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

Table of Contents

[1. Introduction 1](#_Toc178812028)

[1.1 Core Concepts in AI 1](#_Toc178812029)

[1.2 Prerequisites 1](#_Toc178812030)

[1.3 Machine Learning lifecycle 2](#_Toc178812031)

[1.4 Types of Machine Learning 2](#_Toc178812032)

[1.5 Introduction to Data in Machine Learning 4](#_Toc178812033)

[1.6 Best Python libraries for Machine Learning (Link) 4](#_Toc178812034)

[1.6.1 NumPy 4](#_Toc178812035)

[1.6.2 SciPy 5](#_Toc178812036)

[1.6.3 Scikit-learn 5](#_Toc178812037)

[1.6.4 TensorFlow 5](#_Toc178812038)

[1.6.5 Keras 5](#_Toc178812039)

[1.6.6 PyTorch 5](#_Toc178812040)

[1.6.7 Pandas 6](#_Toc178812041)

[1.6.8 Matplotlib 6](#_Toc178812042)

[1.6.9 Most Used Library 6](#_Toc178812043)

[2. Statistics 7](#_Toc178812044)

[2.1 Statistics 7](#_Toc178812045)

[2.2 Mean, Variance and Standard Deviation 7](#_Toc178812046)

[2.3 Types of Statistics 7](#_Toc178812047)

[2.4 Descriptive Statistic 7](#_Toc178812048)

[2.4.1 Types of Descriptive Statistics 7](#_Toc178812049)

[2.4.2 Applications of Descriptive Statistics 8](#_Toc178812050)

[2.5 Inferential Statistics 8](#_Toc178812051)

[2.5.1 Uses cases of Inferential Statistics 9](#_Toc178812052)

[2.5.2 Applications of Inferential Statistics 9](#_Toc178812053)

[2.6 Bias and Variance 9](#_Toc178812054)

[2.6.1 Bias 9](#_Toc178812055)

[2.6.2 Variance 10](#_Toc178812056)

[2.6.3 Different Combinations of Bias-Variance 10](#_Toc178812057)

[2.6.4 Bias Variance Tradeoff 11](#_Toc178812058)

[2.7 Hypothesis Testing 11](#_Toc178812059)

[2.7.1 Defining Hypotheses 12](#_Toc178812060)

[2.8 Covariance and Correlation 12](#_Toc178812061)

[2.8.1 Covariance 12](#_Toc178812062)

[2.8.2 Correlation 13](#_Toc178812063)

[2.8.3 Difference between Covariance and Correlation 14](#_Toc178812064)

[2.8.4 Applications of Covariance and Correlation 15](#_Toc178812065)

[3. Data and It’s Processing (Link) 16](#_Toc178812066)

[3.1 Data Type 16](#_Toc178812067)

[3.2 Exploratory Data Analysis (EDA) 17](#_Toc178812068)

[3.3 Data Cleaning vs Data Preprocessing 17](#_Toc178812069)

[3.4 Data Cleaning (Link) 17](#_Toc178812070)

[3.5 Data Preprocessing steps (Link) 18](#_Toc178812071)

[3.6 Outliers 19](#_Toc178812072)

[3.6.1 Outlier detection and removal 19](#_Toc178812073)

[3.6.2 Outlier vs Anomaly 20](#_Toc178812074)

[3.7 Feature Scaling Technique (Link)(Normalization or Standardization) 20](#_Toc178812075)

[3.7.1 Normalization or Min-Max Scaling 20](#_Toc178812076)

[3.7.2 Standardization or Z-Score Normalization 21](#_Toc178812077)

[3.8 Key Differences Between Feature Scaling and Feature Engineering 21](#_Toc178812078)

[3.9 Encoding Technique 22](#_Toc178812079)

[3.9.1 Label Encoding 22](#_Toc178812080)

[3.9.2 One-Hot Encoding 23](#_Toc178812081)

[3.9.3 Dummy Encoding 23](#_Toc178812082)

[3.9.4 Ordinal Encoding 24](#_Toc178812083)

[3.9.5 Binary Encoding 24](#_Toc178812084)

