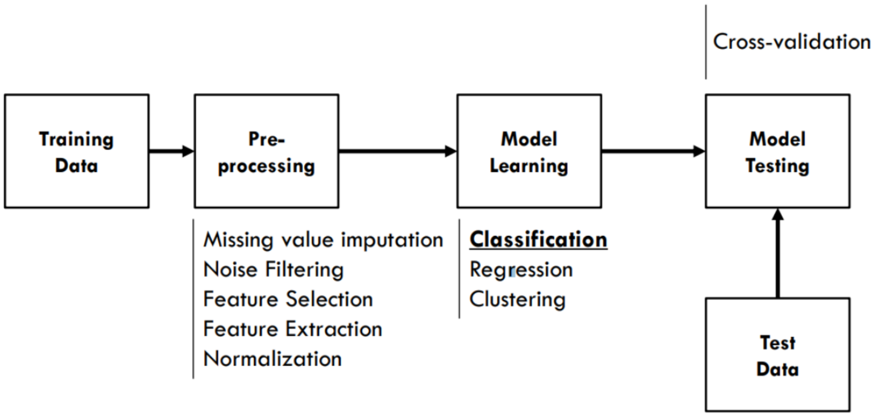
Unit 2 : Supervised Learning

Learning Process in ML



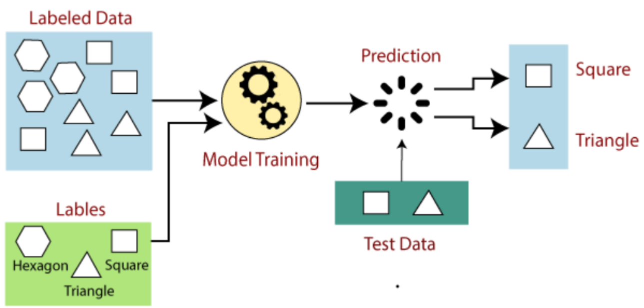
- **Training Data**: The dataset used to train the machine learning model, containing input features and (for supervised learning) corresponding labels.
- **Pre-processing**: Data is cleaned and transformed by handling missing values, filtering noise, selecting/extracting features, and normalizing for better model performance.
- **Model Learning**: The model is trained using algorithms to identify patterns and relationships in the pre-processed data.
- **Model Testing**: The trained model is evaluated on unseen test data to measure its accuracy and generalization ability.
- **Test Data**: A separate dataset used to validate the model's performance and ensure it works well on real-world data.

Supervised Learning :

- Supervised learning is a subset of AI and ML.
- It is also know as supervised machine learning as is defined by its ability to train algorithms to categorize data and predict outcomes accutately.

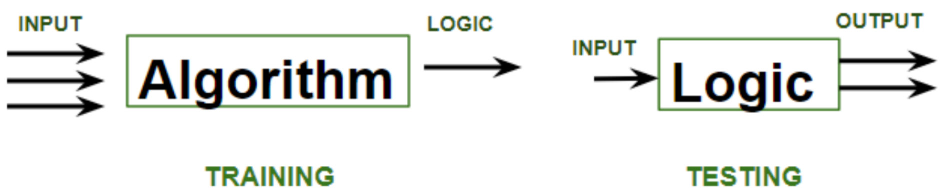
SL is a type of machine learning where a model is trained on labeled data—meaning each input is paired with the correct output.

- **WORKING :**

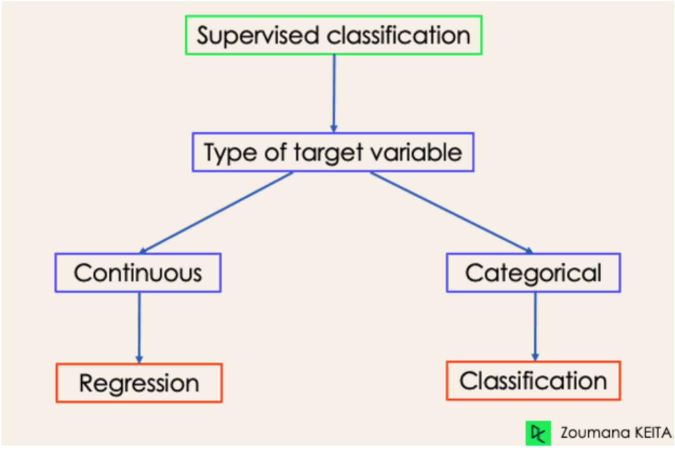


Where **supervised learning algorithm** consists of input features and corresponding output labels. The process works through:

