|  |  |
| --- | --- |
|  |  |
| **Machine Learning** |  |
|  |  |
|  | DATE – 07/11/2022  Machine Learning |
|  | Zodex |

Table of Contents

**No table of contents entries found.**

# Introduction

Machine learning is a branch of artificial intelligence that enables algorithms to uncover hidden patterns within datasets, allowing them to make predictions on new, similar data without explicit programming for each task. Traditional machine learning combines data with statistical tools to predict outputs, yielding actionable insights. This technology finds applications in diverse fields such as image and speech recognition, natural language processing, recommendation systems, fraud detection, portfolio optimization, and automating tasks.

## 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

## Type of Machine Learning

There are several types of machine learning, each with special characteristics and applications. Some of the main types of machine learning algorithms are as follows:

1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Semi-Supervised Machine Learning
4. Reinforcement Learning

### Supervised Machine Learning

**Supervised machine learning** is a fundamental approach within the broader field of machine learning and artificial intelligence. It involves training algorithms using labeled datasets, where each input is paired with the correct output. **Supervised learning** allows the algorithm to learn the mapping from inputs to outputs, enabling it to make predictions or decisions when presented with new, unseen data.

#### Types of Supervised Machine Learning

1. **Classification**

[**Classification**](https://www.geeksforgeeks.org/getting-started-with-classification/)**deals with predicting categorical target variables, which represent discrete classes or labels. For instance, classifying emails as spam or not spam, or predicting whether a patient has a high risk of heart disease. Classification algorithms learn to map the input features to one of the predefined classes.**

**Here are some classification algorithms:**

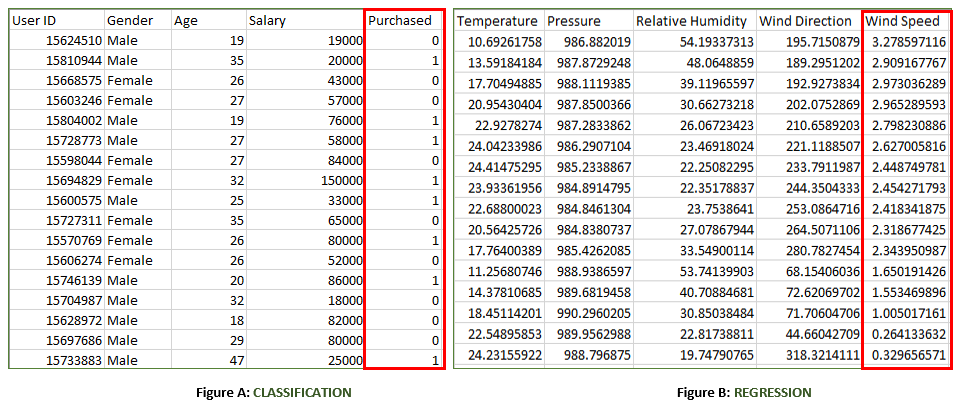
* [**Logistic Regression**](https://www.geeksforgeeks.org/understanding-logistic-regression/)
* [**Support Vector Machine**](https://www.geeksforgeeks.org/support-vector-machine-algorithm/)
* [**Random Forest**](https://www.geeksforgeeks.org/random-forest-regression-in-python/)
* [**Decision Tree**](https://www.geeksforgeeks.org/decision-tree/)
* [**K-Nearest Neighbors (KNN)**](https://www.geeksforgeeks.org/k-nearest-neighbours/)
* [**Naive Bayes**](https://www.geeksforgeeks.org/naive-bayes-classifiers/)

1. **Regression**

[**Regression**](https://www.geeksforgeeks.org/regression-classification-supervised-machine-learning/)**, on the other hand, deals with predicting continuous target variables, which represent numerical values. For example, predicting the price of a house based on its size, location, and amenities, or forecasting the sales of a product. Regression algorithms learn to map the input features to a continuous numerical value.**

**Here are some regression algorithms:**

* [**Linear Regression**](https://www.geeksforgeeks.org/ml-linear-regression/)
* [**Polynomial Regression**](https://www.geeksforgeeks.org/videos/polynomial-regression-algorithm-machine-learning/)
* [**Ridge Regression**](https://www.geeksforgeeks.org/videos/lasso-ridge-regression-algorithm-machine-learning/)
* [**Lasso Regression**](https://www.geeksforgeeks.org/videos/lasso-ridge-regression-algorithm-machine-learning/)
* [**Decision tree**](https://www.geeksforgeeks.org/decision-tree-introduction-example/)
* [**Random Forest**](https://www.geeksforgeeks.org/random-forest-regression-in-python/)