[3.9.6 Frequency Encoding (Count Encoding) 25](#_Toc178812085)

[3.9.7 Target Encoding (Mean Encoding) 26](#_Toc178812086)

[3.9.8 Leave-One-Out Encoding 26](#_Toc178812087)

[3.10 Handling Imbalanced Data with SMOTE and Near Miss Algorithm 27](#_Toc178812088)

[3.10.1 SMOTE (Synthetic Minority Oversampling Technique) 27](#_Toc178812089)

[3.10.2 NearMiss Algorithm – Undersampling 27](#_Toc178812090)

[3.11 Dummy variable trap in Regression Models 28](#_Toc178812091)

[3.12 Dimensionality Reduction – Principal Component Analysis (PCA) 29](#_Toc178812092)

[3.13 Feature Engineering 29](#_Toc178812093)

[3.14 Feature Selection 30](#_Toc178812094)

[3.15 Best practices for pre-processing data 31](#_Toc178812095)

# Introduction

Machine learning (ML) is a type of Artificial Intelligence (AI) that allows computers to learn without being explicitly programmed. It involves feeding data into algorithms that can then identify patterns and make predictions on new data.

## 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.

## Machine Learning lifecycle

1. Study the Problems
2. Data Collection
3. Data Preparation

* Data cleaning
* Data Transformation
* Explanatory Data Analysis and Feature Engineering
* Split the dataset for training and testing.

1. Model Selection
2. Model building and Training
3. Model Evaluation
4. Model Tuning
5. Deployment
6. Monitoring and Maintenance

## Types of Machine Learning

* [Supervised Machine Learning](https://www.geeksforgeeks.org/machine-learning/#su)
* [Unsupervised Machine Learning](https://www.geeksforgeeks.org/machine-learning/#unsu)
* [Reinforcement Machine Learning](https://www.geeksforgeeks.org/machine-learning/#hmm)

**1. Supervised Machine Learning:**

Supervised learning is a type of machine learning in which the algorithm is trained on the labelled dataset. It learns to map input features to targets based on labelled training data. In supervised learning, the algorithm is provided with input features and corresponding output labels, and it learns to generalize from this data to make predictions on new, unseen data.

There are two main types of supervised learning:

* [**Regression**](https://www.geeksforgeeks.org/types-of-regression-techniques/): Regression is a type of supervised learning where the algorithm learns to predict continuous values based on input features. The output labels in regression are continuous values, such as stock prices, and housing prices. The different regression algorithms in machine learning are: Linear Regression, Polynomial Regression, Ridge Regression, Decision Tree Regression, Random Forest Regression, Support Vector Regression, etc
* [**Classification**](https://www.geeksforgeeks.org/basic-concept-classification-data-mining/): Classification is a type of supervised learning where the algorithm learns to assign input data to a specific category or class based on input features. The output labels in classification are discrete values. Classification algorithms can be binary, where the output is one of two possible classes, or multiclass, where the output can be one of several classes. The different Classification algorithms in machine learning are: Logistic Regression, Naive Bayes, Decision Tree, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), etc

**2. Unsupervised Machine Learning:**

Unsupervised learning is a type of machine learning where the algorithm learns to recognize patterns in data without being explicitly trained using labelled examples. The goal of unsupervised learning is to discover the underlying structure or distribution in the data.

There are two main types of unsupervised learning:

* [**Clustering**](https://www.geeksforgeeks.org/clustering-in-machine-learning/): Clustering algorithms group similar data points together based on their characteristics. The goal is to identify groups, or clusters, of data points that are similar to each other, while being distinct from other groups. Some popular clustering algorithms include K-means, Hierarchical clustering, and DBSCAN.
* [**Dimensionality reduction:**](https://www.geeksforgeeks.org/dimensionality-reduction/) Dimensionality reduction algorithms reduce the number of input variables in a dataset while preserving as much of the original information as possible. This is useful for reducing the complexity of a dataset and making it easier to visualize and analyse. Some popular dimensionality reduction algorithms include Principal Component Analysis (PCA), t-SNE, and Autoencoders.