- **Training Data:** The model is provided with a training dataset that includes input data (features) and corresponding output data (labels or target variables).
- **Learning Process:**the model learns by comparing its predictions with the actual answers provided in the training data.
- This is achieved by adjusting the model's parameters to minimize the difference between its predictions and the actual labels.
- Over time, it adjusts itself to minimize errors and improve accuracy.
- The goal of supervised learning is to make accurate predictions when given new, unseen data.



- there are two types of SL. i.e Regression & Classification



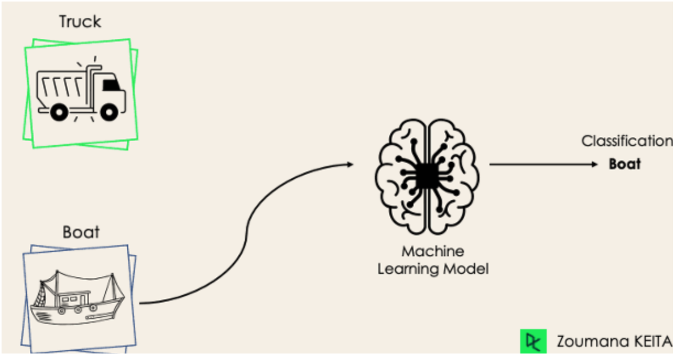
Classification

Classification teaches a machine to sort things into categories. It learns by looking at examples with labels. After learning, it can decide which category new items belong to.

Types of classification :

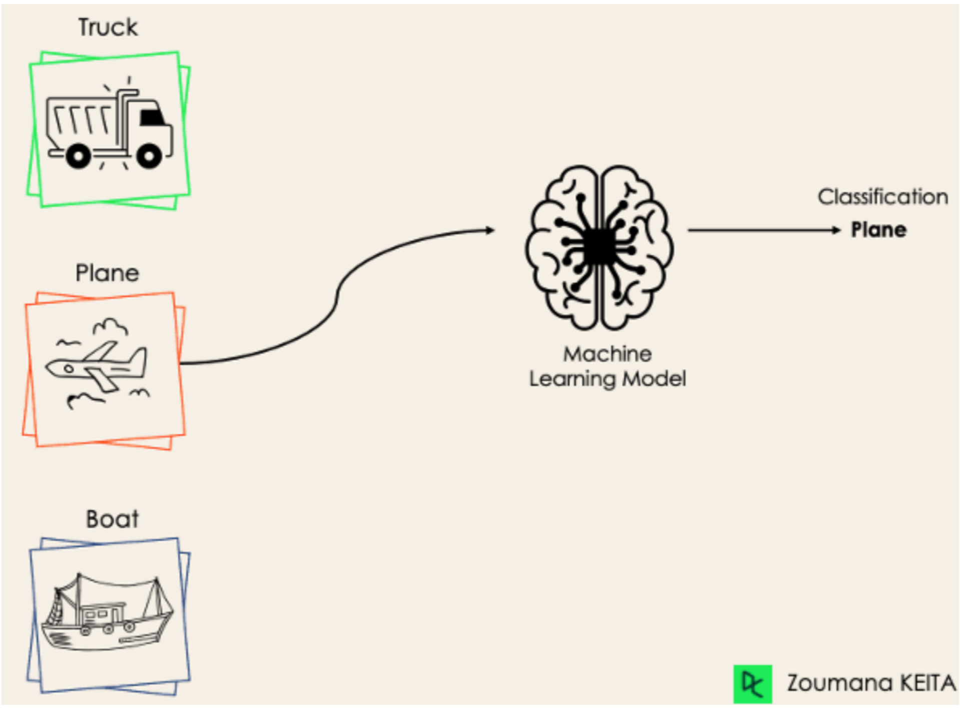
Binary Classification:

- In a binary classification task, the goal is to classify the input data into two mutually exclusive categories.
- The training data in such a situation is labeled in a binary format: true and false; positive and negative; 0 and 1.
- Depending on the problem being tackled.
- For instance, we might want to detect whether a given image is a truck or a boat.
- Logistic Regression and Support Vector Machines algorithms are natively designed for binary classifications.
- However, other algorithms such as K-Nearest Neighbors and Decision Trees can also be used for binary classification.



**Multi Class Classification

- The multi-class classification, on the other hand, has at least two mutually exclusive class labels, where the goal is to predict to which class a given input example belongs to.
- In the following case, the model correctly classified the image to be a plane.



- Most of the binary classification algorithms can be also used for multi-class classification. These algorithms include but are not limited to:
  - Random Forest
  - Naive Bayes
  - K-Nearest Neighbors
  - Gradient Boosting

- SVM
- Logistic Regression.

However, we can apply binary transformation approaches such as one-versus-one and one-versus-all to adapt native binary classification algorithms for multi-class classification tasks.

One-Versus-One (OvO) vs. One-Versus-Rest (OvR) Classification

Both **One-Versus-One (OvO)** and **One-Versus-Rest (OvR)** are strategies used for **multiclass classification** by breaking down the problem into multiple **binary classification tasks**.

---

1. One-Versus-One (OvO)

- This strategy **trains a separate binary classifier for every possible pair of classes**.
- For **N classes**, the number of classifiers needed is **N × (N-1) / 2**.
- Each classifier is trained on a **subset** of the data containing only the two classes involved in that specific comparison.
- The final prediction is made by a **majority vote**, where the class receiving the most votes across all pairwise classifiers is chosen.
- **Example:** In a 3-class classification problem (Cat, Dog, Bird), the model trains three classifiers:
    - **Classifier 1:** Cat vs. Dog
    - **Classifier 2:** Cat vs. Bird
    - **Classifier 3:** Dog vs. Bird
- **Best for: Support Vector Machines (SVM)** and **kernel-based algorithms** because they work efficiently with smaller datasets per classifier.

---

2. One-Versus-Rest (OvR) (One-Versus-All)

- This strategy **trains N binary classifiers**, where each classifier distinguishes **one class from all the others combined**.
- Each classifier is trained to answer: "Is this input class **X**, or is it something else?"

- **Example:** In a 3-class classification problem (Cat, Dog, Bird), the model trains three classifiers:
  - **Classifier 1:** Cat vs. (Dog + Bird)
  - **Classifier 2:** Dog vs. (Cat + Bird)
  - **Classifier 3:** Bird vs. (Cat + Dog)
- The final prediction is made based on the classifier with the **highest confidence score or probability**.
- **Best for: Logistic regression, decision trees, neural networks**, and when the number of classes is large.

Multi-Labelled Classification

Definition

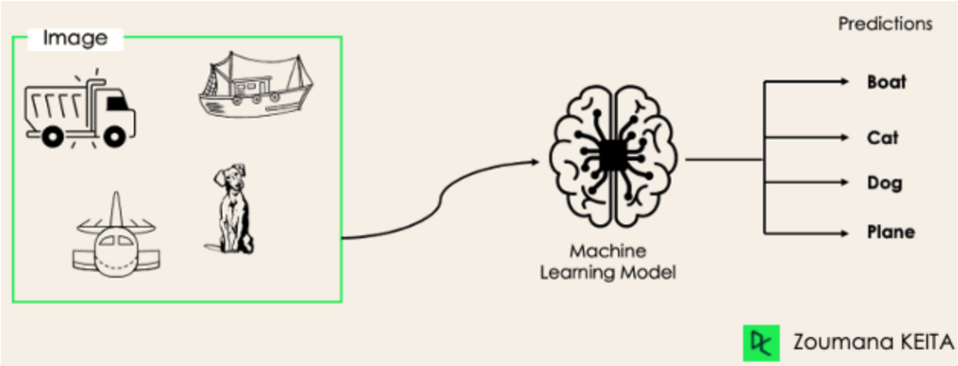
- In **multi-label classification**, each input example can be assigned **0 or more labels**, unlike **binary** or **multi-class classification**, where each example belongs to only one class.
- There is **no mutual exclusivity** among labels.

