It has 2 Axis
* It has a **Rank of 2**
* It is a 2-dimensional Tensor

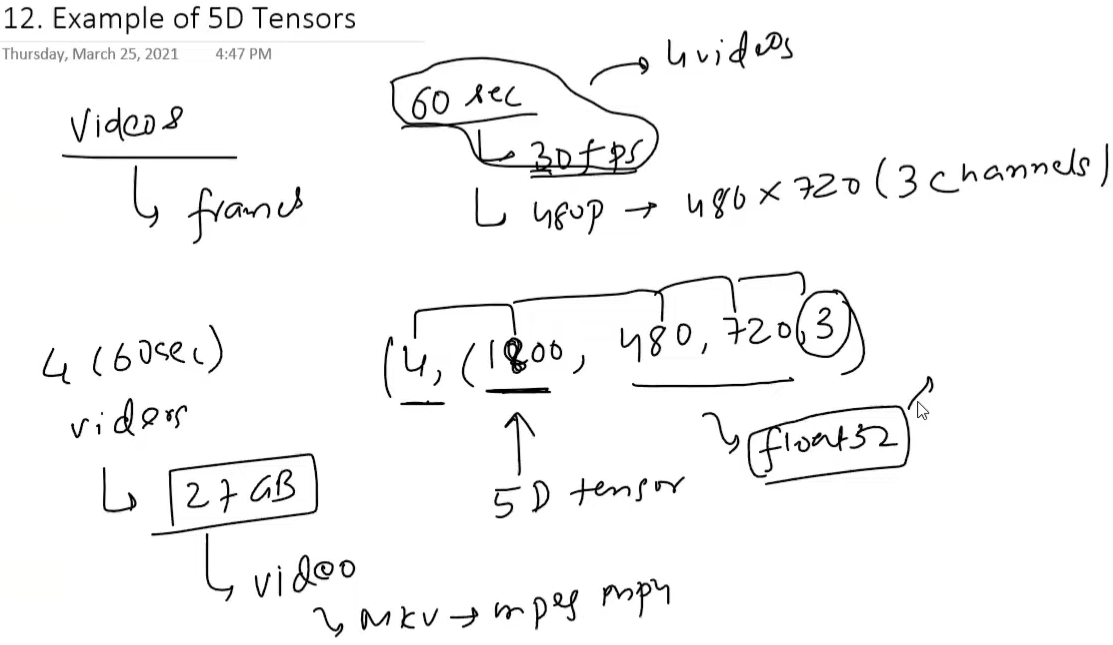








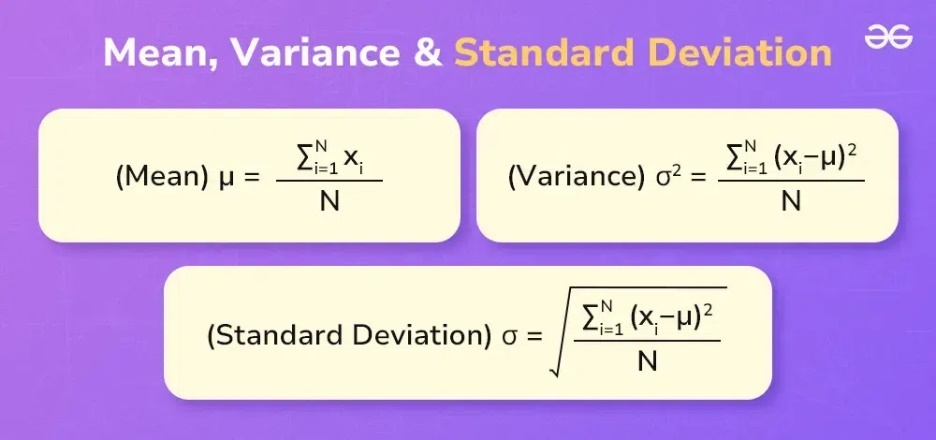




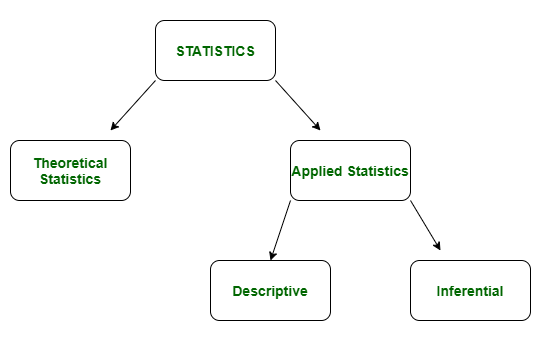
## Statistics

Statistics is a branch of mathematics dealing with the collection, analysis, interpretation, and presentation of masses of numerical data.

## Mean, Variance and Standard Deviation



## Types of Statistics



## Descriptive Statistic

In descriptive statistics, we describe our data in some manner and present it in a meaningful way so that it can be easily understood.

### Types of Descriptive Statistics

1. [Measures of Central Tendency](https://www.geeksforgeeks.org/measures-of-central-tendency/)

It represents the whole set of data by a single value. It gives us the location of the central points. There are three main measures of central tendency:

* Mean
* Mode
* Median

1. [Measure of Variability](https://www.geeksforgeeks.org/measures-of-spread-range-variance-and-standard-deviation/)

Measures of variability are also termed measures of dispersion as it helps to gain insights about the dispersion or the spread of the observations at hand. Some of the measures which are used to calculate the measures of dispersion in the observations of the variables are as follows:

* Range
* Variance
* Standard deviation

1. [Measures of Frequency Distribution](https://www.geeksforgeeks.org/frequency-distributions/)

Measures of frequency distribution help us gain valuable insights into the distribution and the characteristics of the dataset. Measures like,

* Count
* Frequency
* Relative Frequency
* Cumulative Frequency

### Applications of Descriptive Statistics

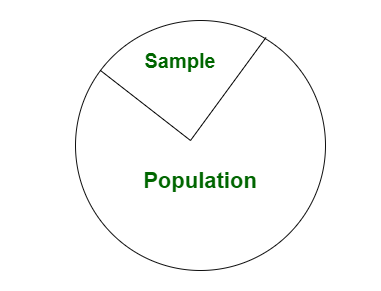
Business Analysis: Summarizing sales data to identify trends and make informed business decisions.

Healthcare: Analysing patient data to understand the distribution of health outcomes.

Engineering: Monitoring manufacturing processes through quality control charts to ensure consistency.

## Inferential Statistics

Inferential statistics is used to make predictions by taking any group of data in which you are interested. It can be defined as a random sample of data taken from a population to describe and make inferences about the population. Any group of data that includes all the data you are interested in is known as population. It basically allows you to make predictions by taking a small sample instead of working on the whole population.



### Uses cases of Inferential Statistics

Estimation

Hypothesis Testing

Regression Analysis

### Applications of Inferential Statistics

Market Research: Making predictions about consumer behavior based on survey samples.

Clinical Trials: Drawing conclusions about the effectiveness of new treatments from sample data.

Engineering: Predicting product performance and reliability through sample testing and analysis.

## Bias and Variance

### Bias

Bias is simply defined as the inability of the model because of that there is some difference or error occurring between the model’s predicted value and the actual value. These differences between actual or expected values and the predicted values are known as error or bias error or error due to bias. Bias is a systematic error that occurs due to wrong assumptions in the machine learning process.

* **Low Bias:** Low bias value means fewer assumptions are taken to build the target function. In this case, the model will closely match the training dataset.
* **High Bias:** High bias value means more assumptions are taken to build the target function. In this case, the model will not match the training dataset closely.

The high-bias model will not be able to capture the dataset trend. It is considered as the [underfitting](https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/) model which has a high error rate. It is due to a very simplified algorithm.

**Ways to reduce high bias:**

* **Use a more complex model:**One of the main reasons for high bias is the very simplified model. it will not be able to capture the complexity of the data. In such cases, we can make our mode more complex by increasing the number of hidden layers in the case of a [deep neural network.](https://www.geeksforgeeks.org/introduction-deep-learning/) Or we can use a more complex model like [Polynomial regression](https://www.geeksforgeeks.org/python-implementation-of-polynomial-regression/) for [non-linear datasets](https://www.geeksforgeeks.org/non-linear-regression-examples-ml/), [CNN](https://www.geeksforgeeks.org/introduction-convolution-neural-network/) for [image processing](https://www.geeksforgeeks.org/image-processing/), and [RNN](https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/) for sequence learning.
* **Increase the number of features:** By adding more features to train the dataset will increase the complexity of the model. And improve its ability to capture the underlying patterns in the data.
* **Reduce**[**Regularization**](https://www.geeksforgeeks.org/regularization-in-machine-learning/)**of the model:**Regularization techniques such as [L1 or L2 regularization](https://www.geeksforgeeks.org/ml-implementing-l1-and-l2-regularization-using-sklearn/) can help to prevent [overfitting](https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/) and improve the generalization ability of the model. if the model has a high bias, reducing the strength of regularization or removing it altogether can help to improve its performance.
* **Increase the size of the training data:** Increasing the size of the training data can help to reduce bias by providing the model with more examples to learn from the dataset.

### Variance

Variance is the amount by which the performance of a predictive model changes when it is trained on different subsets of the training data. More specifically, variance is the variability of the model that how much it is sensitive to another subset of the training dataset. i.e. how much it can adjust on the new subset of the training dataset.

Variance errors are either low or high-variance errors.

* **Low variance:** Low variance means that the model is less sensitive to changes in the training data and can produce consistent estimates of the target function with different subsets of data from the same [distribution](https://www.geeksforgeeks.org/introduction-of-statistical-data-distributions/). This is the case of **underfitting** when the model fails to generalize on both training and test data.
* **High variance:** High variance means that the model is very sensitive to changes in the training data and can result in significant changes in the estimate of the target function when trained on different subsets of data from the same distribution. This is the case of **overfitting** when the model performs well on the training data but poorly on new, unseen test data. It fits the training data too closely that it fails on the new training dataset.