**3. Reinforcement Machine Learning**

Reinforcement learning is a type of machine learning where an agent learns to interact with an environment by performing actions and receiving rewards or penalties based on its actions. The goal of reinforcement learning is to learn a policy, which is a mapping from states to actions, that maximizes the expected cumulative reward over time.

There are two main types of reinforcement learning:

* **Model-based reinforcement learning:** In model-based reinforcement learning, the agent learns a model of the environment, including the transition probabilities between states and the rewards associated with each state-action pair. The agent then uses this model to plan its actions in order to maximize its expected reward. Some popular model-based reinforcement learning algorithms include Value Iteration and Policy Iteration.
* [**Model-free reinforcement learning**](https://www.geeksforgeeks.org/ml-reinforcement-learning-algorithm-python-implementation-using-q-learning/): In model-free reinforcement learning, the agent learns a policy directly from experience without explicitly building a model of the environment. The agent interacts with the environment and updates its policy based on the rewards it receives. Some popular model-free reinforcement learning algorithms include Q-Learning, SARSA, and Deep Reinforcement Learning.

## Introduction to Data in Machine Learning

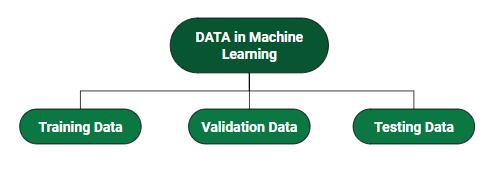
Data is typically divided into two types:

* 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****

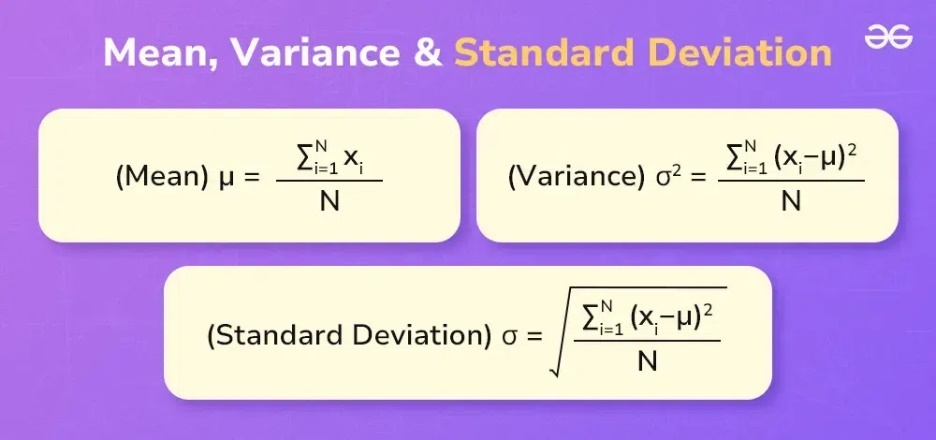
* **Pandas** and **NumPy** are the most universally used libraries because they form the foundation of almost any data science or machine learning project.
* For machine learning, **Scikit-learn** is a go-to for traditional algorithms, while **TensorFlow** and **PyTorch** are the dominant players in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

# Statistics

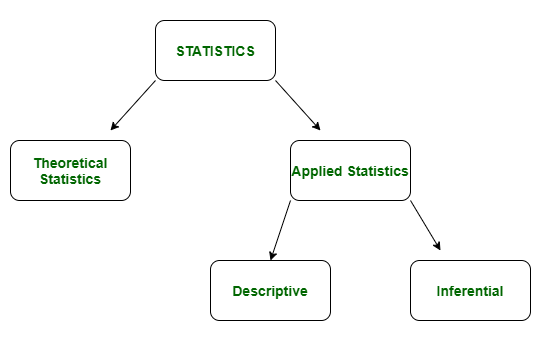
## Statistics

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

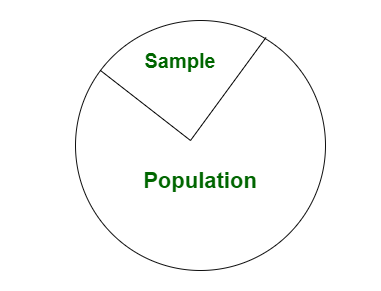
Business Analysis: Summarizing sales data to identify trends and make informed business decisions.