Examples

- **Text Classification**: A news article can belong to **both "Politics" and "Economy"** categories.
- **Image Recognition**: A picture may contain **both a "Car" and a "Person"**.

How It Works?

- The **model** analyzes input features and predicts **multiple labels**.
- In the **example image**, a **machine learning model** processes an input **image** and identifies multiple objects:



Why Not Use Multi-Class or Binary Classification?

- **Binary classification** can only assign **one of two labels** (e.g., "Spam" or "Not Spam").
- **Multi-class classification** assigns **one class out of many possible classes** (e.g., predicting an animal as either "Dog", "Cat", or "Bird").
- **Multi-label classification** is required when an instance needs **multiple labels** at once.

Algorithms Used

Some standard classification models have specialized versions for multi-label classification:

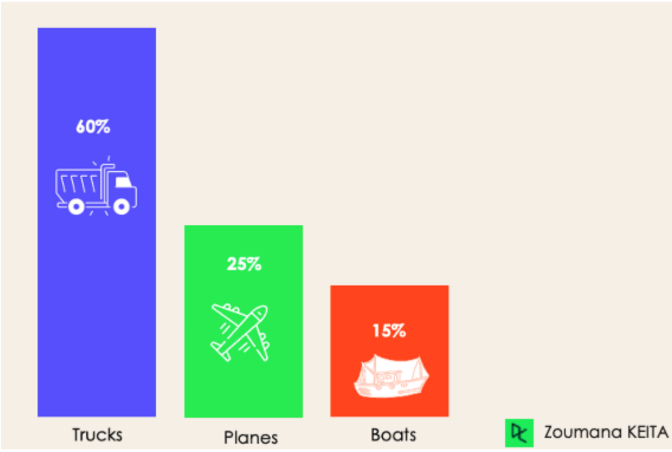
- **Multi-label Decision Trees**
- **Multi-label Gradient Boosting**
- **Neural Networks with Sigmoid Activation**

Imbalanced Classification

Definition

- **Imbalanced classification** occurs when the number of examples in different classes is **unevenly distributed** in the training dataset.
- Some classes have **more data points**, while others have very few, leading to bias in the model.

Example Scenario (from the image)



- A **3-class classification problem** with:
  - **60% Trucks** (majority class)
  - **25% Planes** (medium-sized class)
  - **15% Boats** (minority class)
- The model may become biased and **favor the majority class**, often misclassifying minority classes.

Real-World Applications

- **Fraudulent transaction detection** in financial industries (fraud cases are rare).
- **Rare disease diagnosis**, where very few patients have a certain condition.
- **Customer churn analysis**, where only a small percentage of customers leave a service.

Challenges & Problems

- Traditional models like **Decision Trees and Logistic Regression** may be ineffective.
- The model might **overpredict the majority class** and ignore the minority class.
- **Minority class examples** can be **misclassified as noise** instead of being learned properly.

Solutions to Handle Imbalanced Data

- **Resampling techniques**: Oversampling the minority class (SMOTE) or undersampling the majority class.
- **Class-weight adjustment**: Assigning higher penalties for misclassifying minority classes.
- **Anomaly detection methods**: Useful when rare events need to be detected.

Decision Tree

A **Decision Tree** is a machine learning algorithm used for classification and regression tasks. It splits data into subsets based on the most significant feature at each step, forming a tree-like structure with **nodes** (questions) and **branches** (decisions/outcomes).

How It Works:

1. **Splitting**: The tree recursively divides the data by choosing the feature that best splits the data (using metrics like Gini impurity or variance reduction).
2. **Nodes**: The root node asks the first question, and internal nodes further split data based on other features. Leaf nodes contain the final decision or prediction.
3. **Stopping**: The tree stops growing when it reaches a predefined depth or when splitting no longer improves results.