**Ways to Reduce the reduce Variance in Machine Learning:**

* [**Cross-validation**](https://www.geeksforgeeks.org/cross-validation-machine-learning/)**:** By splitting the data into training and testing sets multiple times, cross-validation can help identify if a model is overfitting or underfitting and can be used to tune hyperparameters to reduce variance.
* [**Feature selection:**](https://www.geeksforgeeks.org/feature-selection-techniques-in-machine-learning/)By choosing the only relevant feature will decrease the model’s complexity. and it can reduce the variance error.
* [**Regularization**](https://www.geeksforgeeks.org/regularization-in-machine-learning/)**:** We can use L1 or L2 regularization to reduce variance in machine learning models
* [**Ensemble methods**](https://www.geeksforgeeks.org/ensemble-classifier-data-mining/)**:** It will combine multiple models to improve generalization performance. [Bagging, boosting](https://www.geeksforgeeks.org/bagging-vs-boosting-in-machine-learning/), and stacking are common ensemble methods that can help reduce variance and improve generalization performance.
* **Simplifying the model:**Reducing the complexity of the model, such as decreasing the number of parameters or layers in a neural network, can also help reduce variance and improve generalization performance.
* [**Early stopping**](https://www.geeksforgeeks.org/choose-optimal-number-of-epochs-to-train-a-neural-network-in-keras/)**:** Early stopping is a technique used to prevent overfitting by stopping the training of the deep learning model when the performance on the validation set stops improving.

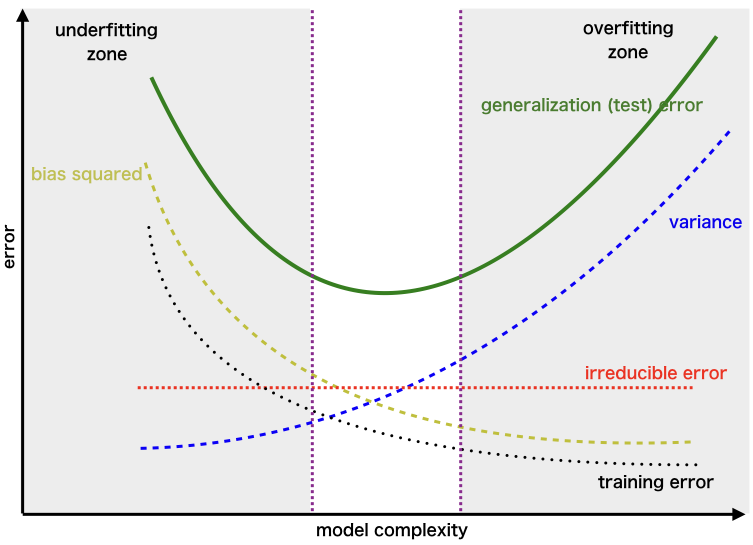
### Different Combinations of Bias-Variance

There can be four combinations between bias and variance.

* **High Bias, Low Variance:** A model with high bias and low variance is said to be underfitting.
* **High Variance, Low Bias:**A model with high variance and low bias is said to be overfitting.
* **High-Bias, High-Variance:**A model has both high bias and high variance, which means that the model is not able to capture the underlying patterns in the data (high bias) and is also too sensitive to changes in the training data (high variance). As a result, the model will produce inconsistent and inaccurate predictions on average.
* **Low Bias, Low Variance:** A model that has low bias and low variance means that the model is able to capture the underlying patterns in the data (low bias) and is not too sensitive to changes in the training data (low variance). This is the ideal scenario for a machine learning model, as it is able to generalize well to new, unseen data and produce consistent and accurate predictions. But in practice, it’s not possible.

### Bias Variance Tradeoff

If the algorithm is too simple (hypothesis with linear equation) then it may be on high bias and low variance condition and thus is error-prone. If algorithms fit too complex (hypothesis with high degree equation) then it may be on high variance and low bias. In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as a Trade-off or Bias Variance Trade-off. This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can’t be more complex and less complex at the same time. For the graph, the perfect tradeoff will be like this.



## Hypothesis Testing

Hypothesis testing is a statistical method that is used to make a statistical decision using experimental data. Hypothesis testing is basically an assumption that we make about a population parameter. It evaluates two mutually exclusive statements about a population to determine which statement is best supported by the sample data.

To test the validity of the claim or assumption about the population parameter:

* A sample is drawn from the population and analysed.
* The results of the analysis are used to decide whether the claim is true or not.

***Example:*** *You say an average height in the class is 30 or a boy is taller than a girl. All of these is an assumption that we are assuming, and we need some statistical way to prove these. We need some mathematical conclusion whatever we are assuming is true.*

### Defining Hypotheses

* **Null hypothesis (H0):**In statistics, the null hypothesis is a general statement or default position that there is no relationship between two measured cases or no relationship among groups. In other words, it is a basic assumption or made based on the problem knowledge.  
  **Example:** A company’s mean production is 50 units/per da H0: μ*μ* = 50.
* **Alternative hypothesis (H1):**The alternative hypothesis is the hypothesis used in hypothesis testing that is contrary to the null hypothesis.

**Example:** A company’s production is not equal to 50 units/per day i.e. H1:μ*μ*≠=50.

## Covariance and Correlation

### Covariance

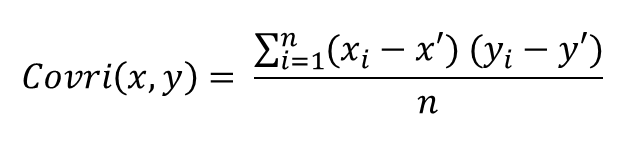
**Covariance is a statistical measure** that **indicates the direction of the linear relationship between two variables**. It assesses how much two variables change together from their mean values.

Types of Covariance:

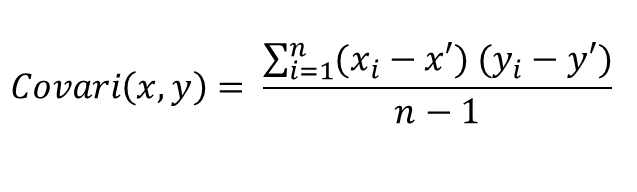
* **Positive Covariance**: When one variable increases, the other variable tends to increase as well, and vice versa.
* **Negative Covariance**: When one variable increases, the other variable tends to decrease.
* **Zero Covariance**: There is no linear relationship between the two variables; they move independently of each other.

#### ****Covariance Formula****

**For Population:**

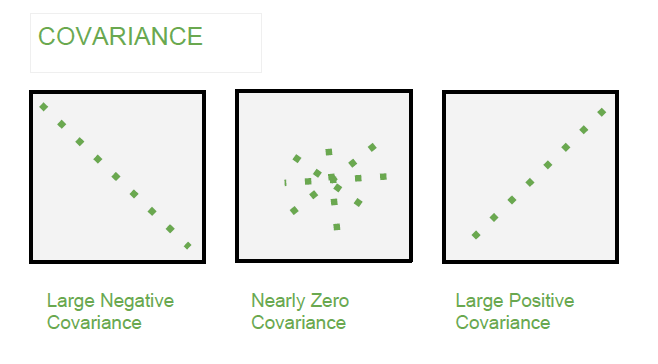


**For Sample:**



Here, x’ and y’ = mean of given sample set n = total no of sample xi and yi = individual sample of set

**Example –**



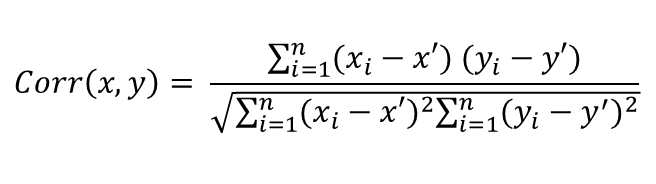
### Correlation

**Correlation** is a **standardized measure of the strength and direction of the linear relationship between two variables**. It is derived from covariance and **ranges between -1 and 1**. Unlike covariance, which only indicates the direction of the relationship, correlation provides a standardized measure.

* **Positive Correlation (close to +1)**: As one variable increases, the other variable also tends to increase.
* **Negative Correlation (close to -1)**: As one variable increases, the other variable tends to decrease.
* **Zero Correlation**: There is no linear relationship between the variables.

#### Correlation Coefficient

The Pearson correlation coefficient is the most often used metric of correlation. It expresses the linear relationship between two variables in numerical terms. The Pearson correlation coefficient, written as “r” or Corr(x, y) is as follows:



Here, x’ and y’ = mean of given sample set n = total no of sample xi and yi = individual sample of set.

**Interpretation of Correlation coefficients**

Perfect: 0.80 to 1.00

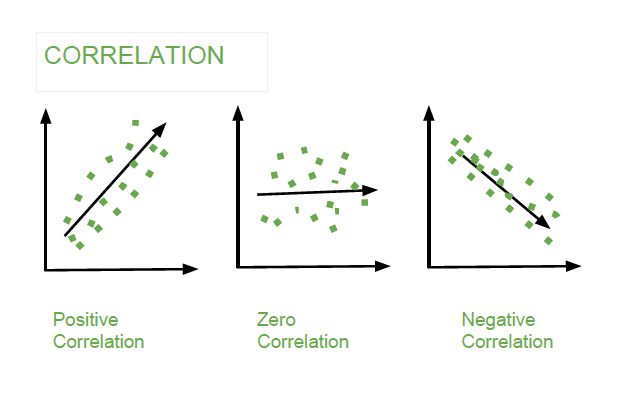
Strong: 0.50 to 0.79

Moderate: 0.30 to 0.49

Weak: 0.00 to 0.29

Value greater than 0.7 is considered a strong correlation between variables.

**Example –**



#### Different Correlation Coefficients

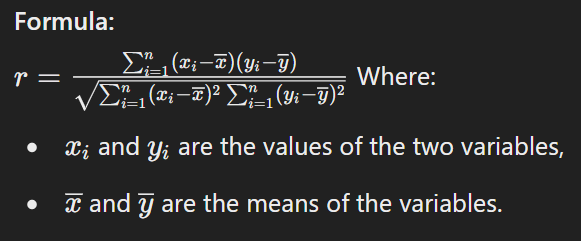
The different types of correlation coefficients used to measure the relation between two variables are:

**1. Pearson Correlation Coefficient (r)**

The **Pearson Correlation Coefficient** measures the linear relationship between two continuous variables. It assesses how much one variable changes in response to changes in another variable. It ranges from -1 to 1, where:

* **+1** indicates a perfect positive linear relationship.
* **-1** indicates a perfect negative linear relationship.
* **0** means no linear correlation.

**Formula:**



It assumes that the data are normally distributed and the relationship between the variables is linear.

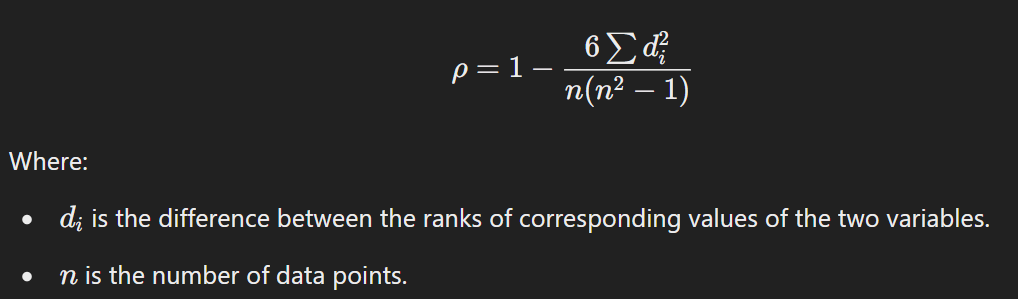
**Key Properties:**

* Sensitive to outliers.
* Cannot detect non-linear relationships.