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.

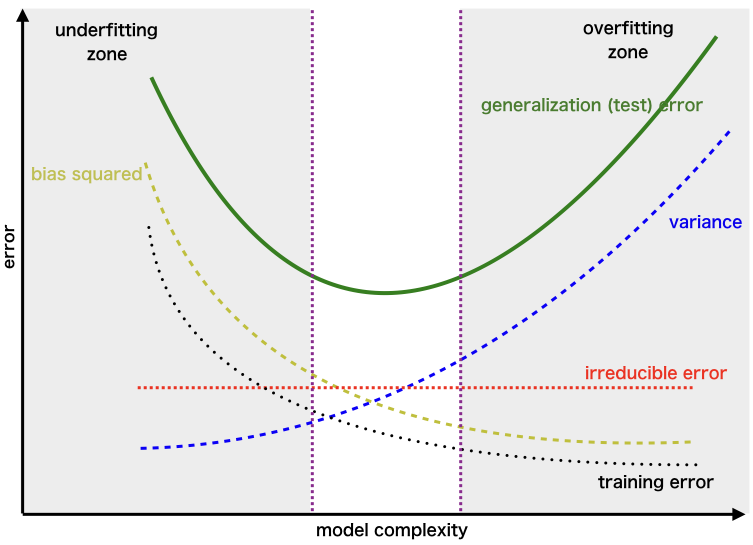
### Different Combinations of Bias-Variance

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

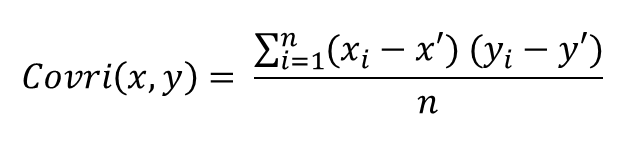
**Covariance is a statistical measure** that **indicates the direction of the linear relationship between two variables**. It assesses how much two variables change together from their mean values.

Types of Covariance:

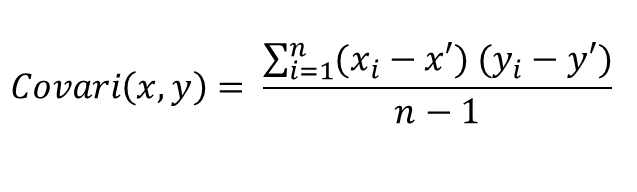
* **Positive Covariance**: When one variable increases, the other variable tends to increase as well, and vice versa.
* **Negative Covariance**: When one variable increases, the other variable tends to decrease.
* **Zero Covariance**: There is no linear relationship between the two variables; they move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

