- **Enhances Interpretability**: Focusing on a few significant principal components simplifies data visualization and interpretation.

Applications of PCA

- **Image Compression**: PCA is often used to reduce the dimensionality of image data, keeping essential features while compressing image size.
- **Data Visualization**: PCA enables high-dimensional data to be visualized in 2D or 3D plots, helping identify clusters, patterns, or anomalies.
- **Noise Reduction**: By removing minor components, PCA can help reduce noise and improve the quality of predictions in noisy datasets.

Unit 3

Supervised Learning

Supervised Learning is a type of machine learning in which a model learns from a labeled dataset to make predictions or decisions. In supervised learning, the algorithm is trained on a dataset where each data point includes both **input features** and the **correct output** (label). The model's goal is to learn the relationship between inputs and outputs, allowing it to accurately predict outputs for new, unseen data.

How Supervised Learning Works

1. **Data Collection**: Gather a dataset with labeled examples where each instance has input features and an associated output label.
2. **Data Splitting**: Split the data into **training** and **testing** sets. The training set is used to teach the model, and the testing set evaluates its performance.
3. **Model Training**: Use the training set to build a model that maps inputs to outputs. The model adjusts based on the error between its predictions and the actual labels.
4. **Model Evaluation**: Test the model on the testing set to see how well it generalizes to new, unseen data.
5. **Prediction**: Once trained, the model can predict the output for new inputs based on the patterns it learned during training.

Types of Supervised Learning Tasks

1. **Classification**:

    - In classification tasks, the model learns to assign a category or label to each input.
    - **Examples**: Email spam detection (spam or not spam), image recognition (e.g., recognizing cats vs. dogs).
2. **Regression**:
    - In regression tasks, the model learns to predict a continuous output based on the input data.
    - **Examples**: Predicting house prices based on features like size and location, forecasting stock prices.

Examples of Supervised Learning Algorithms

1. **Linear Regression**: Used in regression tasks to model the relationship between input and output using a linear equation.
2. **Logistic Regression**: A classification algorithm for binary classification tasks, predicting probabilities for two classes.
3. **Decision Trees**: Models decisions based on feature conditions, suitable for both classification and regression.
4. **Support Vector Machines (SVM)**: A classification algorithm that finds the hyperplane best separating the classes.
5. **k-Nearest Neighbors (k-NN)**: A simple classification algorithm that predicts based on the labels of the closest data points.
6. **Neural Networks**: Complex models that can learn intricate patterns, suitable for tasks like image and speech recognition.

Advantages of Supervised Learning

- **High Accuracy**: Labeled data allows the model to learn specific patterns, leading to high accuracy in predictions.
- **Interpretability**: Supervised models can be easier to interpret, especially simpler algorithms like linear regression and decision trees.
- **Efficiency**: Well-trained supervised models can be computationally efficient, allowing fast predictions.

Disadvantages of Supervised Learning

- **Requires Labeled Data**: Collecting and labeling data can be time-consuming and costly, especially for large datasets.
- **Risk of Overfitting**: Models may become too specific to the training data and fail to generalize well to new data.
- **Limited by Training Data**: The model's performance depends heavily on the quality and diversity of the training data.

Applications of Supervised Learning

- **Image Recognition**: Identifying objects, animals, or faces in images.
- **Spam Detection**: Classifying emails as spam or not spam.
- **Sentiment Analysis**: Analyzing text to determine sentiment (positive, negative, neutral).
- **Medical Diagnosis**: Predicting diseases or conditions based on patient data.
- **Financial Forecasting**: Predicting stock prices, sales, or economic indicators.

---

k-Nearest Neighbors (k-NN)

k-Nearest Neighbors (k-NN) is a simple, yet effective supervised learning algorithm used for **classification** and **regression** tasks. It is based on the idea that similar data points are often close together, or "neighbors," in the feature space. The algorithm classifies a new data point by looking at the kkk nearest data points in the training set and making a decision based on their labels or values.

How k-NN Works

1. **Choose a Value for k**: Select the number of neighbors, k, which is usually a small, odd number (e.g., 3 or 5) to avoid ties in classification tasks.
2. **Calculate Distances**: For a new data point, calculate the distance between it and all points in the training dataset. Common distance measures include:
    - **Euclidean Distance**: The most common distance metric for k-NN, calculated as:

$$\text{distance} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

    - where xi and yi are the values of each feature for the two points.
3. **Identify the k Nearest Neighbors**: Sort the distances and select the k closest data points.
4. **Make a Prediction**:
    - **For Classification**: The new data point is assigned the class that is most common among the k neighbors (majority voting).
    - **For Regression**: The prediction is the average of the k neighbors' values.