**2. Spearman Rank Correlation Coefficient (ρ or Spearman’s rho)**

The **Spearman Rank Correlation Coefficient** measures the strength and direction of the **monotonic** relationship between two ranked variables. It is a **non-parametric** alternative to Pearson's correlation and is used when the relationship is not necessarily linear.

**Formula:**



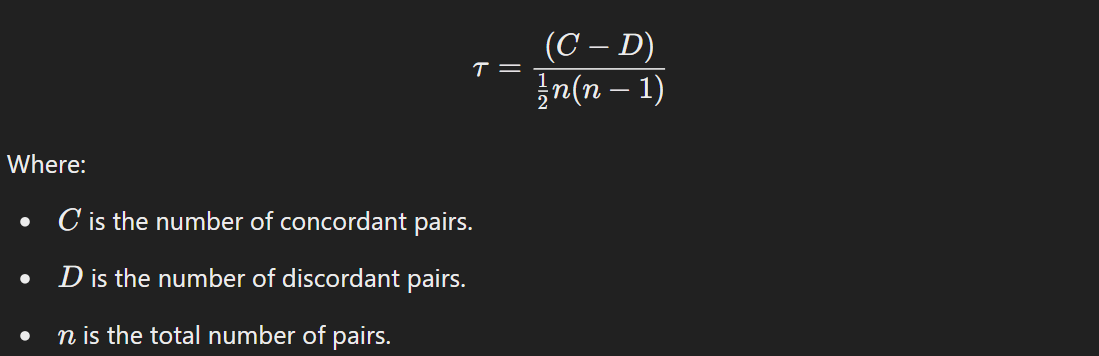
**Key Properties:**

* Can handle both continuous and ordinal variables.
* Can be used when the assumption of normality is not met.
* More robust to outliers compared to Pearson.

**3. Kendall Tau Coefficient (τ)**

The **Kendall Tau Coefficient** is a non-parametric statistic used to measure the association between two ranked variables. It assesses the similarity of the orderings of the data when ranked by each of the quantities.

**Formula:**

****

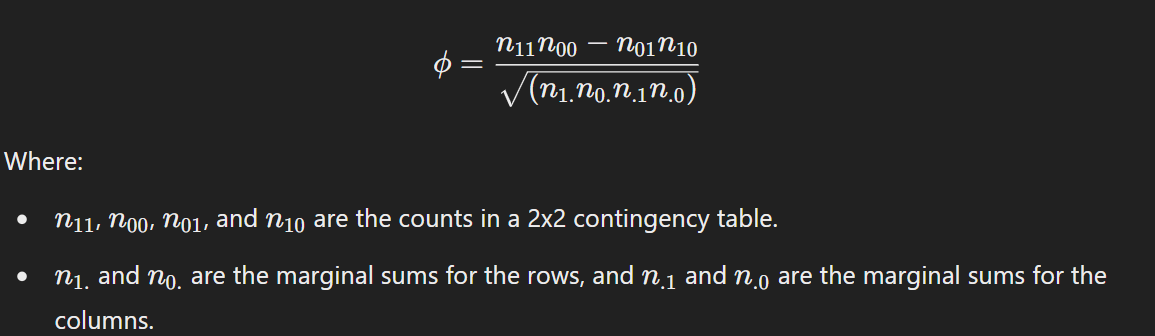
**Key Properties:**

* Focuses on the ordinal relationship (i.e., ranks).
* Values range from -1 (perfect negative correlation) to +1 (perfect positive correlation), where 0 means no correlation.
* More appropriate for small datasets and data with ties.

**4. Phi Coefficient (φ)**

The **Phi Coefficient** measures the association between two **binary** variables (also known as dichotomous variables). It is essentially the Pearson correlation for binary data and is commonly used in 2x2 contingency tables.

**Formula:**



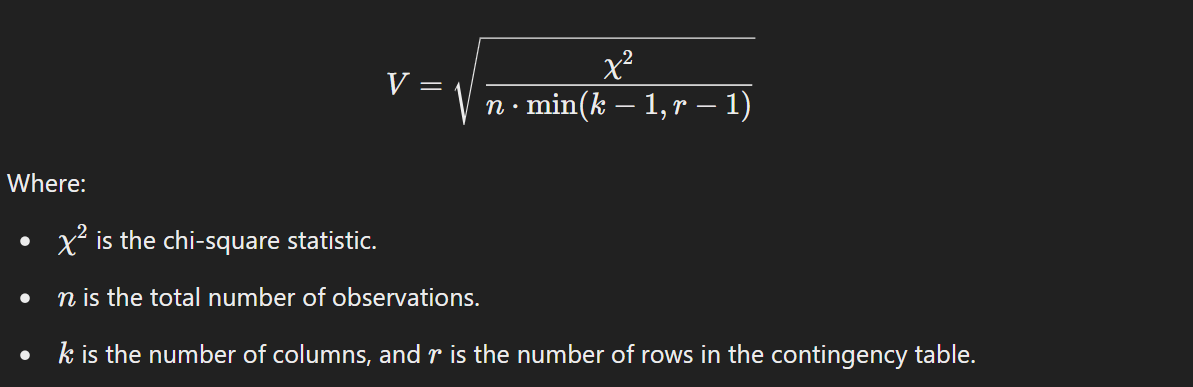
**Key Properties:**

* Ranges from -1 to +1, where 0 indicates no association.
* Used in categorical data analysis.

**5. Cramér’s V**

**Cramér's V** is an extension of the Phi coefficient for larger contingency tables. It measures the strength of association between two nominal variables. It is based on chi-square statistics and adjusts for the size of the table.

**Formula:**

****

**Key Properties:**

* Ranges from 0 to 1, where 0 indicates no association and 1 indicates perfect association.
* Used for tables larger than 2x2.
* Values near 1 suggest a strong association, but they do not imply causality.

**Summary of Use Cases:**

* **Pearson**: Linear relationships between continuous variables.
* **Spearman**: Non-parametric data or when data has a monotonic relationship.
* **Kendall**: Ranked data, especially with small sample sizes or tied ranks.
* **Phi**: Binary categorical data (2x2 contingency tables).
* **Cramér's V**: Nominal data in larger contingency tables.

### Difference between Covariance and Correlation

|  |  |
| --- | --- |
| **Covariance** | **Correlation** |
| Covariance is a measure of how much two random variables vary together | Correlation is a statistical measure that indicates how strongly two variables are related. |
| Involves the relationship between two variables or data sets | Involves the relationship between multiple variables as well |
| Lie between -infinity and +infinity | Lie between -1 and +1 |
| Measure of correlation | Scaled version of covariance |
| Provides direction of relationship | Provides direction and strength of relationship |
| Dependent on scale of variable | Independent on scale of variable |
| Have dimensions | Dimensionless |

### Applications of Covariance and Correlation

#### Applications of Covariance

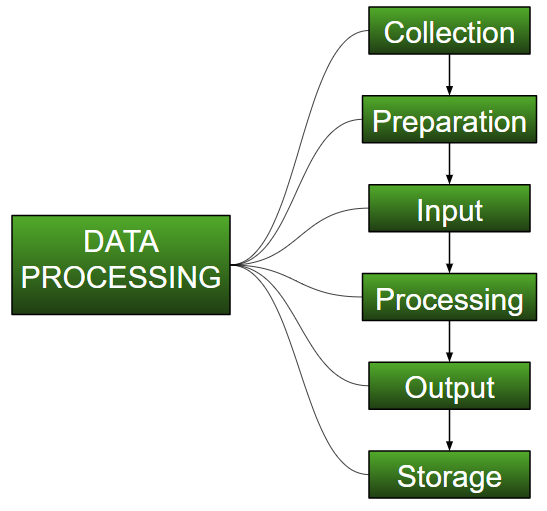
* **Portfolio Management in Finance**: Covariance is used to measure how different stocks or financial assets move together, aiding in portfolio diversification to minimize risk.
* **Genetics**: In genetics, covariance can help understand the relationship between different genetic traits and how they vary together.
* **Econometrics**: Covariance is employed to study the relationship between different economic indicators, such as the relationship between GDP growth and inflation rates.
* **Signal Processing**: Covariance is used to analyse and filter signals in various forms, including audio and image signals.
* **Environmental Science**: Covariance is applied to study relationships between environmental variables, such as temperature and humidity changes over time.

#### Applications of Correlation

* **Market Research**: Correlation is used to identify relationships between consumer behaviour and sales trends, helping businesses make informed marketing decisions.
* **Medical Research**: Correlation helps in understanding the relationship between different health indicators, such as the correlation between blood pressure and cholesterol levels.
* **Weather Forecasting**: Correlation is used to analyse the relationship between various meteorological variables, such as temperature and humidity, to improve weather predictions.
* **Machine Learning**: Correlation analysis is used in feature selection to identify which variables have strong relationships with the target variable, improving model accuracy.

# Data and It’s Processing ([Link](https://www.geeksforgeeks.org/ml-understanding-data-processing/))

Data Processing is the task of converting data from a given form to a much more usable and desired form i.e. making it more meaningful and informative.



## Data Type

Big Data includes huge volume, high velocity, and extensible variety of data. There are 3 types: Structured data, Semi-structured data, and Unstructured data.

1. **Structured data**

[Structured data](https://www.geeksforgeeks.org/what-is-structured-data/) is data whose elements are addressable for effective analysis. It has been organized into a formatted repository that is typically a database. It concerns all data which can be stored in database [SQL](https://www.geeksforgeeks.org/sql-tutorial/) in a table with rows and columns. They have relational keys and can easily be mapped into pre-designed fields. Today, those data are most processed in the development and simplest way to manage information.

*Example:* Relational data, Spreadsheets such as Excel, OLTP Systems, Online forms, Sensors such as GPS or RFID tags, Network and Web server logs, medical devices.

1. **Semi-Structured data**

[Semi-structured data](https://www.geeksforgeeks.org/what-is-semi-structured-data/) is information that does not reside in a relational database but that has some organizational properties that make it easier to analyse. With some processes, you can store them in the relation database (it could be very hard for some kind of semi-structured data), but Semi-structured exist to ease space.