#### Supervised Machine Learning Algorithms

**Supervised learning** can be further divided into several different types, each with its own unique characteristics and applications. Here are some of the most common types of supervised learning algorithms:

* [**Linear Regression**](https://www.geeksforgeeks.org/ml-linear-regression/): Linear regression is a type of **supervised learning regression algorithm** that is used to predict a continuous output value. It is one of the simplest and most widely used algorithms in supervised learning.
* [**Logistic Regression**](https://www.geeksforgeeks.org/understanding-logistic-regression/): Logistic regression is a type of **supervised learning classification algorithm** that is used to predict a binary output variable. It is commonly used in machine learning applications where the output variable is either true or false, such as in fraud detection or spam filtering.
* [**Decision Trees**](https://www.geeksforgeeks.org/decision-tree/): Decision tree is a tree-like structure that is used to model decisions and their possible consequences. Each internal node in the [tree](https://www.geeksforgeeks.org/introduction-to-tree-data-structure-and-algorithm-tutorials/)represents a decision, while each leaf node represents a possible outcome. Decision trees can be used to model complex relationships between input features and output variables. A decision tree is a type of **machine learning algorithm** that is used for both classification and regression tasks.
* [**Random Forests**](https://www.geeksforgeeks.org/random-forest-regression-in-python/): Random forests again are made up of multiple decision trees that work together to make predictions. Each tree in the forest is trained on a different subset of the input features and data. The final prediction is made by aggregating the predictions of all the trees in the forest. Random forests are an ensemble **machine learning technique** that is used for both **classification and regression tasks in supervised learning**.
* [**Support Vector Machine(SVM)**](https://www.geeksforgeeks.org/support-vector-machine-algorithm/): The SVM algorithm creates a hyperplane to segregate n-dimensional space into classes and identify the correct category of new data points. The extreme cases that help create the hyperplane are called support vectors, hence the name Support Vector Machine. A Support Vector Machine is a type of supervised machine learning algorithm that is also used for both classification and regression tasks.
* [**K-Nearest Neighbors**](https://www.geeksforgeeks.org/k-nearest-neighbours/)**(KNN) :**KNN works by finding k training examples closest to a given input and then predicts the class or value based on the majority class or average value of these neighbors. The performance of KNN can be influenced by the choice of k and the distance metric used to measure proximity. However, it is intuitive but can be sensitive to noisy data and requires careful selection of k for optimal results. A K-Nearest Neighbors (KNN) is a type of algorithm that is used for both classification and regression tasks.
* [**Gradient Boosting**](https://www.geeksforgeeks.org/ml-gradient-boosting/): Gradient Boosting combines weak learners, like [decision trees](https://www.geeksforgeeks.org/decision-tree/), to create a strong model. It iteratively builds new models that correct errors made by previous ones. Each new model is trained to minimize residual errors, resulting in a powerful predictor capable of handling complex data relationships. A Gradient Boosting is a type of algorithm that is used for both classification and regression tasks.
* [**Naive Bayes Algorithm**](https://www.geeksforgeeks.org/naive-bayes-classifiers/): The **Naive Bayes algorithm** is a **supervised machine learning algorithm** based on applying [Bayes’ Theorem](https://www.geeksforgeeks.org/bayes-theorem/) with the “naive” assumption that features are independent of each other given the class label. Despite this simplifying assumption, Naive Bayes performs well for many real-world tasks, especially in text classification, spam detection, and document categorization.

### Unsupervised Learning

Unsupervised machine learning models, in contrast to [supervised learning](https://www.geeksforgeeks.org/supervised-machine-learning/), are given unlabelled data and allow discover patterns and insights on their own—without explicit direction or instruction.

#### Types of Unsupervised Machine Learning

There are mainly 3 types of Algorithms which are used for Unsupervised dataset.

* Clustering
* Association Rule Learning
* Dimensionality Reduction

1. **Clustering**

[Clustering](https://www.geeksforgeeks.org/clustering-in-machine-learning/) in unsupervised machine learning is the process of grouping unlabelled data into clusters based on their similarities. The goal of clustering is to identify patterns and relationships in the data without any prior knowledge of the data’s meaning.