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

**For Population:**

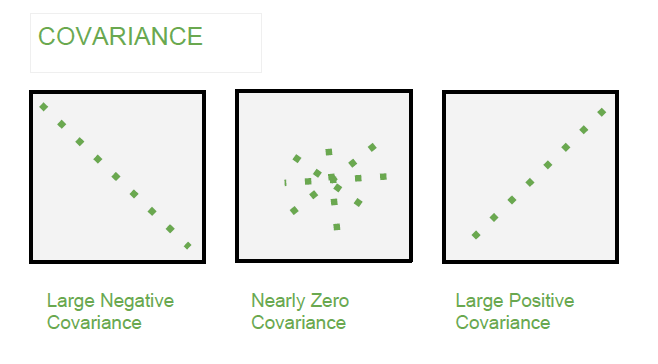


**For Sample:**



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

**Example –**



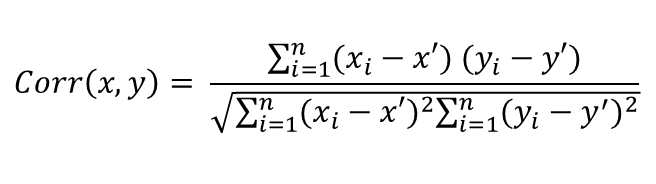
### Correlation

**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

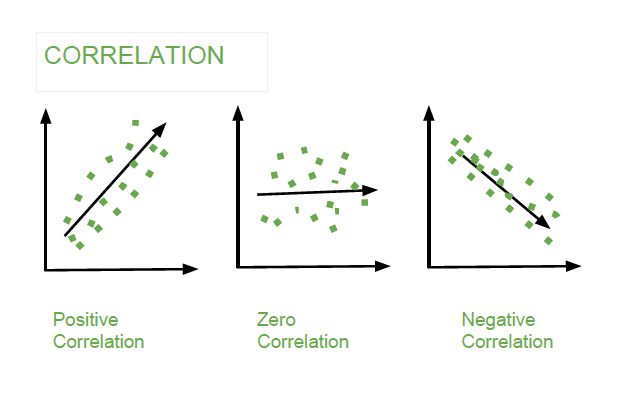
Strong: 0.50 to 0.79

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:

|  |  |  |  |
| --- | --- | --- | --- |
| **Correlation Coefficient** | **Type of Relation** | **Levels of Measurement** | **Data Distribution** |
| **Pearson Correlation Coefficient** | Linear | Interval/Ratio | Normal distribution |
| **Spearman Rank Correlation Coefficient** | Non-Linear | Ordinal | Any distribution |
| **Kendall Tau Coefficient** | Non-Linear | Ordinal | Any distribution |
| **Phi Coefficient** | Non-Linear | Nominal vs. Nominal (nominal with 2 categories (dichotomous)) | Any distribution |
| **Cramer’s V** | Non-Linear | Two nominal variables | Any distribution |

### 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

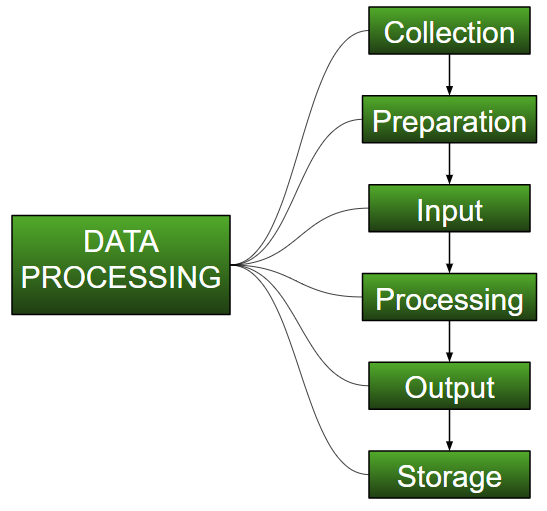
* **Portfolio Management in Finance**: Covariance is used to measure how different stocks or financial assets move together, aiding in portfolio diversification to minimize risk.
* **Genetics**: In genetics, covariance can help understand the relationship between different genetic traits and how they 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.

## 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.

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

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

## 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.

## [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 and removal

1. **Visualizing and Removing Outliers Using Box Plot**
2. **Visualizing and Removing Outliers Using Scatterplot**
3. **Removal of Outliers with Z-Score (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).***

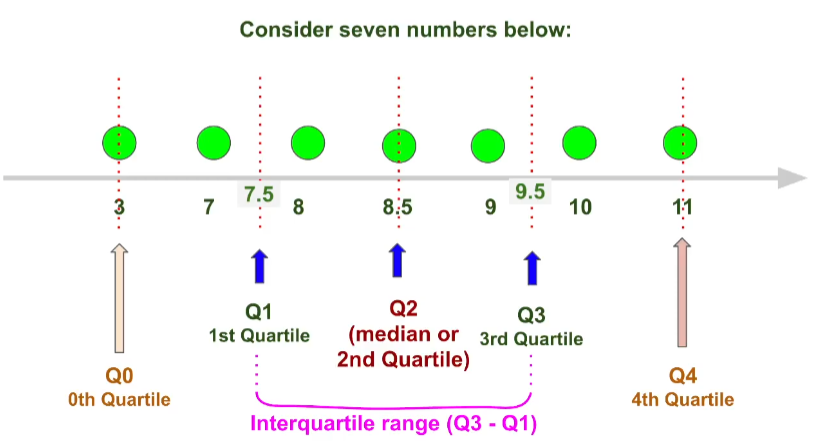
1. **Removal of outliers with using IQR (More Robust Approach)**

