- **Fraud Detection**: Identifying fraudulent transactions by analyzing transaction characteristics.

Decision Tree Variants

- **Pruning**: Pruning reduces the complexity of a tree by removing branches that have little importance, reducing overfitting.
- **Ensemble Methods**: Algorithms like **Random Forests** and **Gradient Boosted Trees** combine multiple Decision Trees to improve accuracy and reduce overfitting.

Linear Regression

Linear Regression is a simple yet powerful algorithm in machine learning and statistics used to model the relationship between a **dependent variable (target)** and one or more **independent variables (features)**. The main objective of linear regression is to find the best-fit line that represents the relationship between the variables, allowing us to make predictions for the target variable.

Types of Linear Regression

1. **Simple Linear Regression**:
   - Models the relationship between a single independent variable and the dependent variable.
   - The best-fit line is defined by the equation:

     $$y = mx + b$$

     where y is the predicted output, m is the slope of the line (the coefficient of x), and b is the intercept (the value of y when x=0).

2. **Multiple Linear Regression**:
   - Models the relationship between two or more independent variables and the dependent variable.
   - The equation for multiple linear regression becomes:

     $$y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n$$

     where b0 is the intercept, and b1,b2,…,bn are the coefficients for each independent variable.

Objective of Linear Regression

The objective is to find the values of b0,b1,…,bn that minimize the difference between the predicted values and the actual values in the dataset. This difference is usually measured using the **Mean Squared Error (MSE)**, which calculates the average squared difference between predicted and actual values. By minimizing MSE, linear regression finds the best-fit line.

The **loss function** or **cost function** for MSE is given by:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where yi is the actual value, yi^ is the predicted value, and N is the total number of observations.

Assumptions of Linear Regression

To work effectively, linear regression relies on several key assumptions:

1. **Linearity**: There is a linear relationship between the dependent and independent variables.
2. **Independence**: Observations are independent of each other.
3. **Homoscedasticity**: The variance of the residuals (errors) is constant across all levels of the independent variables.
4. **Normality**: The residuals are normally distributed.
5. **No Multicollinearity** (for multiple linear regression): Independent variables should not be highly correlated with each other.

How Linear Regression Works

1. **Fitting the Line**:
   - Linear regression tries to find the line that best fits the data points by adjusting the coefficients to minimize the error.
   - This is typically achieved using **Ordinary Least Squares (OLS)**, which minimizes the sum of squared residuals.
2. **Prediction**:
   - Once the model is trained, predictions for new data points can be made using the learned coefficients and the equation of the line.
3. **Evaluation**:
   - The model's performance is evaluated using metrics such as **R-squared** (which shows the proportion of variance explained by the model) and **Mean Absolute Error (MAE)** or **Root Mean Squared Error (RMSE)**.

Example of Linear Regression

Suppose we want to predict a house's price based on its size. Using simple linear regression:

1. We plot house prices (y) against house sizes (x).
2. The model fits a line that minimizes the distance between the actual prices and the predicted prices based on size.
3. Using the line's equation, we can predict the price for a house of any given size.

Advantages of Linear Regression

- **Simplicity**: Linear regression is easy to understand and interpret.
- **Efficiency**: It requires minimal computation and works well with smaller datasets.
- **Interpretability**: Coefficients give a clear indication of the relationship between each feature and the target.

Disadvantages of Linear Regression

- **Sensitive to Outliers**: Outliers can significantly affect the fit of the line.
- **Limited to Linear Relationships**: Linear regression cannot model complex, non-linear relationships unless transformed variables are used.
- **Assumptions**: Violation of its assumptions (such as homoscedasticity or independence) can reduce its accuracy.

Applications of Linear Regression

- **Predicting Prices**: Widely used to forecast prices, like real estate, stock prices, and sales.
- **Risk Assessment**: Used in financial sectors to estimate risk and returns.
- **Medical Research**: To examine relationships between medical factors and outcomes.
- **Marketing and Advertising**: Predicts customer spending, sales based on past data, and advertising budgets.