Advantages:

- Easy to understand and interpret
- Can handle non-linear relationships
- Doesn’t require feature scaling

Disadvantages:

- Prone to overfitting (if too deep)
- Can be unstable with small changes in data
- May underfit if the tree is too shallow

SVM

Support Vector Machine (SVM) Overview

SVM is a supervised learning algorithm used for classification and regression. It finds an optimal hyperplane to separate classes while maximizing the margin.

Key Concepts:

- **Hyperplane:** Separates data points in a feature space.
- **Margin:** Distance between the hyperplane and closest points.
- **Support Vectors:** Critical data points that define the hyperplane.

Types of SVM:

- **Linear SVM:** Works when data is linearly separable.
- **Non-Linear SVM:** Uses kernel functions (e.g., RBF, polynomial) for complex data.

Pros/ Advantages :

✅ Works well in high-dimensional spaces.

✅ Memory-efficient, using only support vectors.

✅ Effective for small to medium-sized datasets.

✅ Handles non-linear classification using kernels.

Cons/Disadvantages:

❌ Computationally expensive for large datasets.

❌ Choosing the right kernel function is tricky.

❌ Less effective when data is highly noisy or overlapping.

❌ Hard to interpret compared to decision trees or logistic regression.

Random Forest

Random Forest Overview

Random Forest is a powerful ensemble learning algorithm used for classification and regression. It builds multiple decision trees using different subsets of data and features, then combines their outputs to improve accuracy and reduce overfitting. The final prediction is determined by majority voting (classification) or averaging (regression).

Key Features:

- **Bagging:** Uses random subsets of data to train each tree.
- **Feature Randomness:** Selects random feature subsets to make trees more diverse.
- **Ensemble Learning:** Aggregates multiple trees for better generalization.

Pros & Cons:

✅ High accuracy and robustness.

✅ Reduces overfitting compared to individual trees.

✅ Provides feature importance insights.

❌ Computationally expensive for large datasets.

❌ Less interpretable than a single decision tree.

Naive Bayes

Naive Bayes Overview

Naive Bayes is a supervised machine learning algorithm based on **Bayes' Theorem**, which calculates the probability of a class given certain features. It assumes that all features are independent within a class, simplifying computations. Despite this "naive" assumption, it performs well in many real-world scenarios, especially in **text classification, spam filtering, and sentiment analysis**.

Key Variants:

- **Gaussian Naive Bayes:** Assumes continuous features follow a normal distribution.
- **Multinomial Naive Bayes:** Used for text classification, handling word counts or frequencies.
- **Bernoulli Naive Bayes:** Works with binary features, commonly used in spam detection.

Pros & Cons:

✅ Fast, simple, and efficient for large datasets.

✅ Performs well with high-dimensional data.

✅ Effective for text-based applications.

❌ Assumes feature independence, which may not hold in real-world data.

❌ Struggles with unseen categorical values (solved by Laplace smoothing).

multi class classification using scikitlearn

Regression

Regression is a supervised machine learning technique used to predict a continuous numerical value based on one or more input features. Unlike classification, which deals with categorizing data, regression focuses on establishing a relationship between independent variables and a dependent variable to make predictions about future or unseen data.

Types of regression :

1. Simple Linear Regression

Simple linear regression is used when there is a **linear relationship** between a dependent variable and one independent variable. The model fits a straight line through the data points by minimizing the error (difference between the actual and predicted values)

- **Use Case:** Predicting outcomes such as the relationship between years of experience and salary in a company.