*Example*: E-mails, XML and other markup languages, TCP/IP packets, Zipped files, Web pages

1. **Unstructured data**

[Unstructured data](https://www.geeksforgeeks.org/what-is-unstructured-data/) is a data which is not organized in a predefined manner or does not have a predefined data model; thus, it is not a good fit for a mainstream relational database. So, for Unstructured data, there are alternative platforms for storing and managing, it is increasingly prevalent in IT systems and is used by organizations in a variety of business intelligence and analytics applications. *Example*: Word, PDF, Text, Media logs.

## Data Cleaning vs Data Preprocessing

* **Data Cleaning** is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data. It focuses on ensuring the data is accurate and usable.
* **Data Preprocessing** is a broader concept that involves transforming raw data into an understandable format. It encompasses data cleaning, but also involves other steps like data transformation, scaling, encoding, and feature extraction.

## Data Cleaning ([Link](https://www.geeksforgeeks.org/data-cleansing-introduction/))

* 1. Handling Missing Values:
* Imputation (mean, median, or mode replacement)

|  |  |  |
| --- | --- | --- |
| **Method** | **Best For** | **When to Use** |
| **Mean** | Numerical (Continuous) | Normal distribution, no outliers |
| **Median** | Numerical (Continuous) | Skewed distribution, presence of outliers |
| **Mode** | Categorical/Ordinal/Discrete | Categorical data, most frequent value |

* + - Dropping rows or columns with missing values
    - Filling missing values with forward/backward fill techniques
  1. Handling Duplicates:
     + Identifying and removing duplicate records that can skew results.
  2. Correcting Inaccuracies:
     + Identifying incorrect or inconsistent data entries (e.g., typos, mis formatted dates).
  3. Handling Outliers:
     + Detecting outliers that could distort results and deciding how to treat them (e.g., capping, removing).
  4. Standardizing Data Formats:
     + Ensuring consistent formats for dates, numbers, categories, etc.
  5. Handling Noise:
     + Removing irrelevant data (e.g., extra spaces, unwanted characters).

## Data Preprocessing steps ([Link](https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/))

* 1. Data Cleaning: As described above, this is the first step of preprocessing.
  2. Data Transformation:
     + Check [Correlation](https://www.geeksforgeeks.org/exploring-correlation-in-python/) and Outcomes (Y labels) Proportionality
     + Normalization/Standardization: Scaling data so that it fits a particular range or distribution (min-max scaling, z-score normalization).
     + Encoding: Converting categorical variables into numeric form (e.g., one-hot encoding, label encoding).
  3. Feature Engineering:
     + Creating new features that might improve the model (e.g., combining or decomposing existing features).
  4. Dimensionality Reduction:
     + Reducing the number of features to improve performance (e.g., Principal Component Analysis - PCA).
  5. Handling Imbalanced Data:
     + Adjusting class distributions using techniques like oversampling, under sampling, or synthetic methods (e.g., SMOTE).
  6. Splitting Data:
     + Dividing the data into training, validation, and test sets to evaluate model performance.

## Exploratory Data Analysis (EDA)

EDA is the process of reviewing data to discover the main patterns in a data set. Data analysts can then leverage these data-driven insights to understand relationships between variables, pinpoint anomalies, verify hypotheses and complete other tasks.

Exploratory data analysis often involves developing data visualizations like scatter plots, histograms and box plots to spot trends.

### Type of EDA - Univariate, Bivariate and Multivariate data analysis

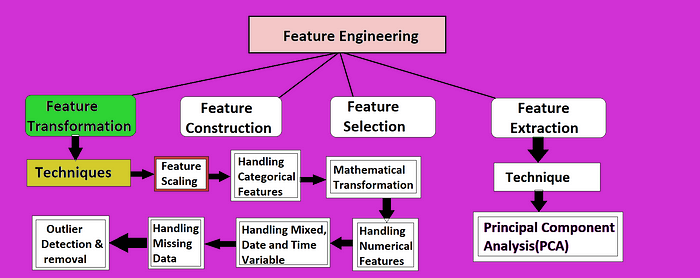
Univariate analysis looks at one variable, Bivariate analysis looks at two variables and their relationship. Multivariate analysis looks at more than two variables and their relationship. Aside from the techniques mentioned above, there are numerous software and packages that help us data analysts and data scientists uncover the story of the data. Understanding which analysis to use and when helps us navigate through the data.

**4 Packages to Automate Your Exploratory Data Analysis in Python:**

1. DataPrep
2. Pandas Profiling
3. SweetViz
4. AutoViz

## Feature Engineering

Feature engineering is the process of extracting meaningful features from raw data.



1. **Feature Transformation**

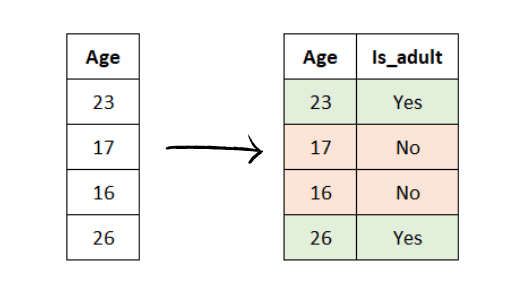
Feature transformation is the process of modifying features to make them more suitable for machine learning algorithms.

This includes handling missing values, converting categorical features to numerical values, detecting outliers, and scaling features to a common range.

1. **Feature Construction**

The process of developing new features from existing features or upon our domain knowledge is known as feature construction.

By making the features more informative and relevant to the task at hand, this helps machine learning models perform better.



There are numerous ways to build features, but some typical techniques include:

* Repurposing existing features
* Using domain expertise
* Using feature selection algorithms

## Feature Scaling Technique ([Link](https://www.geeksforgeeks.org/ml-feature-scaling-part-2/))([Normalization or Standardization](https://www.geeksforgeeks.org/standardscaler-minmaxscaler-and-robustscaler-techniques-ml/))

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units.

### Types of Feature Scaling:

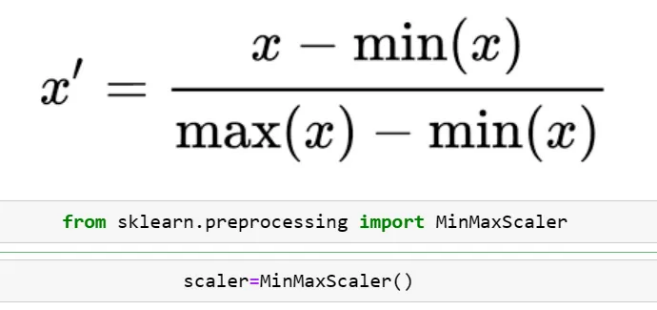
* **Standardization:**
  + Standard Scaler (Centered around 0, Mean = 0, SD = 1)
* **Normalization:**
  + Min Max Scaling [0, 1]
  + Mean Normalization [-1, 1]
  + Max Absolute Scaling [0, 1] or [-1, 1]
  + Robust Scaling (Varies, typically preserves the interquartile range)

### Normalization or Min-Max Scaling

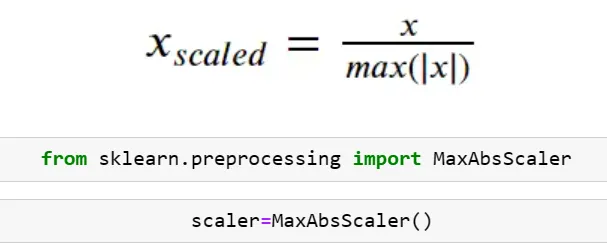
It is a scaling technique method in which data points are shifted and rescaled so that they end up in a range of 0 to 1. It is also known as **min-max scaling**.

Type of normalization:

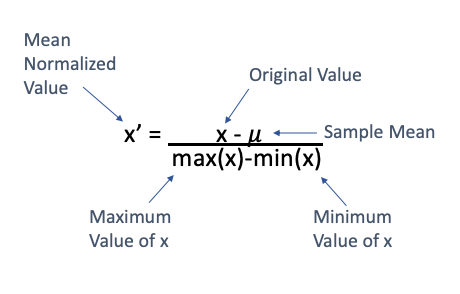
* MinMax Scaling (mostly used)



* MaxAbScaling

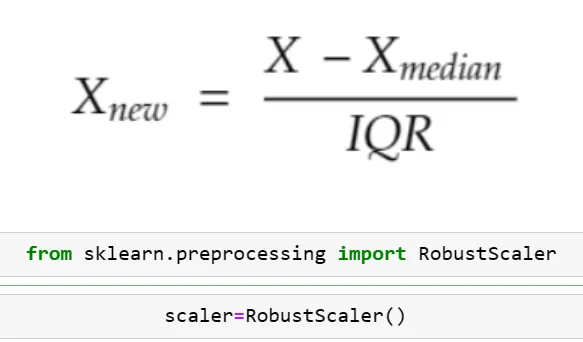


* Mean Normalization



Scikitlearn does not have any specific class for mean normalization. However, you can do this very easily using numpy.

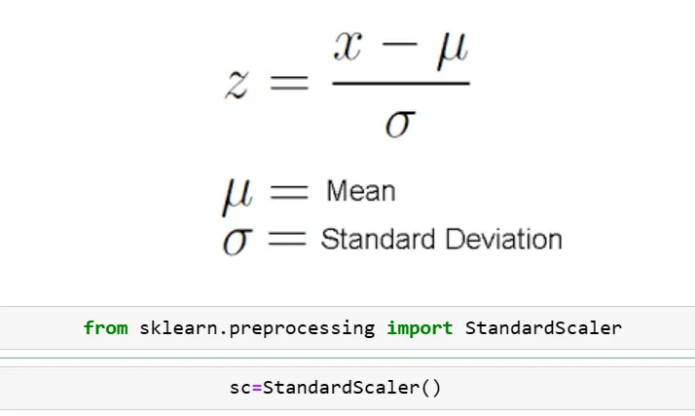
* Robust Scaling



### Standardization or Z-Score Normalization

Standardization is another scaling method where the values are centred around mean with a unit standard deviation. It means if we will calculate mean and standard deviation of standard scores it will be 0 and 1 respectively.

The formula for standardized values:

******

This Z is called standard score and it represents the number of standard deviations above or below the mean that a specific observation falls.

**Normalization** is preferred over standardization when our data doesn’t follow a normal distribution. It can be useful in those machine learning algorithms that do not assume any distribution of data like the k-nearest neighbour and neural networks.

**Standardization** is good to use when our data follows a normal distribution. It can be used in a machine learning algorithm where we make assumptions about the distribution of data like linear regression etc.