[**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*

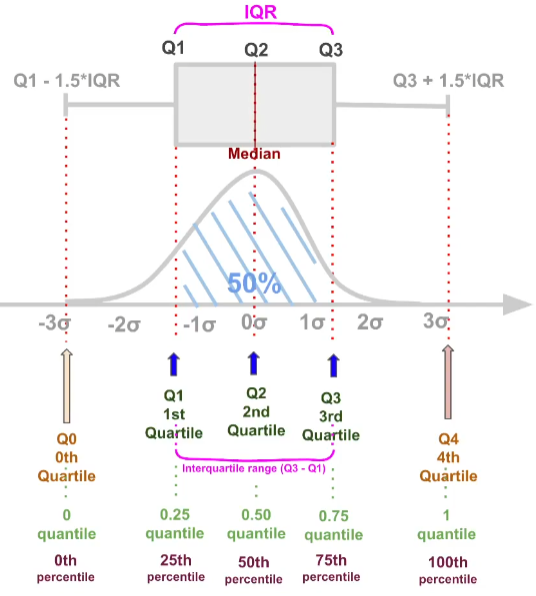
**

***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



### 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 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:

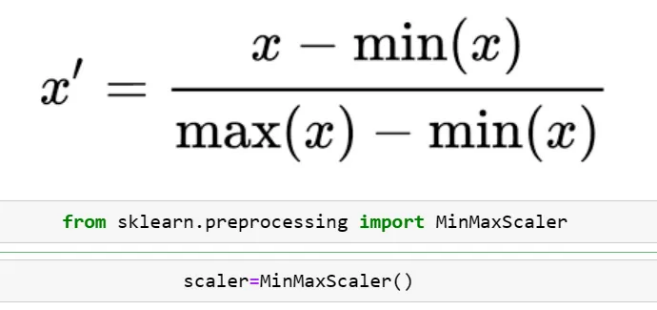
* **Standardization:**
  + Standard Scaler (Centered around 0)
* **Normalization:**
  + Min Max Scaling [0, 1]
  + Mean Normalization [-1, 1]
  + Max Absolute Scaling [0, 1] or [-1, 1]
  + Robust Scaling (Varies, typically preserves the interquartile range)

### Normalization or Min-Max Scaling

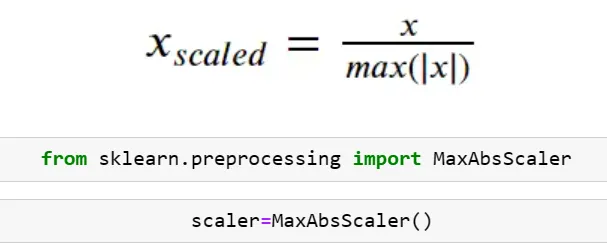
It is a scaling technique method in which data points are shifted and rescaled so that they end up in a range of 0 to 1. It is also known as **min-max scaling**.

Type of normalization:

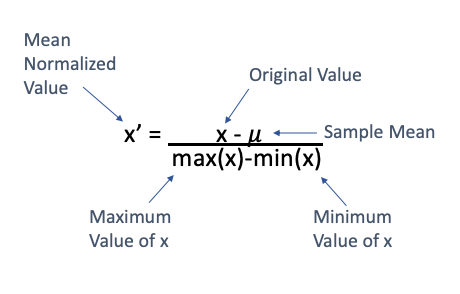
* MinMax Scaling (mostly used)