2. Multiple Linear Regression

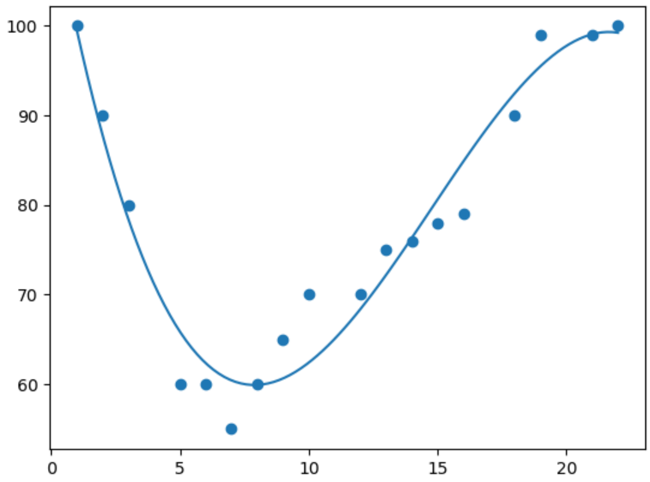
Multiple linear regression extends simple linear regression by using **multiple independent variables** to predict a dependent variable. This model is useful when the prediction depends on several factors.

- **Use Case:** Predicting house prices based on multiple factors such as size, location, and the number of bedrooms.

3. Polynomial Regression

Polynomial regression is an extension of linear regression, but it can model **curvilinear relationships** by fitting a polynomial equation instead of a straight line. This allows it to capture non-linear patterns in the data.

- **Use Case:** Modeling the trajectory of a moving object or predicting data with exponential growth.



4. Ridge Regression

Ridge regression is a regularized form of linear regression that penalizes large coefficients in order to **reduce overfitting**. It adds an L2 regularization term (the sum of the squared coefficients) to the loss function.

- **Use Case:** Ridge regression is particularly useful when you have multicollinearity (high correlation between predictors).

5. Lasso Regression

Lasso regression (Least Absolute Shrinkage and Selection Operator) is similar to ridge regression but uses **L1 regularization**, which penalizes the absolute values of the coefficients. This often results in some coefficients becoming exactly zero, effectively **performing feature selection**.

- **Use Case:** Lasso is useful when you want to reduce the number of features and only keep the most significant ones.

6. Elastic Net Regression

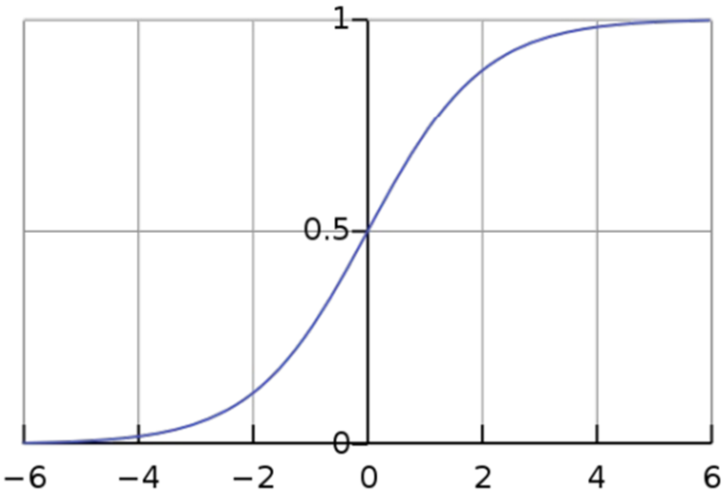
Elastic Net is a **combination** of ridge and lasso regression, incorporating both L1 and L2 regularization terms. It is useful when there are **multiple correlated features**. It can handle situations where there are more predictors than data points and where the predictors are highly correlated.

- **Use Case:** Elastic Net is ideal when you have many predictors with high multicollinearity or need a mix of feature selection and regularization.