***In nutshell, Standardization is not suitable for data that have outliers. It adjusts the mean to 0. Robust scaler is more suitable for data that have outliers. Min-max scaler is not suitable for data that have outliers. It adjusts the data between 0 and 1. Max-absolute scaler is not suitable for data that have outliers. It adjusts the data between -1 and 1.***

1. If you do not know which scaler to use, apply all and check the effect on the models.

2. If you do not understand the data, use standard scaler. It works most of the times.

3. If you know the max and min values of the feature, then use min max scaler. Like in CNN.

4. If most of the values in the feature column is 0 or sparce matrix, then use Max Absolute Scaling.

5. If the data has outliers, use Robust Scaling.

|  |  |  |
| --- | --- | --- |
| S.NO. | Normalization | Standardization |
| 1. | Minimum and maximum value of features are used for scaling | Mean and standard deviation is used for scaling. |
| 2. | It is used when features are of different scales. | It is used when we want to ensure zero mean and unit standard deviation. |
| 3. | Scales values between [0, 1] or [-1, 1]. | It is not bounded to a certain range. |
| 4. | It is really affected by outliers. | It is much less affected by outliers. |
| 5. | Scikit-Learn provides a transformer called MinMaxScaler for Normalization. | Scikit-Learn provides a transformer called StandardScaler for standardization. |

## Encoding Technique

These encoding methods are used to convert **categorical variables** into a format that can be understood by machine learning algorithms.

Here’s a detailed overview of the most commonly used **encoding techniques** for categorical data, including their definitions, use cases, advantages, and disadvantages.

### Label Encoding

**Definition:**

Label encoding assigns a unique integer to each category in a categorical variable.

**Use Cases:**

* When the categorical data is **ordinal** (has an inherent order) or **binary** (two categories).
* Often used to encode the **target variable (Y Value)** in classification problems.

**Advantages:**

* Simple and fast to implement.
* Efficient for binary categorical features or ordinal data.

**Disadvantages:**

* Can **mislead models** if applied to **nominal data**, as it introduces an arbitrary ranking between categories.



### One-Hot Encoding

**Definition:**

One-hot encoding converts each unique category into a new binary column, where each column corresponds to one category. Each observation will have a value of 1 in the column representing its category and 0 in all other columns.

E.g. Male and Female mapped to 0 and 1. But this can add bias in our model as it will start giving higher preference to the Female parameter as 1>0 but ideally, both labels are equally important in the dataset. To deal with this issue we will use the One Hot Encoding technique.

**Use Cases:**

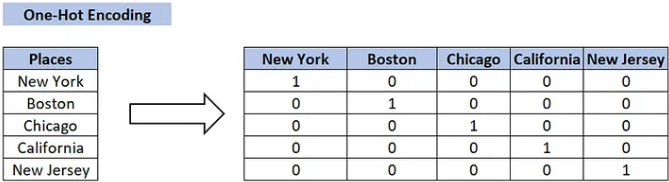
* **Nominal data** where categories have no order (e.g., colours, countries).
* Works well for categorical data with **small numbers of unique categories**.

**Advantages:**

* Prevents the model from assuming any ordinal relationship between categories.
* Ensures that each category is treated independently.

**Disadvantages:**

* **Increases dimensionality**: The number of columns grows with the number of unique categories, leading to sparse data.
* Can be inefficient for **high cardinality** features.



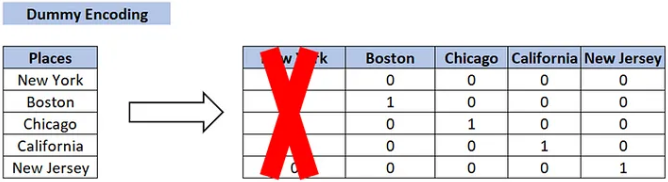
### Dummy Encoding

• Dummy coding scheme is **similar to one-hot encoding**.

• This categorical data encoding method transforms the categorical variable into a set of binary variables [0/1].

• In the case of **one-hot encoding**, for N categories in a variable, it uses N binary variables.

• The dummy encoding is a small improvement over one-hot-encoding. Dummy encoding uses N-1 features to represent N labels/categories.



### Ordinal Encoding

**Definition:**

Ordinal encoding is similar to label encoding but allows you to explicitly define the mapping between categories and integer labels. This is especially useful when there is a clear and predefined ordinal relationship. You manually specify the order of categories and map them to integers accordingly.

**Use Cases:**

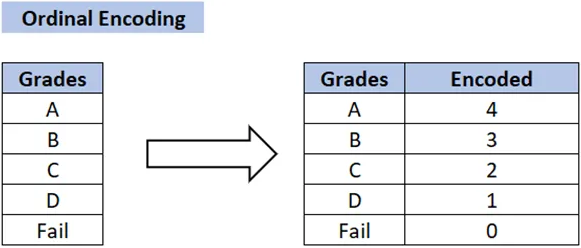
* For **ordinal categorical data** where the categories have a clear ranking (e.g., "Low", "Medium", "High").

**Advantages:**

* Captures the inherent order in the data.
* Simple to implement and results in low-dimensional encoding.

**Disadvantages:**

* **Inappropriate for nominal data**, as it imposes a false sense of order where none exists.



### Binary Encoding

**Definition:**

Binary Encoding is similar to One-Hot Encoding, but instead of creating a separate column for each category, the categories are represented as binary digits.

**Use Cases:**

* For **high cardinality** categorical features (many unique categories).
* Used when one-hot encoding results in too many columns.

**Advantages:**

* **Reduces dimensionality** compared to one-hot encoding.
* Efficient for high cardinality data (e.g., product IDs, customer IDs).

**Disadvantages:**

* Less interpretable than one-hot encoding.
* Can still introduce some ordinal relationships between categories, depending on how the binary encoding is interpreted by the model.



### Frequency Encoding (Count Encoding)

**Definition:**

Frequency encoding replaces each category with the **frequency of its occurrence** in the dataset. Categories with the highest occurrences are assigned the highest frequency values.

**Use Cases:**

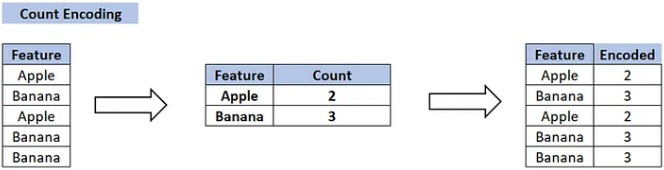
* Useful when categories have varying frequencies that may carry useful information.
* For **nominal data**, especially when there is a meaningful difference between how often categories occur.

**Advantages:**

* Captures category importance based on frequency.
* Reduces dimensionality and is more efficient than one-hot encoding.

**Disadvantages:**

* Assumes that higher frequency categories are more important, which may not always be the case.



### Target Encoding (Mean Encoding)

**Definition:**

Target encoding replaces categories with the **mean of the target variable** for each category. For example, if a category corresponds to a higher average sales value, it would get a higher encoded value.

**Use Cases:**

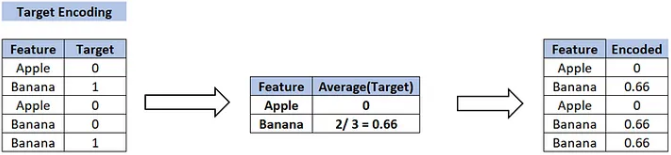
* Used in **supervised learning** problems where the target variable is available during training.
* For both **binary and categorical target variables**.

**Advantages:**

* Leverages the relationship between the category and the target variable, making it more informative than one-hot encoding.

**Disadvantages:**

* **Risk of overfitting**: It directly uses the target variable, which can lead to data leakage if not handled properly (especially with small datasets).



### Leave-One-Out Encoding

**Definition:**

Leave-one-out encoding is a variation of target encoding. For each category, the mean of the target variable is calculated, **excluding the current row**. This helps to reduce data leakage and overfitting compared to regular target encoding.

**Use Cases:**

* For **supervised learning** when using categorical variables.
* When target encoding is needed, but you want to reduce overfitting.

**Advantages:**

* Reduces data leakage compared to regular target encoding.
* Preserves useful information related to the target variable.

**Disadvantages:**

* More computationally expensive than target encoding.
* Still carries the risk of overfitting if not handled carefully.

## Dummy variable trap in Regression Models

Dummy Variable in Regression Models:

In statistics, especially in regression models, we deal with various kinds of data. The data may be quantitative (numerical) or qualitative (categorical). The numerical data can be easily handled in regression models but we can’t use categorical data directly, it needs to be transformed in some way.

For transforming categorical attributes to numerical attributes, we can use the label encoding procedure (label encoding assigns a unique integer to each category of data). But this procedure is not alone that suitable, hence, ***one hot encoding*** is used in regression models following label encoding. This enables us to create new attributes according to the number of classes present in the categorical attribute i.e. if there are n number of categories in categorical attribute, n new attributes will be created. These attributes created are called ***Dummy Variables***. Hence, dummy variables are “proxy” variables for categorical data in regression models.

These dummy variables will be created with one-hot encoding and each attribute will have a value of either 0 or 1, representing the presence or absence of that attribute.

**Dummy Variable Trap:**

The Dummy variable trap is a scenario where there are attributes that are highly correlated (Multicollinear) and one variable predicts the value of others. When we use one-hot encoding for handling the categorical data, then one dummy variable (attribute) can be predicted with the help of other dummy variables. Hence, one dummy variable is highly correlated with other dummy variables. Using all dummy variables for regression models leads to a dummy variable trap. So, the regression models should be designed to exclude one dummy variable.

**For Example –**

Let’s consider the case of gender having two values male (0 or 1) and female (1 or 0). Including both the dummy variable can cause redundancy because if a person is not male in such case that person is a female, hence, we don’t need to use both the variables in regression models. This will protect us from the dummy variable trap.

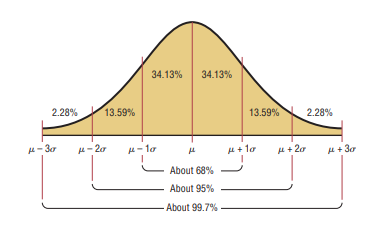
## Data Transformation

Data Transformations are like backstage artists preparing the actors (features) for the grand performance (model training). They are special tools that apply a user-defined function to each column of the dataset, performing custom transformations on the data. This process ensures that the data is not just fed to the model but is crafted and tailored to enhance its predictive prowess.