* MaxAbScaling

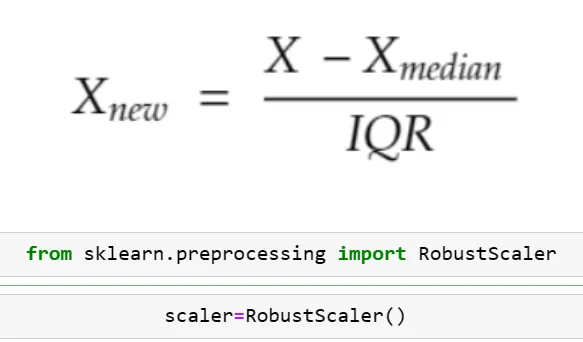


* Mean Normalization



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

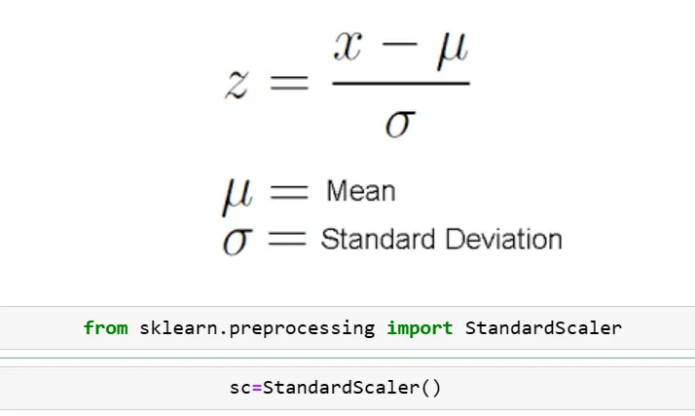
* Robust Scaling



### 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. |

## Key Differences Between Feature Scaling and Feature Engineering

|  |  |  |
| --- | --- | --- |
| Aspect | Feature Scaling | Feature Engineering |
| Definition | Adjusting feature ranges and magnitudes. | Creating or modifying features to enhance model performance. |
| Goal | Ensure numerical features are comparable in range. | Increase the predictive power by using domain knowledge. |
| Techniques | Normalization, Standardization, Robust Scaling. | Polynomial Features, Encoding, Log Transformation, Feature Binning. |
| When to Use | Before applying models like KNN, SVM, Linear Regression. | During feature selection, model building, and data preprocessing. |
| Impact | Ensures uniform scaling, helping with optimization. | Creates new, more meaningful features that help the model learn. |
| Model Dependency | Highly dependent on the machine learning algorithm. | Typically, useful across different types of algorithms. |

* **Feature Scaling** is often used with machine learning models that involve distance calculations or gradient descent (like **Neural Networks**, **K-Means**, or **SVM**).

**Scenario**: You have features like height (in cm), weight (in kg), and income (in thousands). You apply **Min-Max Scaling** or **Standardization** to bring them into a similar range before training a model.

* **Feature Engineering** can significantly improve a model’s accuracy and performance by adding new insights.

**Scenario**: You have transaction data with columns like "product\_price" and "quantity". You create a new feature called "total\_purchase" to represent the total value of each transaction.

## 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** 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:**

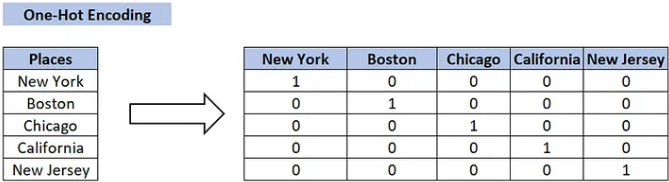
* **Nominal data** where categories have no order (e.g., colours, countries).
* Works well for categorical data with **small numbers of unique categories**.

**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.



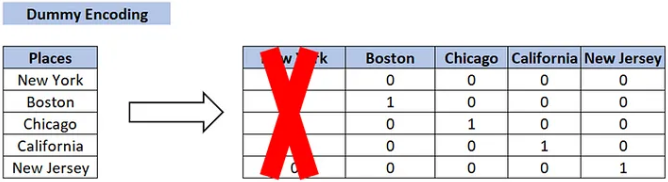
### Dummy Encoding

• 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:**

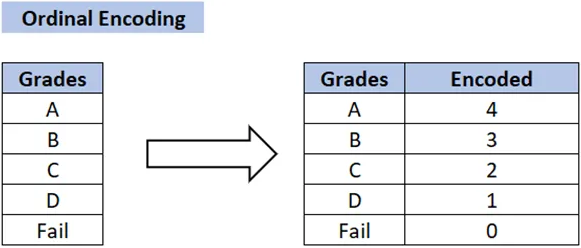
* For **ordinal categorical data** where the categories have a clear ranking (e.g., "Low", "Medium", "High").