Logistic Regression

Logistic Regression is a popular supervised learning algorithm used for **binary classification** tasks. Unlike linear regression, which predicts continuous values, logistic regression predicts **probabilities** that a given input belongs to a particular class, often classifying results as 0 or 1. It is widely used in areas such as medical diagnosis, email spam detection, and financial risk assessment.

How Logistic Regression Works

Logistic regression models the relationship between input features and the probability of a particular outcome using the **logistic (sigmoid) function**:

$$f(x) = \frac{1}{1 + e^{-z}}$$

where z is a linear combination of the input features:

$$z = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n$$

The sigmoid function squashes the output of the linear equation z into a range between 0 and 1, which can be interpreted as a probability.

- If f(x)>0.5f the model classifies the instance as **1** (positive class).
- If f(x)<0.5 it is classified as **0** (negative class).

Types of Logistic Regression

1. **Binary Logistic Regression**:
    - Used for binary classification (two classes), such as predicting yes/no, success/failure, or spam/not spam.
2. **Multinomial Logistic Regression**:
    - Used for multiclass classification (more than two classes) where classes are not ordered. For example, classifying types of fruits as apple, orange, or banana.
3. **Ordinal Logistic Regression**:
    - Used for ordinal classification tasks where classes have a natural order, such as ratings from 1 to 5.

Log-Loss (Cost Function)

Logistic regression is trained using a cost function called **log-loss** or **binary cross-entropy**. The log-loss function penalizes incorrect predictions more heavily, especially when they are confidently wrong.

The log-loss function is given by:

$$\text{Log-Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

where:

- N is the number of observations,
- Yi is the actual label (0 or 1),
- y^i is the predicted probability.

By minimizing the log-loss function, logistic regression finds the best-fit line that maximizes the probability of correctly classifying each instance.

Example of Logistic Regression

Imagine a logistic regression model that predicts whether a customer will buy a product based on their **age** and **income**. The model uses a linear combination of age and income to compute the probability that a customer will make a purchase.

1. **Compute z**:
    - For a given customer, calculate z=b0+b1×age+b2×income
2. **Apply the Sigmoid Function**:
    - Pass z through the sigmoid function to get a probability.
3. **Classify**:
    - If the probability is greater than 0.5, classify the customer as a buyer (1); otherwise, classify as a non-buyer (0).

Advantages of Logistic Regression

- **Interpretable**: Logistic regression provides clear interpretations of how each feature influences the outcome.
- **Efficient**: It is computationally less intensive and works well with large datasets.
- **Works Well for Linearly Separable Data**: Effective for problems where classes are approximately linearly separable.
- **Probability Outputs**: Logistic regression not only classifies data but also provides probability estimates, useful in decision-making.

Disadvantages of Logistic Regression

- **Limited to Linear Boundaries**: Logistic regression may perform poorly if the relationship between the input variables and the output is highly non-linear.
- **Sensitive to Outliers**: Outliers can influence the model's performance significantly.
- **Requires Feature Scaling**: Logistic regression assumes that features are on a similar scale, so scaling or normalization may be needed.

Applications of Logistic Regression

- **Medical Diagnosis**: Predicting the likelihood of diseases or conditions based on patient data.
- **Credit Scoring**: Assessing the probability of loan default based on applicant data.
- **Marketing**: Estimating the likelihood that a customer will purchase a product or respond to an advertisement.
- **Email Classification**: Classifying emails as spam or not spam.

Support Vector Machines (SVM)

Support Vector Machines (SVM) are a popular supervised learning algorithm used for both **classification** and **regression** tasks. SVMs are especially effective for complex and high-dimensional data, as they aim to find the **optimal hyperplane** that best separates data points of different classes.

How SVM Works

1. **Separating Hyperplane**:
    - In a binary classification task, SVM attempts to find a hyperplane (a line in 2D, a plane in 3D, or a higher-dimensional surface) that maximally separates the two classes.
    - The best hyperplane is the one with the **maximum margin**, or the largest possible distance between the hyperplane and the closest points from each class.
2. **Support Vectors**:
    - The points closest to the hyperplane are called **support vectors**. These points define the margin of separation between the classes.