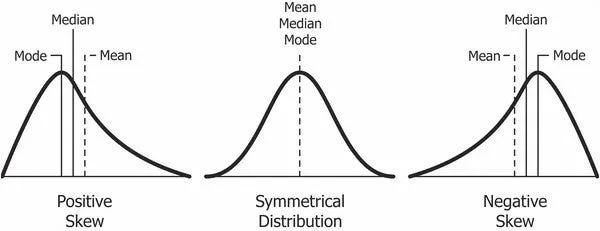
In short, it is a method that we use to transform the data by using some mathematical or custom functions so that your model will get better data for training.

### Why do we need to use transformation on our data?

The normal distribution, also known as the Gaussian distribution, is a continuous probability distribution that is widely used in Machine Learning and statistical modeling. It is a bell-shaped curve that is symmetrical around its mean and is characterized by its mean and standard deviation.



The above image says the data is distributed normally and sometimes data can be skewed to either left to right also. In a right-skewed data scenario, more data points fall to the right side and in a left-skewed, data points fall towards the left.



Maybe this skewed data is not good for model training, why? Machines learn from patterns, and extreme values can mess up their understanding. We want to make sure the model sees the full story, not just a part of it. So, we often tweak the data to make it more balanced and fairer, helping the machine learn better. So, these transformations will help you achieve balanced data.

### Data Transformation Techniques:

1. **Mathematical Transformers**

Mathematical transformers apply mathematical functions to transform data, often to normalize or stabilize variance, improve the distribution, or make the data more suitable for analysis. Here are some common **mathematical transformations**:

1. Function Transformers (Check by hit & trial method)
   * Log Transform (For right skewed data)
   * Reciprocal Transform (For left skewed data)
   * Square Transform (You can try)
   * Sq Root Transform (You can try)
   * Custom Transform (You can try)
2. Power Transformers (Most Used)
   * Box-Cox (Work good for all type of skewed but cannot be used for negative values and 0)
   * Yeo-Johnson (Work good for all type of skewed)
3. Quantile Transformer
4. **Binning (Discretization) and Binarization**
5. Binning / Discretization

Binning is the process of grouping a set of continuous or numerical data points into a smaller number of discrete “bins” for analysis.

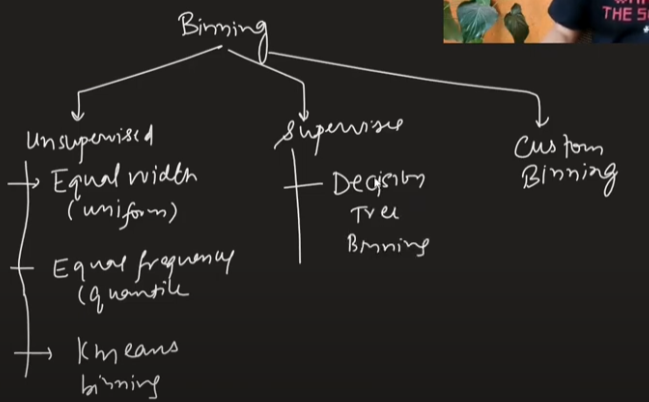
What are Bins? Bins are intervals or ranges into which you divide the range of your continuous numerical data.

**Why do we create Bins?**

Simplification: Binning simplifies the data by converting a range of values into a smaller set of discrete categories, making it easier to understand and interpret.

Handling Non-Linearity: Some machine learning algorithms may assume linear relationships, and binning can help capture non-linear patterns.

Dealing with Outliers: Binning can also be useful for handling outliers by placing extreme values into specific bins.



1. Binarization:

Binarization is the process of converting numerical data into binary form, typically 0s and 1s. It involves setting a threshold value, and any data point above the threshold is marked as 1, while those below or equal to the threshold are marked as 0.

## [Outliers](https://www.geeksforgeeks.org/detect-and-remove-the-outliers-using-python/)

An Outlier is a data item/object that deviates significantly from the rest of the (so-called normal) objects. Identifying outliers is important in statistics and data analysis because they can have a significant impact on the results of statistical analyses. The analysis for outlier detection is referred to as outlier mining.

### Outlier detection

1. **Outlier detection Using Box Plot**
2. **Outlier detection Using Scatterplot**
3. **Outlier detection with Z-Score Method (More Robust Approach)**

[**Z- Score**](https://www.geeksforgeeks.org/z-score-in-statistics/)**,**also known as the **standard score**,**tells us the deviation of a data point from the mean** by expressing it in terms of standard deviations above or below the mean. It gives us an idea of how far a data point is from the mean. Hence, the Z-Score is measured in terms of standard deviation from the mean.

***z = (X – μ) / σ***

***mean (μ) and standard deviation (σ).***

* **Common Z-Score Thresholds for Outlier Detection:**

Z > 3 or Z < -3: Any data point with a Z-score greater than 3 or less than -3 is typically considered an outlier.

Z > 2 or Z < -2: A less strict threshold, where points outside 2 standard deviations are sometimes considered outliers in some practical applications.

* **When NOT to Use the Z-Score Method:**

**Non-normal distribution:** If your data is highly skewed, with heavy tails or is not normally distributed, other methods like the IQR (Interquartile Range) or robust statistical methods might be better for outlier detection.

**Small sample size:** The Z-score method might not be reliable if the dataset is small or has many naturally occurring fluctuations.

**Presence of significant skewness or heteroscedasticity:** These issues can affect the calculation of the mean and standard deviation, making Z-scores less useful for outlier detection.

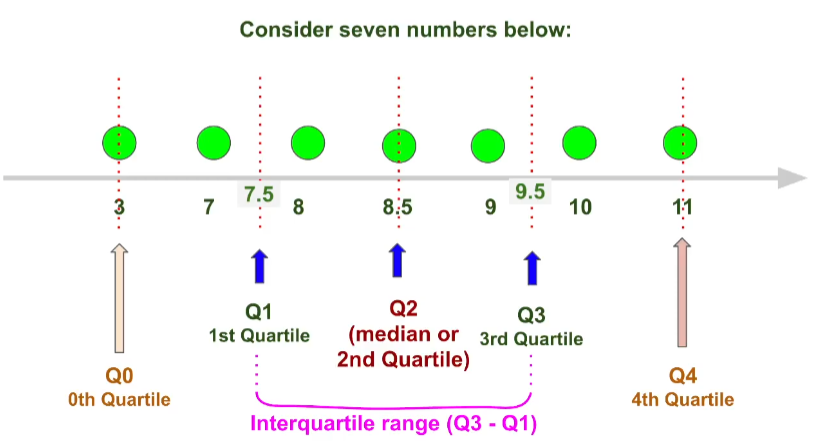
1. **Outlier detection using IQR (More Robust Approach)**

[**IQR (Inter Quartile Range)**](https://www.geeksforgeeks.org/interquartile-range-iqr/)approach to finding the outliers is the most commonly used and most trusted approach used in the research field.

IQR in [Statistics](https://www.geeksforgeeks.org/introduction-of-statistics-and-its-types/) is used to measure variability by dividing a data set into quartiles. The data is sorted in ascending order and split into 4 equal parts. Q1, Q2, Q3 called first, second and third quartiles in the given data.

* Q1 represents the 25th percentile of the data.
* Q2 represents the 50th percentile of the data.
* Q3 represents the 75th percentile of the data.

*IQR = Quartile3 – Quartile1*

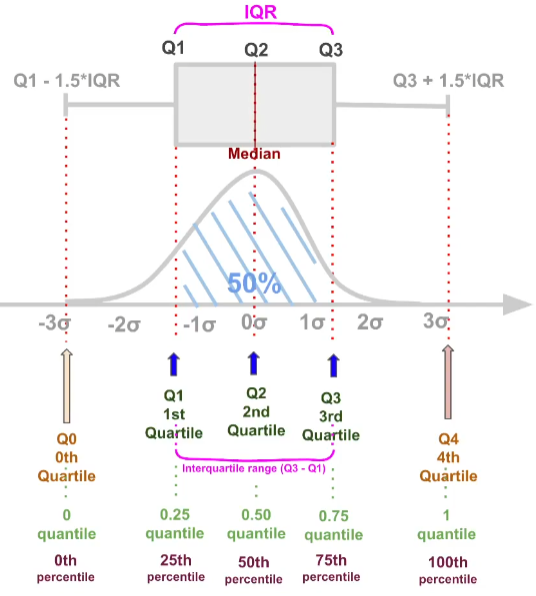
**

***Syntax:*** *Quartile = numpy.percentile(arr, n, axis=None, out=None)****Parameters****:*

* *arr: input array.*
* *n: percentile value.*

upper = Q3 +1.5\*IQR

lower = Q1 – 1.5\*IQR



The **IQR proximity method** is best used when you have **non-normal or skewed univariate data** and need a **robust** and **simple** technique to detect outliers. It is effective when outliers are expected to lie outside a reasonable range of the data but less suited for high-dimensional or correlated datasets.

### Outlier Removal

* 1. Removing Outlier
  2. Copping Outlier – Replacing outliers with max (If it lies on max side) and min (If it lies on min side) limit.

### Outlier vs Anomaly

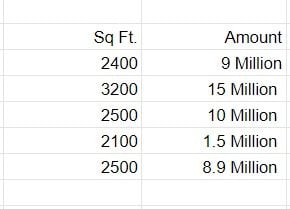
**Outlier** is when some data points are behaving differently but doesn’t contribute to any *bad* thing. For example, employees making too high or too low salary.

**Anomaly** is when something unusual (or unexpected) is experienced by the monitoring system that is alarming. For example, a user who does 500 - 600 RS purchase daily from card, has done 50000 RS purchase.

## Feature Construction and Feature Splitting

***Feature construction*** is the process of creating new features or variables from existing data that can be used to improve the performance of machine learning models. Features are the input variables used by machine learning algorithms to make predictions, and feature construction involves transforming the raw data into a more meaningful representation that captures the underlying patterns and relationships in the data.

To better understand it, let’s look at a simple example. Below are the prices of properties in x city. It shows the area of the house and total price.



To begin, we’ll add a new column to display the cost per square foot.



***Feature splitting*** is a technique in machine learning that involves breaking down a single feature into multiple features. The purpose of this technique is to generate more informative features that can provide greater insight into the relationships between input variables and the target variable. By splitting features, continuous variables can be converted into categorical variables, information can be extracted from date and time features, and text features can be separated into smaller units.

## Dimensionality Reduction

To address the [curse of dimensionality](https://www.geeksforgeeks.org/videos/curse-of-dimensionality-in-machine-learning/), [Feature engineering](https://www.geeksforgeeks.org/what-is-feature-engineering/)techniques are used which include feature selection and feature extraction.

Dimensionality reduction is a process and technique to reduce the number of dimensions -- or features -- in a data set. The goal of dimensionality reduction is to decrease the data set's complexity by reducing the number of features while keeping the most important properties of the original data.