**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:**

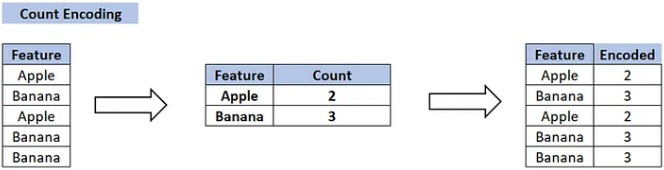
* Useful when categories have varying frequencies that may carry useful information.
* For **nominal data**, especially when there is a meaningful difference between how often categories occur.

**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:**

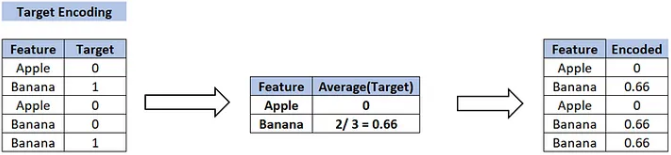
* Used in **supervised learning** problems where the target variable is available during training.
* For both **binary and categorical target variables**.

**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.

## 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.

## 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.

## Dimensionality Reduction – Principal Component Analysis (PCA)

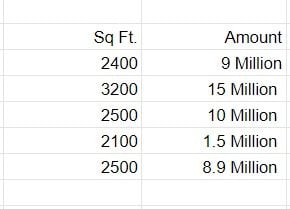
To address the [curse of dimensionality](https://www.geeksforgeeks.org/videos/curse-of-dimensionality-in-machine-learning/), [Feature engineering](https://www.geeksforgeeks.org/what-is-feature-engineering/)techniques are used which include feature selection and feature extraction. [Dimensionality reduction](https://www.geeksforgeeks.org/dimensionality-reduction/) is a type of feature extraction technique that aims to reduce the number of input features while retaining as much of the original information as possible.

Principal Component Analysis (PCA) is used to reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables, retaining most of the sample’s information, and useful for the [regression and classification](https://www.geeksforgeeks.org/regression-classification-supervised-machine-learning/) of data.

## Feature Engineering

Feature engineering is a machine learning technique that leverages data to create new variables that aren’t in the training set. It can produce new features for both supervised and unsupervised learning, with the goal of simplifying and speeding up data transformations while also enhancing model accuracy. Feature engineering is required when working with machine learning models. Regardless of the data or architecture, a terrible feature will have a direct impact on your model.

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.



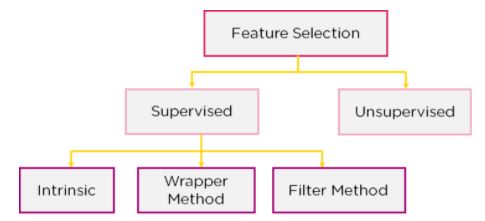
## Feature Selection

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:

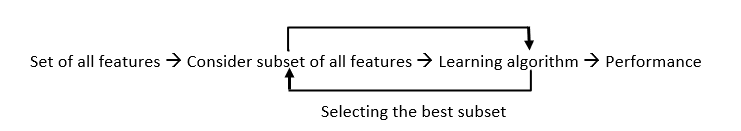
1. Supervised Models: Supervised feature selection refers to the method which uses the output label class for feature selection. They use the target variables to identify the variables which can increase the efficiency of the model
2. Unsupervised Models: Unsupervised feature selection refers to the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



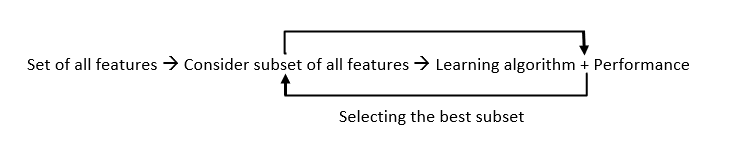
1. **Filter Method**: In this method, features are dropped based on their relation to the output, or how they are correlating to the output. We use correlation to check if the features are positively or negatively correlated to the output labels and drop features accordingly. E.g.: Information Gain, [Chi-Square Test](https://www.simplilearn.com/tutorials/statistics-tutorial/chi-square-test), Fisher’s Score, etc.



2. 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.



3. 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.



## 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.