Broadly this technique is applied to group data based on different patterns, such as similarities or differences, our machine model finds. These algorithms are used to process raw, unclassified data objects into groups. For example, in the above figure, we have not given output parameter values, so this technique will be used to group clients based on the input parameters provided by our data.

Some common clustering algorithms

* [K-means Clustering](https://www.geeksforgeeks.org/k-means-clustering-introduction/): Partitioning Data into K Clusters
* [Hierarchical Clustering](https://www.geeksforgeeks.org/ml-hierarchical-clustering-agglomerative-and-divisive-clustering/): Building a Hierarchical Structure of Clusters
* [Density-Based Clustering (DBSCAN)](https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering/): Identifying Clusters Based on Density
* [Mean-Shift Clustering](https://www.geeksforgeeks.org/ml-mean-shift-clustering/): Finding Clusters Based on Mode Seeking
* [Spectral Clustering](https://www.geeksforgeeks.org/ml-spectral-clustering/): Utilizing Spectral Graph Theory for Clustering

1. **Association Rule Learning**

[Association rule learning](https://www.geeksforgeeks.org/association-rule/) is also known as association rule mining is a common technique used to discover associations in unsupervised machine learning. This technique is a rule-based ML technique that finds out some very useful relations between parameters of a large data set. This technique is basically used for market basket analysis that helps to better understand the relationship between different products. For e.g. shopping stores use algorithms based on this technique to find out the relationship between the sale of one product w.r.t to another’s sales based on customer behavior. Like if a customer buys milk, then he may also buy bread, eggs, or butter. Once trained well, such models can be used to increase their sales by planning different offers.

* [Apriori Algorithm](https://www.geeksforgeeks.org/apriori-algorithm/): A Classic Method for Rule Induction
* [FP-Growth Algorithm](https://www.geeksforgeeks.org/frequent-pattern-growth-algorithm/): An Efficient Alternative to Apriori
* [Eclat Algorithm](https://www.geeksforgeeks.org/ml-eclat-algorithm/): Exploiting Closed Item sets for Efficient Rule Mining
* [Efficient Tree-based Algorithms](https://www.geeksforgeeks.org/introduction-to-tree-data-structure-and-algorithm-tutorials/): Handling Large Datasets with Scalability

1. **Dimensionality Reduction**

Dimensionality reduction is the process of reducing the number of features in a dataset while preserving as much information as possible. This technique is useful for improving the performance of machine learning algorithms and for data visualization. Examples of dimensionality reduction algorithms include Dimensionality reduction is the process of reducing the number of features in a dataset while preserving as much information as possible.

* [Principal Component Analysis (PCA)](https://www.geeksforgeeks.org/principal-component-analysis-pca/): Linear Transformation for Reduced Dimensions
* [Linear Discriminant Analysis (LDA)](https://www.geeksforgeeks.org/ml-linear-discriminant-analysis/): Dimensionality Reduction for Discrimination
* [Non-negative Matrix Factorization (NMF](https://www.geeksforgeeks.org/non-negative-matrix-factorization/)): Decomposing Data into Non-negative Components
* [Locally Linear Embedding (LLE)](https://www.geeksforgeeks.org/locally-linear-embedding-in-machine-learning/): Preserving Local Geometry in Reduced Dimensions
* Isomap: Capturing Global Relationships in Reduced Dimensions

### Semi-Supervised Learning

[Semi-Supervised learning](https://www.geeksforgeeks.org/ml-semi-supervised-learning/)is a machine learning algorithm that works between the [supervised and unsupervised](https://www.geeksforgeeks.org/supervised-unsupervised-learning/) learning so it uses both **labelled and unlabelled** data. It’s particularly useful when obtaining labelled data is costly, time-consuming, or resource-intensive. This approach is useful when the dataset is expensive and time-consuming. Semi-supervised learning is chosen when labelled data requires skills and relevant resources in order to train or learn from it.

We use these techniques when we are dealing with data that is a little bit labelled and the rest large portion of it is unlabelled. We can use the unsupervised techniques to predict labels and then feed these labels to supervised techniques. This technique is mostly applicable in the case of image data sets where usually all images are not labelled.