Dimensionality reduction is advantageous to AI developers or data professionals working with massive data sets, performing data visualization and analyzing complex data. It aids in the process of data compression, allowing the data to take up less storage space as well as reducing computation times. The technique is commonly used in machine learning (ML).

Dimensionality reduction is a particularly useful way to prevent overfitting and to solve classification and regression problems.

### Different Techniques for Dimensionality Reduction

There are two common dimensionality reduction techniques: feature selection and feature extraction.

* In **feature selection**, small subsets of the most relevant features are chosen from a larger set of dimensional data to represent a model by [filtering](https://www.techtarget.com/whatis/definition/filter), wrapping or embedding. The goal here is to reduce the data set's dimensionality while keeping its most important features.
* **Feature extraction** combines and transforms the data set's original features to create new features. The goal is to create a lower-dimensional data set that still has the data set's properties.

Feature selection uses different methods, such as the following:

* **The filter method.** Filters a data set into a subset that only has the most relevant features of the original data set.
* **The wrapper method.** Feeds features into an ML model to evaluate if a feature should be removed or added.
* **The embedded method.** Evaluates the performance of each feature by checking training iterations of the ML model.

Feature extraction uses methods such as the following:

* **Principal component analysis (PCA).** A statistical process that identifies smaller units of features from larger data sets. These small units are called principal components.
* **Linear discriminant analysis (LDA).** A method that finds features that separate different classes of data the best.
* **T-distributed stochastic neighbour embedding (t-SNE).** An unsupervised, nonlinear dimensionality reduction method that creates a probability distribution over pairs of objects and then creates a probability distribution over the points in a low-dimensional map.

Other methods used in dimensionality reduction include the following:

* Factor analysis.
* High [correlation](https://www.techtarget.com/whatis/definition/correlation) filter.
* UMAP.
* Random forest.

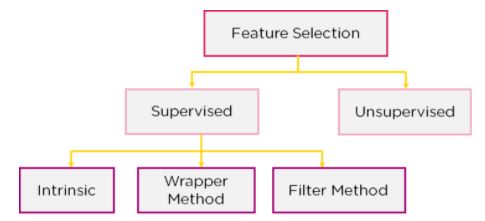
## Feature Selection

Feature Selection is the method of reducing the input variable to your model by using only relevant data and getting rid of noise in data.

It is the process of automatically choosing relevant features for your machine learning model based on the type of problem you are trying to solve. We do this by including or excluding important features without changing them. It helps in cutting down the noise in our data and reducing the size of our input data.

Feature selection models are of two types:

1. Supervised Models: Supervised feature selection refers to the method which uses the output label class for feature selection. They use the target variables to identify the variables which can increase the efficiency of the model
2. Unsupervised Models: Unsupervised feature selection refers to the method which does not need the output label class for feature selection. We use them for unlabelled data.



### Filter Method

In this method, features are dropped based on their relation to the output, or how they are correlating to the output. We use correlation to check if the features are positively or negatively correlated to the output labels and drop features accordingly.

****

**Some Filter Based Techniques**

* Drop Duplicate Features (Columns)

1. **Variance Threshold**
2. **Correlation Coefficient**
3. **ANOVA (Analysis of variance)**
4. **Chi-Square Test**
5. **Fisher’s Score**
6. **Information Gain**
7. **Manual Info**

**Advantages and Disadvantages**

**Advantages**

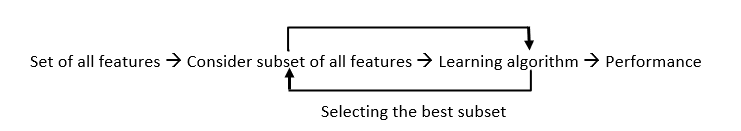
* Simplicity: Filter methods are generally straightforward and easy to understand. They involve calculating a statistic that measures the relevance of each feature, and selecting the top features based on this statistic.
* Speed: These methods are usually computationally efficient. Because they evaluate each feature independently, they can be much faster than wrapper methods or embedded methods, which need to train a model to evaluate feature importance.
* Scalability: Filter methods can handle a large number of features effectively because they don't involve any learning methods. This makes them suitable for high-dimensional datasets.
* Pre-processing Step: They can serve as a pre-processing step for other feature selection methods. For instance, you could use a filter method to remove irrelevant features before applying a more computationally expensive method, such as a wrapper method.

**Disadvantages**

* Lack of Feature Interaction: Filter methods treat each feature individually and hence do not consider the interactions between features. They might miss out on identifying important features that don't appear significant individually but are significant in combination with other features.
* Model Agnostic: Filter methods are agnostic to the machine learning model that will be used for the prediction. This means that the selected features might not necessarily contribute to the accuracy of the specific model you want to use.
* Statistical Measures Limitation: The statistical measures used in these methods have their own limitations. For example, correlation is a measure of linear relationship and might not capture non-linear relationships effectively. Similarly, variance-based methods might keep features with high variance but low predictive power.
* Threshold Determination: For some methods, determining the threshold to select features can be a bit subjective. For example, what constitutes "low" variance or "high" correlation might differ depending on the context or the specific dataset.

### Wrapper Method:

We split our data into subsets and train a model using this. Based on the output of the model, we add and subtract features and train the model again. It forms the subsets using a greedy approach and evaluates the accuracy of all the possible combinations of features. E.g.: Forward Selection, Backwards Elimination, etc.



**Some Wrapper Techniques**

1. **Exhaustive Feature Selection (Best Subset Selection)**
2. **Backward Feature Elimination**
3. **Forward Feature Selection**
4. **Recursive Feature Elimination**

**Advantages and Disadvantages**

**Advantages**

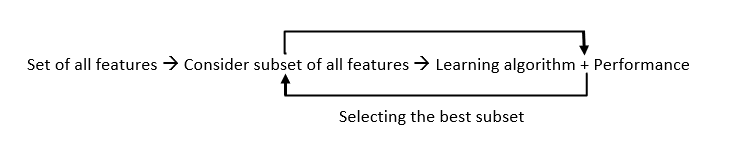
* Accuracy: Wrapper methods usually provide the best performing feature subset for a given machine learning algorithm because they use the predictive power of the algorithm itself for feature selection.
* Interaction of Features: They consider the interaction of features. While filter methods consider each feature independently, wrapper methods evaluate subsets of features together. This means that they can find groups of features that together improve the performance of the model, even if individually these features are not strong predictors.

**Disadvantages**

* Computational Complexity: The main downside of wrapper methods is their computational cost. As they work by generating and evaluating many different subsets of features, they can be very time-consuming, especially for datasets with a large number of features.
* Risk of Overfitting: Because wrapper methods optimize the feature subset to maximize the performance of a specific machine learning model, they might select a feature subset that performs well on the training data but not as well on unseen data, leading to overfitting.
* Model Specific: The selected feature subset is tailored to maximize the performance of the specific model used in the feature selection process. Therefore, this subset might not perform as well with a different type of model.

### Intrinsic Method (Embedded Methods)

This method combines the qualities of both the Filter and Wrapper method to create the best subset. This method takes care of the machine training iterative process while maintaining the computation cost to be minimum. E.g.: Lasso and Ridge Regression.



**Some Wrapper Techniques**

1. **Regularization**
2. **Random Forest Importance**

**Advantages and Disadvantages**

**Advantages**

* Performance: They are generally more accurate than filter methods since they take the interactions between features into account.
* Efficiency: They are more computationally efficient than wrapper methods since they fit the model only once.
* Less Prone to Overfitting: They introduce some form of regularization, which helps to avoid overfitting. For example, Lasso and Ridge regression add a penalty to the loss function, shrinking some coefficients to zero.

**Disadvantages**

* Model Specific: Since they are tied to a specific machine learning model, the selected features are not necessarily optimal for other models.
* Complexity: They can be more complex and harder to interpret than filter methods. For example, understanding why Lasso shrinks some coefficients to zero and not others can be non-trivial.
* Tuning Required: They often have hyperparameters that need to be tuned, like the regularization strength in Lasso and Ridge regression.
* Stability: Depending on the model and the data, small changes in the data can result in different sets of selected features. This is especially true for models that can fit complex decision boundaries, like decision trees.

## Feature extraction

Feature extraction is a process used in [machine learning](https://deepai.org/machine-learning-glossary-and-terms/machine-learning) to reduce the number of resources needed for processing without losing important or relevant information. Feature extraction helps in the reduction of the dimensionality of data which is needed to process the data effectively. In other words, feature extraction involves creating new features that still capture the essential information from the original data but in a more efficient way.

When dealing with large datasets, especially in domains like image processing, [natural language processing](https://deepai.org/machine-learning-glossary-and-terms/natural-language-processing), or signal processing, it's common to have data with numerous features, many of which may be irrelevant or redundant. Feature extraction allows for the simplification of the data which helps algorithms to run faster and more effectively.

**Why is Feature Extraction Important?**

Feature extraction is crucial for several reasons:

* **Reduction of Computational Cost:** By reducing the dimensionality of the data, machine learning algorithms can run more quickly. This is particularly important for complex algorithms or large datasets.
* **Improved Performance:** Algorithms often perform better with a reduced number of features. This is because noise and irrelevant details are removed, allowing the algorithm to focus on the most important aspects of the data.
* **Prevention of Overfitting:** With too many features, models can become overfitted to the training data, meaning they may not generalize well to new, unseen data. Feature extraction helps to prevent this by simplifying the model.
* **Better Understanding of Data:** Extracting and selecting important features can provide insights into the underlying processes that generated the data.

**Methods of Feature Extraction**

There are several methods of feature extraction, and the choice of method depends on the type of data and the desired outcome. Some common methods include:

* **[Principal Component Analysis](https://deepai.org/machine-learning-glossary-and-terms/principal-components-analysis) (PCA):** PCA is a statistical method that transforms the data into a new coordinate system, where the greatest [variance](https://deepai.org/machine-learning-glossary-and-terms/variance) by some projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.
* **Linear Discriminant Analysis (LDA):** LDA is used to find the linear combinations of features that best separate two or more classes of objects or events.
* **Autoencoders:** [Autoencoders](https://deepai.org/machine-learning-glossary-and-terms/autoencoder) are a type of [neural network](https://deepai.org/machine-learning-glossary-and-terms/neural-network) that is trained to attempt to copy its input to its output. During training, the network learns to represent the input as a compressed form, which can be used as features for another task.
* **t-Distributed Stochastic Neighbor Embedding (t-SNE):** t-SNE is a non-linear technique for dimensionality reduction that is particularly well suited for embedding [high-dimensional data](https://deepai.org/machine-learning-glossary-and-terms/high-dimensional-data) into a space of two or three dimensions, which can then be visualized in a scatter plot.
* [**Independent Component Analysis**](https://deepai.org/machine-learning-glossary-and-terms/independent-component-analysis)**(ICA):** ICA is a computational method for separating a multivariate signal into additive subcomponents that are maximally independent.
* **Feature Agglomeration:** This method involves merging similar features together to reduce the dimensionality of the data.