7. Logistic Regression

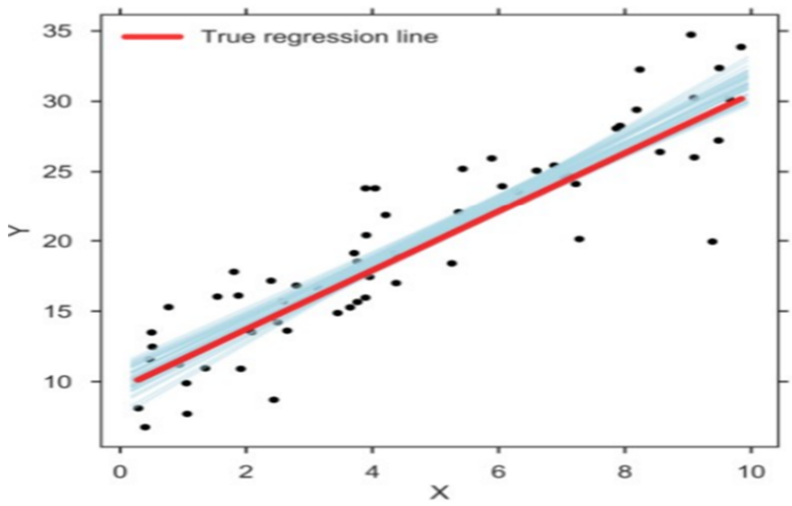
Although called "regression," logistic regression is used for **binary classification** tasks. It estimates the probability of an event occurring by using the **logistic function** (also known as the sigmoid function) to map any real-valued number into the range (0, 1).

- **Use Case:** Logistic regression is widely used in scenarios like **spam detection** (spam or not) or predicting whether a customer will buy a product (yes or no).
-



8. . Bayesian Linear Regression

Bayesian Regression is one of the types of regression in machine learning that uses the Bayes theorem to find out the value of regression coefficients. In this method of regression, the posterior distribution of the features is determined instead of finding the least-squares. Bayesian Linear Regression is like both Linear Regression and Ridge Regression but is more stable than the simple Linear Regression.



regression analysis

Regression Analysis is a statistical technique used to model and analyse the relationship between a dependent variable and one or more independent variables. Its primary goal is to understand how the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed. This method is widely used for prediction, forecasting, and determining the strength of relationships among variables.

Key Components:

- **Dependent Variable (Target):** The outcome or the variable we are trying to predict or explain.
- **Independent Variables (Predictors):** The variables that provide the input or cause the change in the dependent variable.
- All the types of regressions are also the types of regression analysis

classification vs regression

Aspect	Regression	Classification
Objective	Predict a **continuous value** based on input features. Used for tasks where the output is a real number.	Assign input data to a **discrete class** or category. Used when the output is a label or category.

Output	The output is a **real-valued number**. The prediction could be any value within a range (e.g., price, temperature).	The output is a **categorical label**. It assigns the data to one of several possible categories (e.g., "spam" or "not spam").
Examples	Examples include predicting **house prices**, **stock prices**, or **temperatures** based on input data.	Examples include **spam detection**, **disease diagnosis**, and **image classification** (e.g., "dog" or "cat").
Evaluation Metrics	In regression, metrics like **Mean Squared Error (MSE)**, **R-squared**, and **Mean Absolute Error (MAE)** are used to measure prediction accuracy.	In classification, **Accuracy**, **Precision**, **Recall**, and **F1-score** are used to evaluate the model's ability to classify data correctly.
Algorithms Used	Regression tasks typically use **Linear Regression**, **Ridge Regression**, **Lasso Regression**, and similar models to predict continuous values.	Classification tasks often use **Logistic Regression**, **Decision Trees**, **Support Vector Machines (SVM)**, and **Random Forests** to categorize input data into classes.
Nature of Output	The output is **continuous** and can take any real value, such as predicting a person's salary or the future price of a product.	The output is **discrete** and consists of class labels or categories, such as predicting whether an email is "spam" or "not spam".
Task Type	Regression is used when the goal is to predict a **numerical value** based on input data. Common in forecasting and financial predictions.	Classification is used when the goal is to categorize input into **predefined classes**. It is common in tasks like image recognition, medical diagnosis, and sentiment analysis.

**gradient descent & cost function

Gradient Descent

Gradient Descent is a fundamental optimization algorithm in machine learning used to minimize a **cost function** by iteratively adjusting model parameters. The objective is to find the parameters that result in the lowest possible value of the cost function, thereby improving the model's accuracy.

Cost Function:

A **cost function**, also known as a loss function, quantifies the difference between the model's predicted outputs and the actual target values. By calculating this difference, the cost function provides a measure of how well the model is performing. The goal is to minimize this function to enhance the model's predictive accuracy.