#### Types of Semi-Supervised Learning Methods

There are a number of different semi-supervised learning methods each with its own characteristics. Some of the most common ones include:

* **Graph-based semi-supervised learning:** This approach uses a graph to represent the relationships between the data points. The graph is then used to propagate labels from the labeled data points to the unlabeled data points.
* **Label propagation:** This approach iteratively propagates labels from the labeled data points to the unlabeled data points, based on the similarities between the data points.
* **Co-training:** This approach trains two different machine learning models on different subsets of the unlabeled data. The two models are then used to label each other’s predictions.
* **Self-training:** This approach trains a machine learning model on the labeled data and then uses the model to predict labels for the unlabeled data. The model is then retrained on the labeled data and the predicted labels for the unlabeled data.
* [**Generative adversarial networks (GANs)**](https://www.geeksforgeeks.org/generative-adversarial-network-gan/)**:** GANs are a type of deep learning algorithm that can be used to generate synthetic data. GANs can be used to generate unlabeled data for semi-supervised learning by training two neural networks, a generator and a discriminator.

### Reinforcement Machine Learning

[Reinforcement machine learning](https://www.geeksforgeeks.org/what-is-reinforcement-learning/)algorithm is a learning method that interacts with the environment by producing actions and discovering errors. **Trial, error, and delay** are the most relevant characteristics of reinforcement learning. In this technique, the model keeps on increasing its performance using Reward Feedback to learn the behavior or pattern. These algorithms are specific to a particular problem e.g. Google Self Driving car, AlphaGo where a bot competes with humans and even itself to get better and better performers in Go Game.

#### Reinforcement Learning Algorithms

* [**Q-learning:**](https://www.geeksforgeeks.org/q-learning-in-python/) Q-learning is a model-free RL algorithm that learns a Q-function, which maps states to actions. The Q-function estimates the expected reward of taking a particular action in a given state.
* [**SARSA (State-Action-Reward-State-Action):**](https://www.geeksforgeeks.org/sarsa-reinforcement-learning/) SARSA is another model-free RL algorithm that learns a Q-function. However, unlike Q-learning, SARSA updates the Q-function for the action that was actually taken, rather than the optimal action.
* [**Deep Q-learning**](https://www.geeksforgeeks.org/deep-q-learning/)**:** Deep Q-learning is a combination of Q-learning and deep learning. Deep Q-learning uses a neural network to represent the Q-function, which allows it to learn complex relationships between states and actions.

## Model-Based vs Instance-Based Learning

### Model-Based Learning

Model-based learning involves creating a mathematical model that can predict outcomes based on input data. The model is trained on a large dataset and then used to make predictions on new data. The model can be thought of as a set of rules that the machine uses to make predictions.

**Advantages of Model-Based Learning**

1. Faster predictions
2. More accurate predictions
3. Better understanding of data

**Disadvantages of Model-Based Learning**

1. Requires a large dataset
2. Requires expert knowledge

### Instance-Based Learning

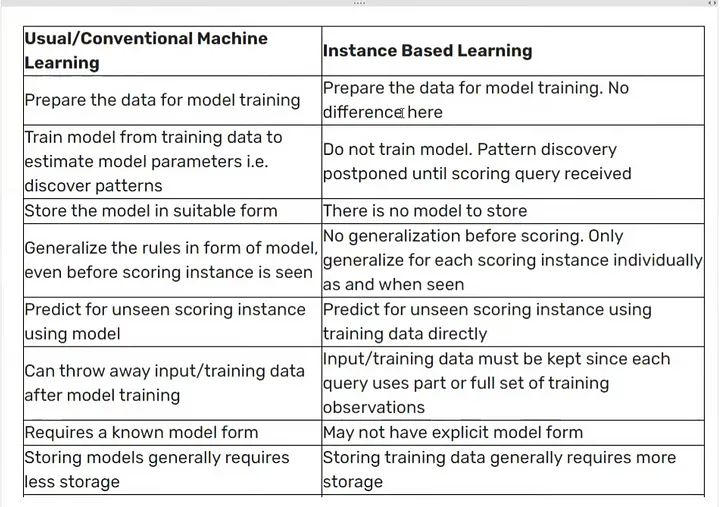
Instance-based learning involves using the entire dataset to make predictions. The machine learns by storing all instances of data and then using these instances to make predictions on new data. The machine compares the new data to the instances it has seen before and uses the closest match to make a prediction.

**Advantages of Instance-Based Learning**

1. No need for model creation
2. Can handle small datasets
3. More flexibility

**Disadvantages of Instance-Based Learning**

1. Slower predictions
2. Less accurate predictions
3. Limited understanding of data



## 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