**Challenges in Feature Extraction**

While feature extraction can be highly beneficial, it also presents challenges:

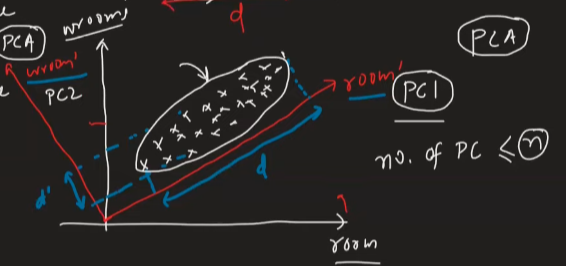
* **Choosing the Right Method:** There is no one-size-fits-all method for feature extraction. The choice of the right technique is crucial and often requires domain knowledge.
* **Loss of Information:** There is always a risk that important information may be lost during the feature extraction process.
* **Computational Complexity:** Some feature extraction methods can be computationally expensive, especially on large datasets.

### Principal Component Analysis (PCA)

Feature extraction helps in the reduction of the dimensionality of data which is needed to process the data effectively. In other words, feature extraction involves creating new features that still capture the essential information from the original data but in a more efficient way.

When dealing with large datasets, especially in domains like image processing, natural language processing, or signal processing, it's common to have data with numerous features, many of which may be irrelevant or redundant. Feature extraction allows for the simplification of the data which helps algorithms to run faster and more effectively.

This is achieved by keeping the principal components with the largest variance and ignoring the lower variance components, which are assumed to contain the noise of the dataset. PCA is sensitive to the relative scaling of the original variables; hence, data normalization is a crucial preprocessing step.



**How PCA Works** [**(Link)**](https://builtin.com/data-science/step-step-explanation-principal-component-analysis)

The steps to perform PCA include:

1. Standardizing the data: PCA is affected by scale, so the data needs to be normalized.
2. Calculating the covariance matrix: To understand how the variables of the input data are varying from the mean with respect to each other.
3. Computing the [eigenvectors](https://deepai.org/machine-learning-glossary-and-terms/eigenvector) and [eigenvalues](https://deepai.org/machine-learning-glossary-and-terms/eigenvalue) of the covariance matrix: To identify the principal components.
4. Choosing components and forming a feature vector: By ranking the eigenvalues in descending order and choosing the top k eigenvectors.
5. Deriving the new data set: This is done by multiplying the original data set by the feature vector.

**Understanding Eigenvalues and Eigenvectors**

In PCA, eigenvectors and eigenvalues of a covariance (or correlation) matrix represent the "core" of PCA: eigenvectors determine directions of the new feature space, and eigenvalues determine their magnitude. In other words, eigenvalues explain the variance of the data along the new feature axes.

Mathematically, for a given square matrix A, a vector v is an eigenvector of A if the following equation is satisfied:

Av = λv

Here, λ represents the eigenvalue associated with the eigenvector v. It's important to note that for a matrix to have eigenvectors and eigenvalues, it must be a square matrix, though not all square matrices have eigenvectors.

**Choosing the Number of Principal Components**

The number of principal components retained in the analysis is a critical decision. In practice, the choice is often made based on the cumulative explained variance, which should be as high as possible, while keeping fewer components. A common rule of thumb is to keep the principal components that explain at least 85% of the variance.

**Advantages and Disadvantages of PCA**

Advantages:

* Removal of multicollinearity: PCA helps in mitigating the problem of multicollinearity in the data by transforming the original variables into a new set of variables that are uncorrelated.
* Reduction of overfitting: By reducing the dimensionality, PCA can help reduce the chances of overfitting in a predictive model.
* Improvement in visualization: [High-dimensional data](https://deepai.org/machine-learning-glossary-and-terms/high-dimensional-data) can be difficult to visualize, but PCA can make this visualization easier by reducing the number of dimensions.

Disadvantages:

* Interpretability: The principal components are linear combinations of the original variables and may not be easily interpretable.
* Sensitivity to scaling: PCA is sensitive to the scaling of the variables, which means that the results can vary depending on how the data was scaled.
* Data loss: While reducing dimensionality, some information is inevitably lost, which might be important depending on the context.

## Handling Imbalanced Data with SMOTE and Near Miss Algorithm

In Machine Learning and Data Science we often come across a term called Imbalanced Data Distribution, generally, happens when observations in one of the classes are much higher or lower than the other classes. As Machine Learning algorithms tend to increase accuracy by reducing the error, they do not consider the class distribution. This problem is prevalent in examples such as Fraud Detection, Anomaly Detection, Facial recognition, etc.

E.g. Let’s assume that XYZ is a bank that issues credit cards to its customers. Now, the bank is concerned that some fraudulent transactions are going on, and when the bank checks their data, they found that for every 2000 transactions, there are only 30 Nos of fraud recorded. So, the fraud per 100 transactions is less than 2% or more than 98% of transactions is “No Fraud.” Here, the class “No Fraud” is called the majority class, and the much smaller “Fraud” class is called the minority class.

**Imbalanced**[**Data Handling**](https://www.geeksforgeeks.org/data-handling/)**Techniques:**

1. SMOTE (Synthetic Minority Oversampling Technique)
2. Near Miss Algorithm (Undersampling Technique)

### SMOTE (Synthetic Minority Oversampling Technique)

It aims to balance class distribution by randomly increasing minority class examples by replicating them. SMOTE synthesizes new minority instances between existing minority instances. It generates the **virtual training records by linear interpolation** for the minority class. These synthetic training records are **generated by randomly selecting one or more of the k-nearest neighbors** for each example in the minority class. After the oversampling process, the data is reconstructed and several classification models can be applied for the processed data.

#### Different way to implement SMOTE: ([LINK](https://analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/))

1. SMOTE: Synthetic Minority Oversampling Technique
2. ADASYN: Adaptive Synthetic Sampling Approach
3. Hybridization: SMOTE + Tomek Links
4. Hybridization: SMOTE + ENN

### NearMiss Algorithm – Undersampling

It aims to balance class distribution by randomly eliminating majority class examples. When instances of two different classes are very close to each other, we remove the instances of the majority class to increase the spaces between the two classes. This helps in the classification process. To prevent problem of **information loss** in most under-sampling techniques, **near-neighbor** methods are widely used.

**The basic intuition about the working of near-neighbor methods is as follows:**

**Step 1:** The method first finds the distances between all instances of the majority class and the instances of the minority class. Here, majority class is to be under-sampled.

**Step 2:** Then, **n** instances of the majority class that have the smallest distances to those in the minority class are selected.

**Step 3:** If there are k instances in the minority class, the nearest method will result in **k\*n** instances of the majority class.

**For finding n closest instances in the majority class, there are several variations of applying NearMiss Algorithm:**

1. **NearMiss – Version 1:**It selects samples of the majority class for which average distances to the k **closest** instances of the minority class is smallest.
2. **NearMiss – Version 2:** It selects samples of the majority class for which average distances to the k **farthest** instances of the minority class is smallest.
3. **NearMiss – Version 3:** It works in 2 steps. Firstly, for each minority class instance, their **M nearest-neighbors** will be stored. Then finally, the majority class instances are selected for which the average distance to the N nearest-neighbors is the largest.

## Best practices for pre-processing data

**1. Understand Your Data**

* **Exploratory Data Analysis (EDA):** Before any preprocessing, spend time understanding your data. Use visualizations and summary statistics to get a sense of the distribution, relationships, and potential anomalies in your data.
* **Identify Data Types:** Know what types of data you are dealing with (e.g., numerical, categorical, text, image, time series).

**2. Ensure Data Consistency**

* **Remove Duplicates:** Ensure that duplicate entries are removed to prevent bias.
* **Ensure Correct Data Types:** Make sure all columns have appropriate data types (e.g., integers for IDs, floats for continuous values).

**3. Handle Missing Values**

* **Removal:** If missing values are few and random, consider removing the rows or columns.
* **Imputation:** Use techniques like mean, median, mode imputation for numerical data or the most frequent value for categorical data. Advanced methods include using machine learning models for imputation.

**4. Handle Outliers**

* **Identification:** Use statistical methods (e.g., Z-scores, IQR) or visualization techniques (e.g., box plots) to identify outliers.
* **Treatment:** Depending on the context, you may choose to remove, transform, or bin the outliers.

**5. Normalize or Standardize Data**

* **Normalization:** Rescale the data to a range of [0, 1] or [-1, 1] using Min-Max scaling. Useful for algorithms like k-NN or neural networks.
* **Standardization:** Transform data to have a mean of 0 and a standard deviation of 1. Useful for algorithms like SVM or logistic regression.

**6. Encode Categorical Variables**

* **Label Encoding:** Convert categorical values into integer values.
* **One-Hot Encoding:** Create binary columns for each category, useful for categorical variables without ordinal relationships.
* **Target Encoding:** Replace categories with the mean of the target variable.

**7. Feature Engineering**

* **Create New Features:** Derive new features from existing ones that might have better predictive power.
* **Polynomial Features:** For linear models, consider adding polynomial features to capture non-linear relationships.
* **Binning:** Group continuous data into bins to reduce the effect of noise and potentially highlight trends.

**8.****Feature Selection**

* **Remove Low-Variance Features:** Features with little variation may not add significant predictive power.
* **Correlation Analysis:** Remove features that are highly correlated with each other to reduce multicollinearity.
* **Model-Based Selection:** Use models like Lasso regression or tree-based methods to select important features.

**9. Dimensionality Reduction**

* **PCA (Principal Component Analysis):** Reduce the number of features while retaining most of the variance.
* **t-SNE or UMAP:** Useful for visualization and understanding the structure of high-dimensional data.

**10. Handling Imbalanced Data**

* **Resampling:** Use techniques like oversampling the minority class or under sampling the majority class.
* **Synthetic Data Generation:** Use techniques like SMOTE to create synthetic samples for the minority class.
* **Class Weighting:** Adjust the class weights in the learning algorithm to handle imbalances.

**10. Data Splitting**

* **Train-Test Split:** Split your data into training and testing sets to evaluate model performance.
* **Cross-Validation:** Use techniques like k-fold cross-validation to ensure your model generalizes well to unseen data.