  - Only the support vectors are used to construct the hyperplane, making SVM computationally efficient even in high-dimensional spaces.
3. **Maximizing the Margin**:
  - SVM tries to maximize the margin (distance between the hyperplane and support vectors) to ensure better generalization.
  - The larger the margin, the lower the risk of misclassification on new data.

Types of SVM

1. **Linear SVM**:
  - If the data is linearly separable (can be separated by a straight line in 2D, or a plane in 3D), a **linear SVM** can be used.
  - The goal is to find a linear hyperplane that separates the classes with the maximum margin.
2. **Non-Linear SVM**:
  - For data that is not linearly separable, SVM uses a **kernel trick** to map data into a higher-dimensional space where a linear separation is possible.
  - Common kernels include:
    - **Polynomial Kernel**: Creates a polynomial decision boundary.
    - **Radial Basis Function (RBF) or Gaussian Kernel**: Maps data points into an infinite-dimensional space, effective for complex, non-linear boundaries.
    - **Sigmoid Kernel**: Often used in neural networks, approximates the behavior of a sigmoid function.

SVM Optimization Objective

The objective of SVM is to find the hyperplane that minimizes classification error by maximizing the margin. This is achieved by solving an **optimization problem** that minimizes the norm of the weight vector while maintaining correct classification for all training points.

For the linear SVM, the optimization problem can be summarized as:

$$\text{Minimize } \frac{1}{2}\|w\|^2$$

subject to the constraints:

$$y_i(w \cdot x_i + b) \geq 1$$

where w is the weight vector, b is the bias term, yi is the label (either +1 or -1), and xi represents the data points.

Soft Margin and Hard Margin

- **Hard Margin SVM**:
  - Used for perfectly linearly separable data, allowing no misclassification. This approach, however, is sensitive to outliers and noise.
- **Soft Margin SVM**:
  - Allows some misclassification to handle cases where data is not perfectly separable.
  - A **regularization parameter (C)** is introduced to control the trade-off between maximizing the margin and minimizing classification errors.
  - Higher values of C create a narrower margin with fewer misclassifications, while lower values allow a wider margin with some misclassifications.

Advantages of SVM

- **Effective in High Dimensions**: SVM performs well in high-dimensional spaces and is often effective when the number of features is greater than the number of samples.
- **Robust to Overfitting**: The margin maximization approach helps reduce overfitting, especially for linearly separable data.
- **Works Well with Non-linear Data**: With kernel tricks, SVM can handle non-linear relationships in the data.

Disadvantages of SVM

- **Computational Complexity**: Training an SVM can be slow for very large datasets, especially with complex kernel functions.
- **Less Effective with Noisy Data**: SVM may struggle with overlapping classes and noisy data, as it attempts to find a distinct margin.
- **Requires Feature Scaling**: SVM is sensitive to feature scaling, so features should be normalized or standardized.

Example of SVM for Binary Classification

Imagine we want to classify emails as **spam** or **not spam** based on features such as word frequency, email length, and sender reputation.

1. **Select Hyperplane**:
   - SVM will find a hyperplane that separates spam and non-spam emails with the maximum margin.
2. **Support Vectors**:
   - Emails closest to the hyperplane act as support vectors.
3. **Prediction**:
   - New emails are classified based on their position relative to the hyperplane.

Applications of SVM

- **Text Classification**: Commonly used for spam detection and sentiment analysis.
- **Image Recognition**: SVM is effective for image classification tasks, such as handwriting recognition.
- **Bioinformatics**: Used in gene classification, protein structure prediction, and cancer classification.
- **Face Detection**: SVMs can classify regions in an image as face or non-face.

Unit 4

Unsupervised Learning

Unsupervised Learning is a type of machine learning where the algorithm learns from **unlabeled data** without predefined categories or target outcomes. The goal is for the model to **discover patterns, structures, or relationships** within the data independently. This approach is especially useful when labeled data is unavailable, and it enables insights into data organization or grouping based on intrinsic characteristics.

How Unsupervised Learning Works